Computational Voting Theory:

Game-Theoretic and Combinatorial Aspects

by

Lirong Xia

Department of Computer Science
Duke University

Date: ______________________

Approved:

____
Vincent Conitzer, Supervisor

____
Jérôme Lang

____
Kamesh Munagala

____
Ronald Parr

____
Aleksandar Sasa Pekec

____
Curtis R. Taylor

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Computer Science
in the Graduate School of Duke University
2011

Abstract

(Computer Science)

Computational Voting Theory: Game-Theoretic and Combinatorial Aspects

by

Lirong Xia

Department of Computer Science
Duke University

Date: ____________________

Approved:

____
Vincent Conitzer, Supervisor

____
Jérôme Lang

____
Kamesh Munagala

____
Ronald Parr

____
Aleksandar Sasa Pekec

____
Curtis R. Taylor

An abstract of a dissertation submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in the Department of Computer Science
in the Graduate School of Duke University
2011

Abstract

For at least two thousand years, voting has been used as one of the most effective ways to aggregate people's ordinal preferences. In the last 50 years, the rapid development of Computer Science has revolutionize every aspect of the world, including voting. This motivates us to study (1) **conceptually, how computational thinking changes the traditional theory of voting**, and (2) **methodologically, how to better use voting for preference/information aggregation with the help of Computer Science**.

My Ph.D. work seeks to investigate and foster the interplay between Computer Science and Voting Theory. In this thesis, I will discuss two specific research directions pursued in my Ph.D. work, one for each question asked above. The first focuses on investigating how computational thinking affects the game-theoretic aspects of voting. More precisely, I will discuss the rationale and possibility of using computational complexity to protect voting from a type of strategic behavior of the voters, called *manipulation*. The second studies a voting setting called *Combinatorial Voting*, where the set of alternatives is exponentially large and has a combinatorial structure. I will focus on the design and analysis of novel voting rules for combinatorial voting that balance computational efficiency and the expressivity of the voting language, in light of some recent developments in Artificial Intelligence.

To my dearest wife Jing. The past four years have been a very hard time for both of us. Thank you for loving me, supporting me, encouraging me, and smiling and crying for me since the very beginning. I will never forget the 61 days we spent together in 2008, 108 days in 2009, 45 days in 2010, and 58 days by Aug. 11 in 2011.

Contents

List of Tables

List of Figures

List of Abbreviations and Symbols

Symbols

$\mathcal{C} = \{c_1, \ldots, c_m\}$	The set of alternatives (candidates).
$P = (V_1, \ldots, V_n)$	An n-profile.
m	Number of alternatives.
n	Number of voters.
n'	Number of manipulators.
r	A voting rule.
r^c	A voting correspondence.

Additional Symbols for Combinatorial Voting

p	Number of issues (characteristics).
X_i	The ith issue.
D_i	The ith local domains.
$\mathcal{X} = D_1 \times \cdots \times D_p$	A multi-issue domain composed of p issues.
r_i	A local rule over D_i.
$Seq_{\mathcal{O}}(r_1, \ldots, r_p)$	The sequential composition of local rules $r_1, \ldots, r_p$.
$\mathcal{L}$	Admissible conditional preference set (Definition 12.2.1).
CPnets($\mathcal{L}$)	The set of all CP-nets consistent with $\mathcal{L}$ (Definition 12.2.2).
Pref($\mathcal{L}$)	The set of all linear orders that are consistent with $\mathcal{L}$ (Definition 12.2.2).
$LD(\mathcal{L})$	The lexicographic preference domain (Definition 12.2.2).

Abbreviations

WCM	Weighted coalitional manipulation.
UCM	Unweighted coalitional manipulation.
UCO	Unweighted coalitional optimization.
COd	Coalitional optimization with divisible votes.
GSR	Generalized scoring rules.
NE	Nash equilibrium.
SPNE	Subgame-perfect Nash equilibrium.

Acknowledgements

I am deeply indebted to my supervisor Vincent Conitzer. Needless to say, everyone who knows him knows how great he is. He is a wonderful researcher to work with, a great friend to chat with, and of course, a superb supervisor who always gives me very helpful advises and suggestions. He has always been extremely supportive, patient, and, mostly amazingly, always available WHENEVER I need help. From him I learned a lot: how to do high-quality research, how to manage time as a junior faculty member, and even how to speak English, etc. He makes me firmly believe that coming to Duke is the best choice for my Ph.D. study.

I am also deeply grateful to Jérôme Lang, who is effectively my secondary supervisor. Jérôme has been a frequent collaborator of me and has been helping me so much in research, job search, and many other things ever since before I came to Duke. Besides Vince and Jérôme, I am very fortunate and proud to have Ron Parr, Kamesh Munagala, Sasa Pekec, and Curt Taylor in my Ph.D. committee. Thank you for your advises, feedback, and support in my research as well as job search. Also I should thank Toby Walsh for very insightful and quick feedback on my dissertation. Special thanks to my supervisor Mingsheng Ying when I was in China, who introduced me to the field of Computational Social Choice, and has been very considerate and supportive. In addition to Vince and Jérôme, I am also extremely thankful to Lane Hemaspaandra, Dave Pennock and Toby Walsh for their support in my job search, especially for writing and submitting recommendation letters so

timely. Many thanks to all other friends who have given me very useful feedback on my research statements and helped me in my job search, including Atila Abdulkadiroglu, Itai Ashlagi, Yiling Chen, Edith Elkind, Maria Gini, Mingyu Guo, Edith Hemaspaandra, Shaili Jain, Dima Korzhyk, Nicolas Lambert, Michel Le Breton, Christopher Painter-Wakefield, David Parkes, and Tuomas Sandholm.

I had a very pleasant and rewarding summer intern experience at Yahoo! Research New York in 2010. It was my great pleasure to work with and learn from many excellent researchers in the Microeconomics group there, including Dave Pennock, Dan Reeves, Sebastien Lahaie, and Giro Cavallo. During my Ph.D. study, I have been very fortunate to work and publish with many fantastic researchers, including Sayan Bhattacharya, Yann Chevaleyre, Vincent Conitzer, Tao Jiang, Jérôme Lang, Lan Liu, Nicolas Maudet, Jérôme Monnot, Kamesh Munagala, Nina Narodytska, Dave Pennock, Ariel Procaccia, Matthew Rognlie, Jeff Rosenschein, Toby Walsh, Jing Xiao, and Michael Zuckerman. I also thank a James B. Duke Fellowship and my supervisor Vince Conitzer's NSF CAREER 0953756 and IIS- 0812113, and an Alfred P. Sloan fellowship for support.

Last but of course not least I want to thank my parents, to whom I owe the most for their selfless love and support.

1

Introduction

People hold different opinions and preferences over almost everything. Yet in many situations a common decision must be made. For example, sometimes people need to select a leader, or decide whether or not to provide a public good such as national defense. The best-known way to achieve these goals is by *voting*, which has been a critical component of democracy since ancient time. As early as around 350 B.C., Plato (424/423 B.C.–348/347 B.C.), in spite of being famous for his objection against democracy, proposed several multi-stage voting processes to elect the "guardians" of the law and officeholders, etc., in his unfinished book "*The Law*". Obviously Plato was not the first person who thought about voting. In fact, Socrates (469 B.C.–399 B.C.), Plato's teacher, was sentenced to death by a majority voting. Plato thus had good reasons to object to democracy. After Plato, the first well-known voting system that is not based on majority voting was proposed by Ramon Llull (1232–1315). Then, the systematic study of the theory of voting prospered with the French Revolution in the 18th century. During that time, two of the most famous philosophers who made significant contributions to the theory of voting are Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet (1743–1794, also known as

Nicolas de Condorcet, who proposed the Condorcet criterion), and Jean-Charles, chevalier de Borda (1733–1799, who proposed the Borda voting rule). More recently, Kenneth Arrow (a co-recipient of the *Nobel Memorial Prize in Economics* in 1972) showed that it is impossible to design a voting rule that satisfies some very natural properties (Arrow, 1950). This seminal work is thus named *Arrow's impossibility theorem*, and is broadly regarded as the beginning of modern *Social Choice Theory*, which is an active research direction in Economics.

In recent years, rapid developments in computers and networks have brought big changes to human society. Computers not only have helped us solve problems faster, but also have brought revolutions to the ideology of the human society. For example, the ultimate goal of *Artificial Intelligence (AI)* is to build computers that are as "intelligent" as, if not more intelligent than, human beings. These changes have led to many new interdisciplinary areas. In particular, the interdisciplinary area lying in the intersection of Computer Science and Economics has attracted huge attention, partly due to the emerging electronic commerce of the Internet era. One place where Computer Science meets Economics is the new subarea of AI called *Multi-Agent Systems*, which studies interactions and collaborations in systems that consist of multiple intelligent agents (Wooldridge, 2009). Similar as for human beings, voting could help intelligent agents to make a joint decision in many situations. For example, in the system developed by Ephrati and Rosenschein (1991), agents use voting to decide the next step in their joint plan. There are also many applications of voting in electronic commerce, for example, Ghosh et al. (1999) proposed to use voting to help build recommendation systems; Pennock et al. (2000) adopted the core method in traditional Voting Theory—the axiomatic approach—to analyze collaborative filtering algorithms in recommendation systems; and Dwork et al. (2001) proposed to treat web-search engines as agents, and use voting to decide the best matching website.

In many new applications of voting, we encounter an extremely large number of alternatives or an overwhelming amount of information, which leads to significant computational challenges. To handle these situations, we need to design faster algorithms or build faster computers. On the other hand, higher computational capability makes it easier for voters to figure out beneficial strategic behavior, which might lead to undesirable outcomes. In order to reap the benefits of these potential applications and overcome the emerging problems, we need to develop new algorithms and methodologies. A burgeoning area—*Computational Social Choice*—aims to address problems in computational aspects of information/preference representation and aggregation in multi-agent scenarios (Chevaleyre et al., 2007).

A first question that should be asked is: why it is voting that people or intelligent agents should want to use to aggregate their preferences? Certainly in some situations people use other mechanisms. For example, sometimes auctions are used to determine an allocation of resources or tasks. A key feature in the situations where people or agents use voting is that they only have, or are limited to express, ordinal preferences, in contrast to cardinal preferences measured by real numbers that represent utilities and allow for monetary transfers. In this dissertation, I put aside the discussion of many important topics, including the comparison between voting and other mechanisms, cardinal vs. ordinal preferences, rationale behind the utility theory, etc. An interested reader may refer to Conitzer (2010) for discussions on such topics. Instead, I will focus on the situations where voting is used. It should be kept in mind that voting is a good option for preference/information aggregation in many, but not all situations. My research seeks to investigate and foster the interplay between Computer Science and Voting Theory. In particular, my research focuses the conceptual and methodological aspects of the interplay: (1) **how computational thinking** (Wing, 2006) **changes the traditional voting theory conceptually**, and (2) **methodologically how can we better use voting for**

preference/information aggregation with the help of Computer Science.

1.1 Structure of This Dissertation

The structure of my dissertation is illustrated in Figure 1.1. Most of my research focuses on Computational Voting Theory, which is the most active branch of Computational Social Choice (Node 1 in Figure 1.1). To make the dissertation coherent and to keep it at a reasonable length, I will discuss two research directions that belongs to the two high-level aspects mentioned in the end of the last section. The first direction focuses on investigating how computational thinking affects the game-theoretic aspects of voting (Node 2 in Figure 1.1). The second direction studies the design and analysis of novel voting rules when the set of alternatives is exponentially large and has a combinatorial structure, with the help of some recent developments in Artificial Intelligence (Node 3 in Figure 1.1). These two research directions converge to the study of the game-theoretic aspects of combinatorial voting (Node 4 in Figure 1.1).

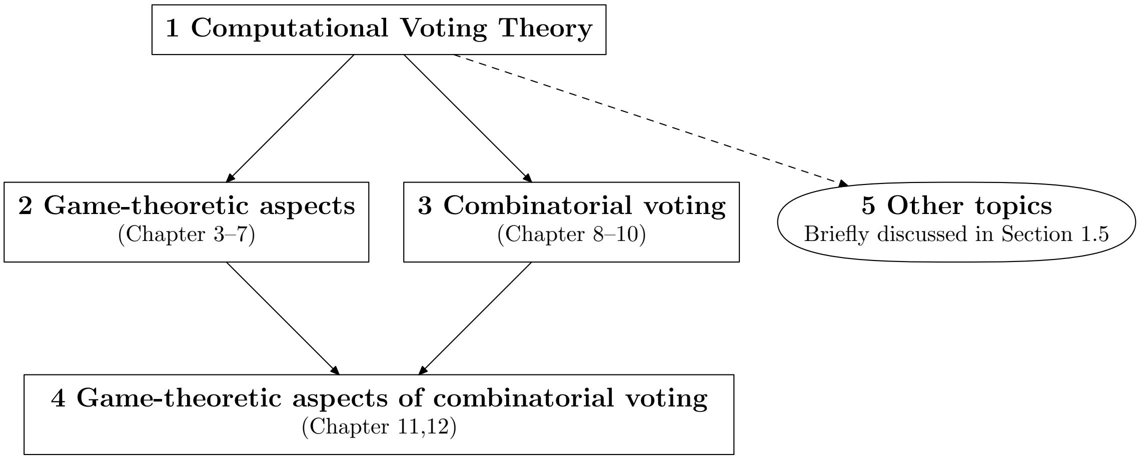

1 Computational Voting Theory
2 Game-theoretic aspects
(Chapter 3–7)
3 Combinatorial voting
(Chapter 8–10)
5 Other topics
Briefly discussed in Section 1.5
4 Game-theoretic aspects of combinatorial voting
(Chapter 11,12)

Figure 1.1: Structure of my dissertation.

In the remainder of this chapter, I will slightly expand on the nodes in Figure 1.1.

1.2 Computational Voting Theory

Computational Voting Theory, which studies computational issues in voting, is the most active branch of Computational Social choice. Throughout the dissertation, a vote is a linear order[1] over the set of alternatives (candidates), we ask each voter (agents) to cast one vote. These votes constitute a *profile*. Then, we apply a voting rule to the profile to determine the winning alternative (the *winner*).

Example 1.2.1. *Suppose three candidates* {*Clinton, Obama, McCain*} *are competing for a presidential position. We use the* plurality *rule to select the winner. That is, the candidate who is ranked at the top the most time in the votes wins, and suppose ties are broken alphabetically. Suppose there are five voters whose votes are as follows:*

Voter 1	:	*Clinton*≻*Obama*≻*McCain*
Voter 2,3:		*Obama*≻*McCain*≻*Clinton*
Voter 4,5:		*McCain*≻*Clinton*≻*Obama*

Then, the winner is McCain, because he is ranked in the top position for two times (tied with Obama), and the tie is broken in favor of McCain.

The formal definition of voting systems and some popular voting rules can be found in Chapter 2. In Computational Voting Theory, researchers have extensively investigated at least the following questions.

- How can we compute the winner or ranking more efficiently?
- How can we communicate and elicit voters' preferences more efficiently?
- How can we use computational complexity to protect elections from bribery and control?

[1] However, see Pini et al. (2007), for a discussion of voting where preferences over the candidates are represented by a partial order.

- How can we prevent voters from misreporting their preferences?
- How can we analyze voters' incentive and strategic behavior?
- How can we design novel voting rules when the set of alternatives has a combinatorial structure, and is exponentially large?

Nodes 2–4 correspond to the last three questions. More detailed discussions as well as references can be found in Chapter 2.

1.3 Node 2: Game-theoretic Aspects

An important yet always implicit assumption when most popular voting rules were designed is that all voters report their preferences truthfully. However, in many real world voting systems, a voter may well lie to make herself better off. This phenomenon is call a *manipulation*. For example, let us recall Example 1.2.1, and suppose that the votes described in the example are the voters' true preferences. We have already seen that if all five voters report truthfully, then McCain is the winner. However, if the first voter reports that her vote is Obama$>$Clinton$>$McCain, while the other voters all report truthfully, then Obama is the winner. Note that the first voter prefers Obama to MaCain, which means that she has an incentive to misreport her preferences to make herself better off. This kind of strategic behavior makes the outcome of the voting process unpredictable, and can sometimes hurt the voters, including the manipulators themselves, when there is more than one manipulator. Therefore, it is important to investigate the strategic behavior of the voters. This falls under *Game Theory* (Fudenberg and Tirole, 1991). First of all, it would be great if we can use a voting rule for which there is never any opportunity for manipulation, i.e., a *strategy-proof* voting rule. This objective might seem to be too ambitious at first glance, but in fact, there are many strategy-proof mechanisms in other settings where voters are allowed to express their cardinal preferences, their preferences are quasi-linear, and monetary transfers are allowed. For example, the well-known VCG mechanisms are strategy-proof (Vickrey, 1961; Clarke, 1971; Groves, 1973). Unfortunately, in voting settings where no monetary transfers are allowed, due to the celebrated Gibbard-Satterthwaite theorem (Gibbard, 1973; Satterthwaite, 1975), when there are three or more alternatives, no strategy-proof voting rule satisfies the following two desired properties: (1) non-imposition (i.e., each alternative wins for some profile) and (2) non-dictatorship (i.e., there is no dictator, a voter whose first-ranked alternative is always the winner). To circumvent this very negative result, economists have proposed to restrict the domain of preferences to obtain strategy-proofness. That is, we assume that voters' preferences always lie in a restricted set of linear orders. One example of such a class is the set of *single-peaked* preferences (Black, 1948). For single-peaked preferences, desirable strategy-proof rules exist, such as the *median* rule (Moulin, 1980). More details can be found in Chapter 12, where I will discuss our own results along this line as well.

Besides this, my research on the game-theoretic aspects of Voting Theory diverges into two directions, illustrated in Figure 1.1. The first direction (the left branch) focuses on exploring the idea of using computational complexity to prevent manipulation. The second direction (the right branch) focuses on analyzing the equilibrium outcome in a type of voting games.

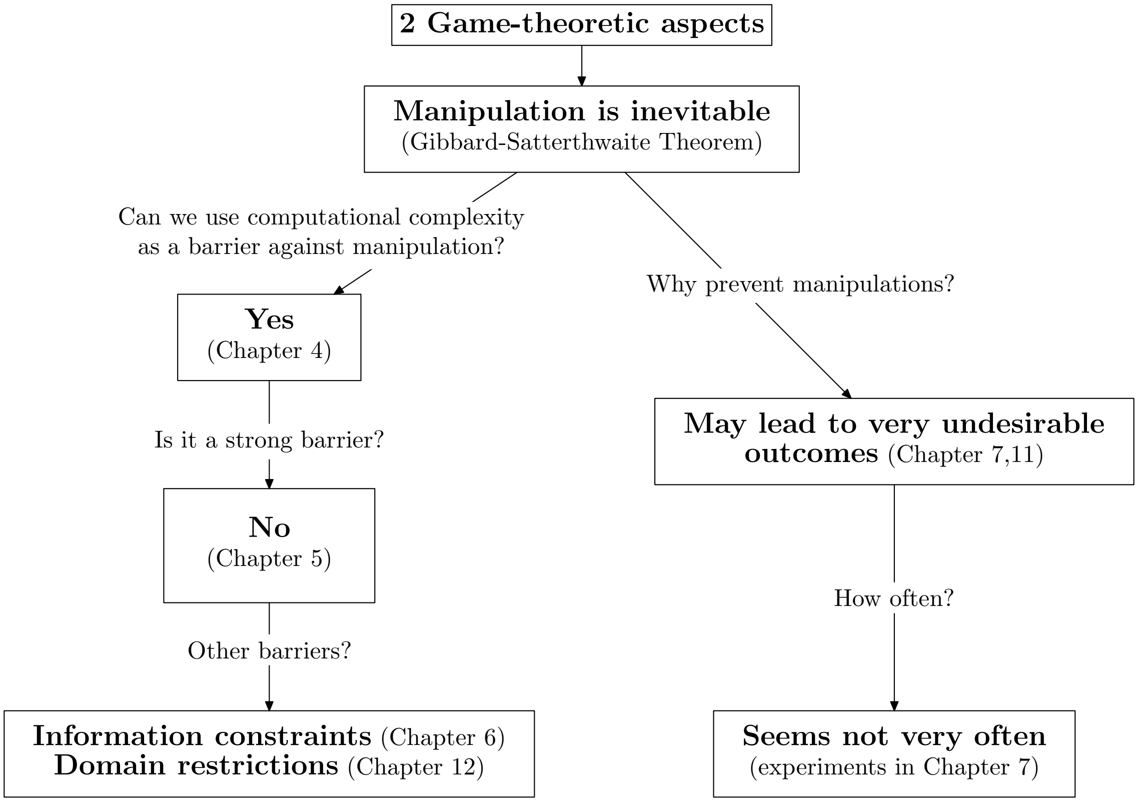

2 Game-theoretic aspects
Manipulation is inevitable
(Gibbard-Satterthwaite Theorem)
Can we use computational complexity as a barrier against manipulation?
Yes
(Chapter 4)
Is it a strong barrier?
No
(Chapter 5)
Other barriers?
Information constraints (Chapter 6)
Domain restrictions (Chapter 12)
Why prevent manipulations?
May lead to very undesirable outcomes (Chapter 7,11)
How often?
Seems not very often
(experiments in Chapter 7)

FIGURE 1.2: Two directions in game-theoretic aspects of voting.

1.3.1 First Direction: Computational Complexity of Manipulation

Even though a manipulation is guaranteed to exist, if we can prove that finding a manipulation is computationally hard for some common voting rules, then a manipulation might not occur simply because the manipulator(s) cannot find it in a reasonable amount of time, or it is computationally too costly to do so. This idea was first explored by Bartholdi et al. (1989a), which, together with Bartholdi et al. (1989b, 1992), have been broadly considered the starting point of Computational Social Choice. After that, a number of results have been obtained on the computational complexity of manipulation in various settings. See Faliszewski et al. (2010b); Faliszewski and Procaccia (2010) for recent surveys. More details will also be given in Chapter 4.

Chapter 4 focuses on the most natural setting where voters are equally weighted, and there are multiple manipulators who want to cast their votes collaboratively to make a favored alternative win. I will show that for some common voting rules, finding a manipulation is NP-hard, while for some other voting rules, there exist polynomial-time algorithms to find a manipulation. Therefore, at least for some common voting rules, the answer to the question "Can we use computational complexity as a barrier against manipulation?" is "Yes". This answer is quite positive, because it implies that at least for these voting rules, even if the potential manipulators use the fastest computer in the world, they are unlikely to find an algorithm that can always tell them the answer quickly even for large instances (assuming P$\neq$ NP). Consequently, these potential manipulators might have less incentive to misreport their preferences.

Proving the NP-hardness of finding a manipulation is only a first step. Even though it is NP-hard to find a manipulation, the manipulators may still not always report their true preferences. For example, they can certainly run a heuristic algorithm for a certain amount of time, say one minute, and if the algorithm returns a successful manipulation, then they will cast the votes returned by the algorithm; otherwise, if the algorithm fails to compute an answer in one minute, they may then report their true preferences. Technically, this problem is due to the fact that NP-hardness is a *worst-case* concept. Therefore, it is natural to ask, informally, whether manipulations are computationally hard to find in "most" cases. Some previous work gave partial answers to this question. Again, more details and discussion can be found in Faliszewski et al. (2010b); Faliszewski and Procaccia (2010) and/or Chapter 4. We will see in Chapter 5 that, for a very general class of voting rules called *generalized scoring rules*, which include many common voting rules, the cases where manipulations are hard to find are exceptions rather than the rule. Therefore, computational complexity does not seem to be a very strong barrier against strategic behavior, so that we need to seek other barriers. For example, we may try to limit the manipulators' information about the preferences of the other voters (Chapter 6), or only allow the voters to pick a vote from a restricted set of linear orders (Chapter 12).[2]

1.3.2 Second Direction: Equilibrium Outcomes in Voting Games

In fact, the very first question that should be asked is, is it ever desirable to prevent the voters' strategic behavior? After all, the ultimate objective of voting is to select a "good" alternative. So if somehow the strategic behavior of the voters leads to the same, or an even better, outcome, then there is no reason to even try to prevent the voters from being strategic. Moreover, in such cases, maybe the strategic behavior should actually be encouraged! Surprisingly, this question was not answered before. To analyze the outcome when voters are strategic, the most natural way is to use Game Theory to model the voting process as a game, and then focus on the winner in the outcome of the game in terms of some solution concept, e.g., *Nash equilibrium*.[3] However, in general a voting game has too many (Nash) equilibria. This makes it very hard to draw any useful conclusions on the impact of strategic behavior on the outcome of voting.

In Chapter 7, we study a type of voting games where voters cast their votes one after another sequentially. We call such games *Stackelberg voting games*. We will focus on a finer solution concept called *subgame-perfect Nash equilibrium*. Fortunately, in any Stackelberg voting game, the outcome is unique in all subgame-perfect Nash equilibria. One might expect that the strategic behavior would sometimes harm the voters, but there are two main difficulties in drawing such a conclusion, which come

[2] As mentioned earlier, this idea has been approached mainly by economists. I will further explore it in the setting of combinatorial voting.

[3] In general simultaneous-move voting games, a Nash equilibrium is a profile where no voter can benefit from casting a different vote. The formal definition of voting games, Nash equilibrium, and its refinement *subgame-perfect Nash equilibrium* can be found in Chapter 7.

from the following two natural questions.

1. *To what extent* can the strategic behavior harm the voters? The main difficulty here is that voting aims at aggregating voters' *ordinal* preferences, which means that generally it is nontrivial to measure how good/bad an alternative is.[4]

2. How *often* does the strategic behavior harm the voters?

Chapter 7 answers the above two questions. The first question is answered by showing some paradoxes, which state that sometimes the (unique) equilibrium winner is ranked in extremely low positions in almost all voters' true preferences. Without doubt this is an extremely undesirable outcome. Therefore, these paradoxes illustrate the cost of strategic behavior of the voters, and suggest that at least in some cases, strategic behavior should be prevented. The second question is partly answered by simulations. Surprisingly, for most common voting rules, the winner in the equilibrium outcome is slightly "better" for the voters on average, compared to the winner when they vote truthfully.

1.4 Node 3: Combinatorial Voting

So far we have been discussing voting over unstructured sets of alternatives. In many real-life situations, there are multiple *issues* (*attributes*, or *characteristics*), and each alternative can be uniquely characterized by a vector of the values these issues take. Such settings are called *combinatorial voting* (or *voting in combinatorial domain*). For instance, when agents vote to select a president and a treasurer, each position corresponds to an issue whose value corresponds to the person selected to hold the

[4] This is in sharp contrast to the settings where there is a well-defined social welfare function, especially in the settings where the agents have quasilinear utility functions, and are allowed to express their cardinal preferences, for example in auctions. In those situations, the cost of strategic behavior can be measured by the *price of anarchy* (Koutsoupias and Papadimitriou, 1999), that is, the ratio of the optimal social welfare over the worst social welfare in equilibrium outcomes.

position. In combinatorial voting, selecting a winner amounts to making a public choice for each of the issues. The main difficulty resides in the exponentially large number of alternatives. Therefore, it is computationally impractical to directly apply a common voting rule designed for unstructured sets of alternatives in the setting of combinatorial voting.[5] For combinatorial voting, we need to design new voting rules that are computationally tractable.

In the literature, researchers in Economics and Political Science have extensively studied voting processes where the agents vote over issues *separately in parallel*. This method works well when agents' preferences over one issue do not depend on any other issues. However, in general agents' preferences over one issue may depend on the value of other issues. For example, if a Democrat is selected to be the president, then a voter may prefer selecting a Republican to be the treasurer; but if a Republican is selected to be the president, then the voter may prefer selecting a Democrat to be the treasurer. There are two main challenges for combinatorial voting: Language-wise we need a more natural way for the agents to truthfully report their preferences. Methodology-wise we also need a more general theory of computational tractable combinatorial voting.

My research in combinatorial voting can be roughly categorized into two directions, illustrated in Figure 1.3. The first direction focuses on designing computationally tractable voting rules for combinatorial voting. The second direction (Node 4 in Figure 1.1) focuses on game-theoretic aspects of combinatorial voting, where we aim at analyzing and preventing voters' strategic behavior in combinatorial voting.

[5] Some voting rules that only use a very small portion of the voters preferences to select the winner, for example the plurality rule, do not have significant computational issues when they are used in combinatorial voting. However, in general these rules will not select a "good" outcome in combinatorial voting. More discussions will be given in Chapter 8.

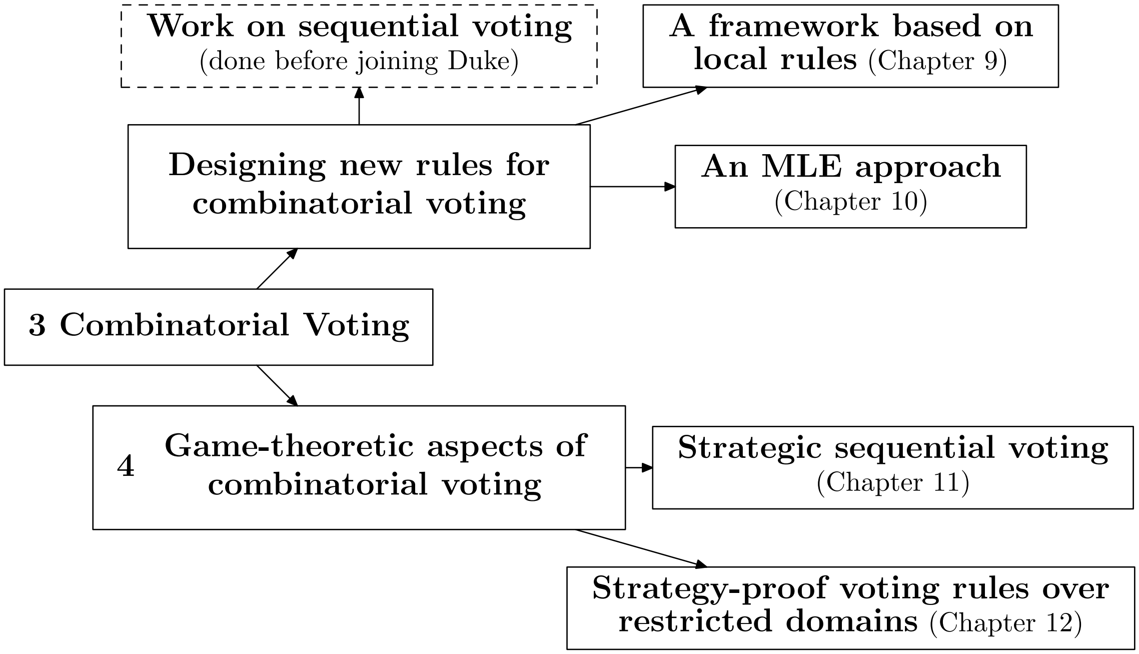

Work on sequential voting
(done before joining Duke)
A framework based on local rules (Chapter 9)
Designing new rules for combinatorial voting
An MLE approach
(Chapter 10)
3 Combinatorial Voting
4
Game-theoretic aspects of combinatorial voting
Strategic sequential voting
(Chapter 11)
Strategy-proof voting rules over restricted domains (Chapter 12)

Figure 1.3: Two directions in combinatorial voting.

1.4.1 Designing New Rules for Combinatorial Voting

One attempt to design computationally tractable voting rules consists of *sequential voting rules*, where agents vote *sequentially*, in the sense that they vote to make the choice for the first issue by a "local" voting rule, then move on to the second issue and vote to make the choice by another local voting rule, etc., given an order over the issues (Lang, 2007). Much of my work was built on the idea of sequential voting, which allows the agents' preferences over one issue to depend on some (but not necessarily all) other issues. Formally, the voters are advised to use a compact voting language called *CP-nets* (Boutilier et al., 2004), which was recently proposed in the Artificial Intelligence community as a preferential counterpart of *Bayesian networks* (Pearl, 1988).

However, in order for sequential voting to work well, there are two levels of technical constraints. First, each voter's preferences must be represented by an *acyclic* CP-net. In other words, for each voter, there exists at least one linear ordering over the issues, such that the voter's preferences over later issues in this order only depend on the values of all previous issues. That is, the voter's preferences are *compatible* with that ordering over issues. The second is that all voters' preferences must be compatible with a common (linear) ordering over the issues. For example, consider a combinatorial voting setting where there are two issues: an issue for "main dish", which can be either fish or beef, and another issue for "wine", which can be either red wine or white wine. The two constraints state that there exists an ordering over the two issues, w.l.o.g. main dish>wine, with which all voters' preferences are compatible. That is, each voter' preferences over wine depends on the value of main dish. From a high-level point of view, these two constraints imply that sequential voting rules have high computational efficiency, but the voting language (i.e., acyclic CP-nets that are compatible with a common ordering over issues) is too restrictive. On the other hand, common voting rules designed for unstructured sets of alternatives have low computational efficiency in the setting of combinatorial voting, but the voters have more flexibility in expressing their preferences.

Designing a good voting rule with high computational efficiency and a fully expressive language seems to be a mission impossible. Therefore, my work in combinatorial voting aims to design voting rules that tradeoff computational efficiency and expressiveness of the voting language. We will start designing such voting rules by assuming all voters vote truthfully (Chapter 9, 10). Complications caused by the strategic behavior of the voters will be examined later (Chapter 11, 12). In Chapter 9, we will see a framework that first considers a directed graph over all alternatives by applying local voting rules, then uses a *choice set function* to select the winner from this graph. This framework allows a voter to use any CP-net (even an acyclic one) to represent her preferences. We will also see that whether or not the voting rule defined by this framework satisfies some desired properties for voting rules, e.g., *anonymity, neutrality, etc.*, depends on both the choice set function and whether the local voting rules satisfy these properties. In general, computing the winners in this framework is hard. However, we will see an algorithm that could save significant amounts of time when the (possibly cyclic) CP-nets that represent voters' preferences share some common structure.

Chapter 10 takes a different approach towards defining new rules for combinatorial voting. Suppose there is a "correct" winner and the voters' preferences are noisy perceptions of it. If we have a probabilistic model that generates voters' preferences given the "correct" winner, and a probability distribution for an alternative to be the "correct" winner, then having seen the voters vote, we can compute the posterior probability for each alternative to be the "correct" winner via standard Bayesian reasoning. In other words, the voting rule defined by this process can be viewed as the *maximum likelihood estimator (MLE)* of the probabilistic model. This idea was actually introduced two hundred years ago by Condorcet (1785) to design a voting rule for unstructured sets of alternatives. The main question is, how should we define the probabilistic model? In Chapter 10, we will see a natural probabilistic model for sets of alternatives composed of binary issues, called *distance-based noise models*, where the conditional probability given the "correct" winner is decomposed into local distributions, one for each issue i. More precisely, the local distribution over any issue i under some setting of the other issues depends only on the Hamming distance from this setting to the restriction of the "correct" winner to the issues other than i. Some results on the computational complexity of winner computation will be presented, followed by discussions about the relation between the MLE approach and sequential voting rules.

1.5 Node 4: Game-Theoretical Aspects of Combinatorial Voting

The formulation of a voting game largely depends on the voting rule used in the voting process. As I argued in the last section, in combinatorial voting it is generally computationally costly to use common voting rules designed for unstructured sets of alternatives. Therefore, the arguments and results in Section 1.3, which were made for common voting rules designed for unstructured sets of alternatives, do not directly apply to combinatorial voting. Since sequential voting is one of the most natural approaches in combinatorial voting, this suggests to study a voting game where voters cast votes strategically on one issue after another, following some ordering over the issues. Indeed, strategic voting is arguably more likely in such a sequential game than in "one shot" voting. We call this type of voting games *strategic sequential voting*, which is the main topic of Chapter 11.[6] Compared to (truthful) sequential voting mentioned in the previous subsection, for strategic sequential voting the focus is on different aspects. In truthful sequential voting, a major concern is how expressive the voting language is. In strategic sequential voting, however, the expressivity of the voting language is not the most important issue. Instead, what really matters is how a strategic voter's preferences and knowledge about the other voters' preferences determine her behavior in the voting game, and thus influence the outcome of the game. Therefore, in the game-theoretic part on combinatorial voting, we are interested in the following two questions. The first question is exactly the same as question 1 asked in Section 1.3.2, but here it is asked for strategic sequential voting.

1. To what extent can the strategic behavior harm the voters in strategic sequen-

[6] We note that strategic sequential voting is different from the Stackelberg voting games mentioned in Section 1.3.2. In Stackelberg voting games voters cast their votes one after another, while in strategic sequential voting, voters cast votes simultaneously on individual issues, one issue after another.

tial voting?

2. If the strategic behavior of the voters can harm the voters badly, how can we prevent it?

The first question is answered by three types of *multiple-election paradoxes*: there exists a profile for which (1) the winner under strategic sequential voting is ranked nearly at the bottom in *all* voters' true preferences, (2) the winner is *Pareto-dominated* by *almost every* other alternative, and as a consequence, (3) the winner is an almost *Condorcet loser*.[7] Even worse, changing the ordering over the issues on which the voters vote cannot completely prevent these paradoxes. Hence, the outcome of strategic sequential voting can be extremely undesirable to all voters. Similar paradoxes have been shown for other models of behavior in combinatorial voting in the literature (Scarsini, 1998; Brams et al., 1998), but as far as we know, we were the first to discover these paradoxes in a strategic environment, to illustrate the cost of the strategic behavior of the voters. See Chapter 11 for more references and discussion.

One approach to addressing the concern raised by the second question is restricting the voters' preferences. We will see in Chapter 11 that by restricting the voters' preferences to be *separable* or *lexicographic*, all three types of multiple-election paradoxes mentioned earlier disappear. In fact, by putting more constraints on the voters' preferences, we can obtain strategy-proof sequential voting rules for combinatorial voting. We can further show that if the domain restriction satisfies some mild conditions, then a voting rule is strategy-proof if and only if it is a sequential voting rule, where each local rule is strategy-proof over its respective local domain. This will be discussed in Chapter 12.

[7] The definitions for Pareto-domination and Condorcet loser will be found in Chapter 11.

1.6 Node 5: Work Excluded from My Dissertation

During my Ph.D. studies, I also have worked on some other important topics in preference/information representation and aggregation. These works will not be discussed in detail in the dissertation due to considerations of length and coherence of the dissertation. In this section, I will briefly describe these works, illustrated in Figure 1.4. An interested reader may also refer to Xia (2010).

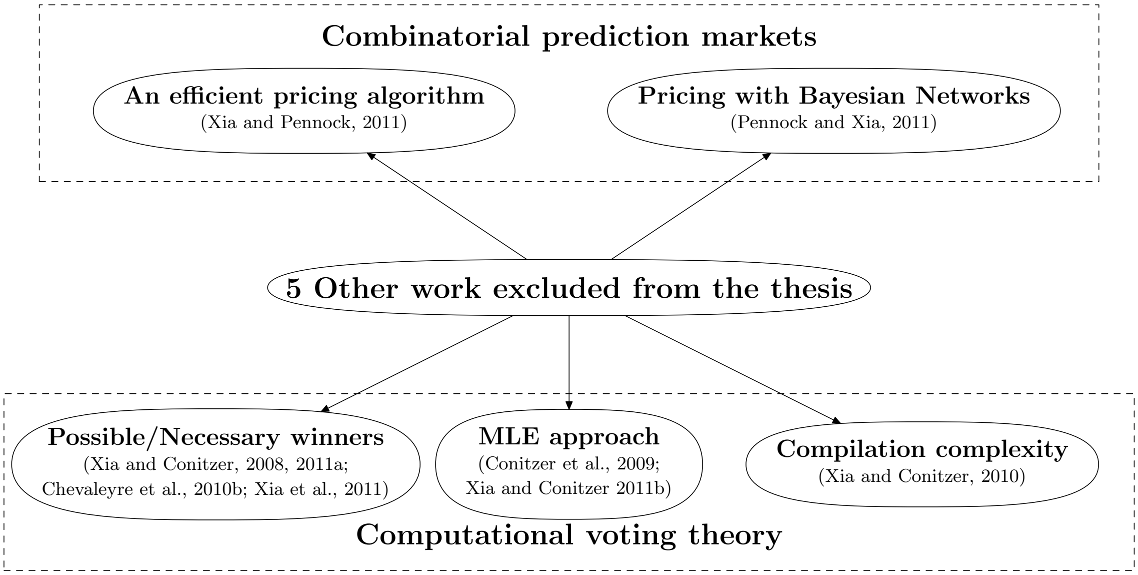

Combinatorial prediction markets
An efficient pricing algorithm
(Xia and Pennock, 2011)
Pricing with Bayesian Networks
(Pennock and Xia, 2011)
5 Other work excluded from the thesis
Possible/Necessary winners
(Xia and Conitzer, 2008, 2011a;
Chevaleyre et al., 2010b; Xia et al., 2011)
MLE approach
(Conitzer et al., 2009;
Xia and Conitzer 2011b)
Compilation complexity
(Xia and Conitzer, 2010)
Computational voting theory

Figure 1.4: Topics excluded from my dissertation.

1.6.1 My Other Work in Computational Voting Theory

In addition to the topics discussed in Section 1.3, 1.4, and 1.5, I have also worked on the following three topics.

- **Computing possible/necessary winners.** In practice, we may not need to know the voters' full preferences to compute the winner. That is, information elicited at an early stage might suffice to conclude who the winner is. For this purpose, it is important to know the answers to the following two computational questions when only part of the voters' preferences are elicited: (1) Is it still possible for a given alternative to win? (2) Has the winner already been determined, so that we may terminate the elicitation process and announce the winner? These two problems are known as the *possible/necessary winner problems*, respectively (Conitzer and Sandholm, 2002; Konczak and Lang, 2005). I investigated the computational complexity of these possible/necessary winner problems for many common voting rules (Xia and Conitzer, 2008a, 2011a), as well as in the special case where the alternatives do not arrive at the same time (Chevaleyre et al., 2010b,a; Xia et al., 2011).

- **Compilation complexity.** One closely related topic to possible/neccessary winner determination is the *compilation complexity* of common voting rules. Here the agents do not arrive at the same time, and we are asked in the middle of the election, what is the lowest number of bits required to store enough information about the votes cast so far to determine the winer (Chevaleyre et al., 2009). In recent work (Xia and Conitzer, 2010a), we proved asymptotically matching upper and lower bounds on the compilation complexity for many common voting rules. We also devised polynomial-time algorithms to "compress" and store the votes in the middle of an election. These algorithms can significantly speedup the algorithm used to compute the subgame-perfect Nash equilibrium in Stackelberg voting games (Chapter 7).

- **A maximum-likelihood estimator approach towards general voting.** As I discussed in Section 1.4.1, one principled way to design a reasonable voting rule is by setting up a probabilistic model, and then define the voting rule to be the maximum-likelihood estimator of this model. Of course this idea is not limited to multi-issue domains, as the idea of using it for unstructured sets of alternatives dates back to Condorcet (1785). In recent work (Conitzer et al., 2009b), we showed that the MLE approach gives us a group of aggregation functions called *ranking scoring rules*, which are used to output an aggregated linear order over all alternatives. The MLE approach can also be used to systematically extend common voting rules that aggregate linear orders to aggregate partial orders (Xia and Conitzer, 2011b).

In addition to the above three topics, I also did some work on sequential voting in combinatorial domains before starting my Ph.D. studies at Duke. This work will be mentioned in Chapter 8 as a part of the literature in combinatorial voting.

1.6.2 Combinatorial Prediction Markets

Prediction markets are financial markets that aggregate agents' probabilistic beliefs about the outcome of a random event. The Iowa Electronic Markets and Intrade are two examples of real prediction markets with a long history of tested results. See Chen and Pennock (2010) for a recent survey of prediction mechanisms. Unfortunately, if the space has a combinatorial structure, then the central problem of computing the prices for securities is #P-hard (Chen et al., 2008a). For example, in the NCAA mens basketball tournament, there are 64 teams and therefore 63 matches in total to predict, where each match can be seen as a binary variable. Such settings are known as *combinatorial prediction markets*.

Recently, I revealed two natural relationships: the first (Xia and Pennock, 2011) bridges combinatorial prediction markets and the *weighted model counting* problem, a central problem in AI; and the second (Pennock and Xia, 2011) bridges combinatorial prediction markets and *probabilistic belief aggregation*, a well-studied problem in both Statistics and AI. Inspired by the first relationship, I designed an efficient novel Monte Carlo sampling technique based on importance sampling that has a good theoretical guarantee, for combinatorial prediction markets for tournaments (Xia and Pennock, 2011). The second relationship helped us further explore the idea of using a compact representation scheme (formally, a Bayesian network) to represent the prices of securities (Chen et al., 2008b), and completely characterize all structure-preserving securities (meaning that these securities can be computationally efficiently priced) (Pennock and Xia, 2011).

1.7 Summary

In this chapter, I categorized some of my Ph.D. work in Computational Voting Theory into two lines of research directions: the game-theoretic aspects and combinatorial voting. I briefly discussed the motivating questions in both lines of research and their intersection, and the results that will be presented in later chapters. To make the dissertation coherent and to keep it at a reasonable length, some of my work that are not included in this dissertation. Some of them were briefly discussed in Section 1.6.

2

Preliminaries

In this chapter, I first give definitions of voting, some common voting rules, and some desired properties. In the end of this chapter, I will give a brief introduction to some other major topics in Computational Social Choice not covered in this dissertation.

Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ denote the set of *alternatives* (or *candidates*). Each voter uses a *linear order* on $\mathcal{C}$ to represent his/her preferences. A linear order is a transitive, antisymmetric, and total relation on $\mathcal{C}$. The set of all linear orders on $\mathcal{C}$ is denoted by $L(\mathcal{C})$. For any natural number n, an n-voter profile P on $\mathcal{C}$ is a vector consisting of n linear orders on $\mathcal{C}$, one from each voter. That is, $P = (V_1, \ldots, V_n)$, where for every $j \leqslant n$, $V_j \in L(\mathcal{C})$. The set of all n-profiles is denoted by $\mathcal{F}_n$. Throughout the dissertation, we let n denote the number of voters, and let m denote the number of alternatives.

For any linear order $V \in L(\mathcal{C})$ and any $i \leqslant m$, we let $\mathrm{Alt}(V, i)$ denote the alternative that is ranked in the ith position in V. A *voting rule* r is a function that maps any profile on $\mathcal{C}$ to a unique winning alternative (the winner), that is, $r : \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \rightarrow \mathcal{C}$. A *voting correspondence* r^c can select more than one winner, that is, $r^c : \mathcal{F}_1 \cup \mathcal{F}_2 \cup \ldots \rightarrow 2^{\mathcal{C}} \backslash \{\varnothing\}$. Mathematically, a voting rule is a special voting correspondence that always selects a unique winner.

2.1 Common Voting Rules

In this section we give definitions of some common voting rules. In fact, most of them are defined to be the maximizer/minimizer of some type of "scores".[1] Therefore, these voting rules are actually defined to be voting correspondences plus some tie-breaking mechanisms. In this paper, if not mentioned specifically, ties are broken in the fixed order $c_1 \succ c_2 \succ \cdots \succ c_m$.[2] Below is a list of common voting rules that will be studied in this thesis.

- **(Positional) scoring rules**: Given a *scoring vector* $\vec{s}_m = (\vec{s}_m(1), \ldots, \vec{s}_m(m))$ of m integers, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $\vec{s}_m(V, c) = \vec{s}_m(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $\vec{s}_m(P, c) = \sum_{j=1}^{n} \vec{s}_m(V_j, c)$. The rule will select $c \in \mathcal{C}$ so that $\vec{s}_m(P, c)$ is maximized. We assume scores are integers and nonincreasing. Some examples of positional scoring rules are *Borda*, for which the scoring vector is $(m-1, m-2, \ldots, 0)$; *plurality*, for which the scoring vector is $(1, 0, \ldots, 0)$; and *veto*, for which the scoring vector is $(1, \ldots, 1, 0)$. When there are only two alternatives, Borda, plurality, and veto (as well as all other voting rules introduced below) are called *majority*.

The definition of positional scoring rules naturally extends to the case in which voters are weighted; the weights are represented by a vector $\vec{w} = (w_1, \ldots, w_n) \in \mathbb{R}_+^n$, where for any $i \leqslant n$, w_i is the weight of voter i. In particular, we let

$$\vec{s}_m(P, \vec{w}, c') = \sum_{i=1}^{n} w_i \cdot \vec{s}_m(V_i, c'),$$

and again, the rule will select $c \in \mathcal{C}$ so that $\vec{s}_m(P, c)$ is maximized.

[1] This idea will be generalized to define a class of voting rules called *generalized scoring rules*. See Section 5.1.

[2] Tie-breaking can have important impact on the properties of voting rules, e.g, the computational complexity of manipulation (Obraztsova et al., 2011; Obraztsova and Elkind, 2011).

- **Copeland**$_\alpha$ $(0 \leqslant \alpha \leqslant 1)$: For any two alternatives c_i and c_j, we can simulate a *pairwise election* between them, by seeing how many votes prefer c_i to c_j, and how many prefer c_j to c_i; the winner of the pairwise election is the one preferred more often. Then, an alternative receives one point for each win in a pairwise election, α points for each tie, and zero point for each loss. The winner is an alternative that maximizes the score.

- **Maximin**: Let $D_P(c_i, c_j)$ denote the number of votes that rank c_i ahead of c_j minus the number of votes that rank c_j ahead of c_i in the profile P. The winner is the alternative c that maximizes $\min\{D_P(c, c') : c' \in \mathcal{C}, c' \neq c\}$.

- **Ranked pairs**: This rule first creates an entire ranking of all the alternatives. In each step, we will consider a pair of alternatives c_i, c_j that we have not previously considered; specifically, we choose the remaining pair with the highest $D_P(c_i, c_j)$. We then fix the order $c_i \succ c_j$, unless this contradicts previous orders that we fixed (that is, it violates transitivity). We continue until we have considered all pairs of alternatives (hence we have a full ranking). The alternative at the top of the ranking wins.[3]

- **Voting trees**: A voting tree is a binary tree with m leaves, where each leaf is associated with an alternative. In each round, there is a pairwise election between an alternative c_i and its sibling c_j; if the majority of voters prefer c_i to c_j, then c_j is eliminated, and c_i is associated with the parent of these two nodes. The alternative that is associated with the root of the tree (i.e., wins all its rounds) is the winner.

- **Bucklin**: The Bucklin score of an alternative c, denoted by $B_P(c)$, is the smallest number t such that more than half of the votes rank c somewhere in the top

[3] We note that at any stage there could be two or more edges whose weights are the highest. In this dissertation, we first use *parallel-universe tie-breaking* (Conitzer et al., 2009b) to select multiple winners, that is, an alternative is a winner if there exists a way to break ties among the edges such that the alternative is ranked in the top position in the ranking created by ranked pairs. After obtaining all "parallel-universe" winners, we use a fixed-order tie-breaking mechanism to select a unique winner from them.

t positions. A Bucklin winner minimizes the lowest Bucklin score, and ties are broken by the number of times that the alternative is ranked within top $B_P(c)$ positions.

• **Plurality with runoff**: The rule has two steps. In the first step, all alternatives except the two that are ranked in the top position the most often are eliminated; in the second round, the plurality rule (a.k.a. *majority* rule in case of two alternatives) is used to select the winner.

• **Single transferable vote (STV), a.k.a. instant-runoff or alternative vote**: The election has m rounds. In each round, the alternative that gets the lowest plurality score (the number of times that the alternative is ranked in the top position) drops out, and is removed from all of the votes (so that votes for this alternative transfer to another alternative in the next round). The last-remaining alternative is the winner.[4]

• **Baldwin's rule:** This is a multi-round voting rule similar to STV. The election has m rounds. In each round, the alternative that gets the lowest Borda score drops out. The last-remaining alternative is the winner.

• **Nanson's rule:** This is another multi-round voting rule similar to STV. The election has multiple rounds. In each round, all alternatives with less than the average Borda score are eliminated. This process then repeated with the reduced set of alternatives until there is a single alternative left. Nanson's rule and Baldwin's rule are closely related, and indeed are sometimes confused (Niou, 1987).

2.2 Axiomatic Properties for Voting Rules

As we discussed in the introduction, since in the voting setting the voters' preferences are ordinal, it is hard to measure how "good" an alternative is to all voters. Therefore, it does not seem to be obvious how can we argue that a voting rule is "good" or

[4] In this dissertation we use fixed-order tie-breaking at all stages. Conitzer et al. (2009b) investigated the STV rule using parallel-universe tie-breaking.

not. To overcome this difficulty, economists have proposed some desired properties (or, *axioms*) that a good voting rule should satisfy, and have investigated how to characterize voting rules by which properties they satisfy. Below, we include a list of such properties. We say a voting rule r satisfies:

- **anonymity**, if the output of the rule is insensitive to the names of the voters;
- **neutrality**, if the output of the rule is insensitive to the names of the alternatives;
- **homogeneity**, if for any profile P and any $n \in \mathbb{N}$, $n > 0$, $r(P) = r(nP)$, where nP is the profile composed of n copies of P;
- **non-imposition**, if any alternative is the winner under *some* profile. That is, for any alternative c and any $n \in \mathbb{N}$, there exists an n-profile P such that that $r(P) = c$;
- **unanimity**, if $\text{Alt}(V, 1) = c$ for all $V \in P$ implies $r(P) = c$;
- **(strong) monotonicity**, if for any profile $P = (V_1, \ldots, V_n)$ and another profile $P' = (V_1', \ldots, V_n')$ such that each V_i' is obtained from V_i by raising only $r(P)$, we have $r(P') = r(P)$;
- **consistency**, if, whenever we have two disjoint profiles P_1, P_2 with $r(P_1) = r(P_2)$, we must have $r(P_1 \cup P_2) = r(P_1) = r(P_2)$;
- **participation**, if for any profile P and any vote V, $r(P \cup \{V\}) \succeq_V r(P)$;
- **Pareto efficiency**, if for any profile P, there is no alternative c that is preferred to $r(P)$ by all the voters;

- **the Condorcet criterion**, if, whenever there exists a *Condorcet winner* in a voting profile P, we must have that $r(P)$ is the Condorcet winner. Here a Condorcet winner is the alternative that wins each pairwise elections;
- **the majority criterion**, if, whenever the majority of voters rank an alternative in the top position, that alternative must be the winner under r.

Table 2.1 summarizes whether some common voting rules mentioned above satisfy these axiomatic properties. The Wikipidea entry for "voting system" (`http://en.wikipedia.org/wiki/Voting_system`) is a good place for the definitions of more voting rules and axiomatic properties.

Table 2.1: Some common voting rules and their axiomatic properties.

	Pos. scoring	Copeland	Maximin	Ranked pairs	STV	Bucklin	Plurality w/ runoff
Anonymity Neutrality Homogeneity Pareto efficiency	Y	Y	Y	Y	Y	Y	Y
Monotonicity	Y	Y	Y	Y	N	Y	N
Consistency Participation	Y	N	N	N	N	N	N
Condorcet	N	Y	Y	Y	Y	N	N
Majority	N	Y	Y	Y	Y	Y	Y

Each of these axiomatic properties evaluates voting rules from a specific viewpoint. For example, anonymity measures how "fair" a voting rule is to the voters, while neutrality measures how "fair" a voting rule is to the alternatives. We next consider some other important concepts in voting.

Definition 2.2.1. *For any profile P, we let* WMG(P) *denote the* weighted majority graph *of P, defined as follows.* WMG(P) *is a directed graph whose vertices are the alternatives. For $i \neq j$, if $D_P(c_i, c_j) \geqslant 0$, then there is an edge (c_i, c_j) with weight $w_{ij} = D_P(c_i, c_j)$.*

Example 2.2.2. *Let P denote the profile defined in Example 1.2.1. The weighted majority graph of P is illustrated in Figure 2.1.*

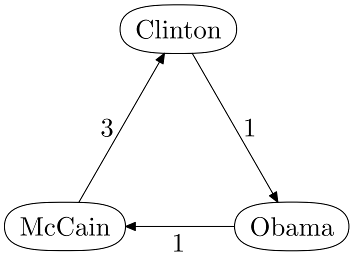


Figure 2.1: The weighted majority graph of the profile define in Example 1.2.1.

We say that a voting rule r is based on the *weighted majority graph (WMG)*, if the winner for r only depends on the weighted majority graph of the input profile. More precisely, for any pair of profiles P_1, P_2 such that $\text{WMG}(P_1) = \text{WMG}(P_2)$, we have $r(P_1) = r(P_2)$.

The following lemma will be frequently used in this dissertation. Informally, the lemma states that for any weighted directed graph G where the weights have the same parity, there exists a polynomially large profile whose WMG is G. This lemma allows us to focus on constructing a WMG that satisfies some desired properties, rather than constructing the profile directly. The lemma was first proved by McGarvey (1953), and there is also some subsequent work studying how to use as few votes as possible to obtain the desired WMG (Erdös and Moser, 1964). In this dissertation, we only need the polynomiality guaranteed by McGarvey's original result.

Lemma 2.2.3. *(McGarvey, 1953) Given a function $F : \mathcal{C} \times \mathcal{C} \to \mathbb{Z}$ such that*

1. *for all $c_1, c_2 \in \mathcal{C}$, $c_1 \neq c_2$, $F(c_1, c_2) = -F(c_2, c_1)$, and*
2. *either for all pairs of candidates $c_1, c_2 \in \mathcal{C}$ (with $c_1 \neq c_2$), $F(c_1, c_2)$ is even, or for all pairs of candidates $c_1, c_2 \in \mathcal{C}$ (with $c_1 \neq c_2$), $F(c_1, c_2)$ is odd,*

there exists a profile P such that for all $c_1, c_2 \in \mathcal{C}$, $c_1 \neq c_2$, $D_P(c_1, c_2) = F(c_1, c_2)$ and

$$|P| \leqslant \frac{1}{2} \sum_{c_1,c_2:\ c_1 \neq c_2} |F(c_1, c_2) - F(c_2, c_1)| \quad .$$

2.3 A Brief Overview of Computational Social Choice

In this section, I will give a more detailed overview of some major topics in Computational Social Choice, which is an emerging interdisciplinary area at the intersection of Computer Science and Economics. Despite being young, Computational Social Choice has already found its place as a major topic in a number of Ph.D. dissertations since 2006, for example, Conitzer (2006a), Estivie (2007), Pini (2007), Altman (2007), Bouveret (2007), LeGrand (2008), Procaccia (2008), Faliszewski (2008), Aziz (2009), Uckelman (2009), Guo (2010), and Betzler (2010). An ever-increasing list of Ph.D. dissertations related to Computational Social Choice can be found at `http://www.illc.uva.nl/COMSOC/theses.html`. The Computational Social Choice workshop (COMSOC) has been held every other year since 2006. Computational Voting Theory is by far the most active research direction in Computational Social Choice. Below I will describe some major research topics in Computational Voting Theory, followed by some other research topics in Computational Social Choice.

2.3.1 Major Topics in Computational Voting Theory

Researchers in Computational Voting Theory have extensively studied the following topics.

- **How can we compute the winner or ranking more efficiently?** In traditional Social Choice Theory, voting rules are designed for aggregating voters' preferences over a generally small set of alternatives, where determining the winner is not a significant computational issue. In fact, computing the winner for many common voting rules can be done in polynomial time. However, for some voting rules that have a long history, it has been shown that computing the winner is hard. For example, computing the winner for Kemeny's rule was shown to be NP-hard by Bartholdi et al. (1989b) and was later shown to be complete for parallel access to NP (Hemaspaandra et al., 2005); similar results have been obtained for Dodgson's rule—computing the winner for Dodgson's rule is NP-hard (Bartholdi et al., 1989b) and is also complete for parallel access to NP (Hemaspaandra et al., 1997). A third example is Slater's rule, for which computing the winner is NP-hard (Ailon et al., 2005; Alon, 2006; Conitzer, 2006b). For these voting rules, efficient approximation/heuristic algorithms have been proposed (Ailon et al., 2005; Conitzer, 2006b; Conitzer et al., 2006; Charon and Hudry, 2000; Hudry, 2006; Betzler et al., 2009a; Caragiannis et al., 2009, 2010). However, if the voters' preferences are restricted to be single-peaked, then a *Condorcet winner* always exists, which means that computing winners for both Kemeny's and Dodgson's rules are in P (Brandt et al., 2010a).

Kemeny's, Dodgson's, and Slater's rules are all defined by first computing the (weighted) majority graph, then applying a *tournament solution* (also called *choice set function* in Chapter 9) to the graph to select the winner. The computational complexity of computing some important tournament solutions has been investigated (Brandt et al., 2009, 2010b, 2011).

- **How can we communicate the voters' preferences more efficiently?** When the number of alternatives is extremely large, it is computationally inefficient for the agents to communicate their full preferences to the center. *Preference elicitation* studies how to query the agents iteratively to elicit enough information for computing the winner (Conitzer and Sandholm, 2002). The lowest number of bits of communication required to compute the winner, called *communication complexity*, was investigated for some common voting rules (Conitzer and Sandholm, 2005b). Eliciting single-peaked preferences were studied in Conitzer (2009) and Farfel and

Conitzer (2011)

Communication complexity provides a worst-case guarantee about the information that must be transmitted in order to compute the winner. However, it is quite likely that in practice, the elicitation process can usually end earlier. As I mentioned in Section 1.6, in these situations one important problem is how to compute the possible/necessary winners (Konczak and Lang, 2005). Besides the works discussed in Section 1.6 (i.e., Chevaleyre et al. (2010b), Chevaleyre et al. (2010a), Xia and Conitzer (2011a), and Xia et al. (2011)), there are a number of other works studying computing possible/necessary winners in different settings (Pini et al., 2007; Walsh, 2007; Betzler et al., 2009b; Betzler and Dorn, 2010; Baumeister and Rothe, 2010; Bachrach et al., 2010; Baumeister and Rothe, 2010; Baumeister et al., 2011).

• **How can we prevent voters from misreporting their preferences?** In this line of research, we investigate the possibility of using computational complexity to prevent manipulation. Therefore, we need computational problem (for a manipulator to compute a manipulation) to be as hard as possible. This topic will be discussed in Section 3.1.

• **How can we use computational complexity to protect elections from bribery and control?** Bribery and control are two ways for someone (not necessarily a voter) to influence the outcome of the election. In general, bribery is the behavior where the briber makes her favored alternative win by paying money to some voters to change their votes. Control in voting is more complicated than bribery in some sense—there are many different types of control, for example, introducing new voters/alternatives, removing existing voters/alternatives, or partition the voters/alternatives. Bartholdi et al. (1992) first studied several types of control in voting systems. Recently, more computational complexity results for bribery and control problems have been obtained. In the bribery problem setting of Faliszewski et al. (2009a), every voter has a cost, and we are asked whether there is a way to bribe some voters to make a given candidate win, subject to a total budget constraint. Elkind et al. (2009b) considered an even finer setting called *swap-bribery*, where the voters are paid to "swap" adjacent alternatives in their votes. Faliszewski et al. (2009b) showed that the Copeland rules (for different α parameters) broadly resist known types of bribery and control. We (Conitzer et al., 2009a) studied the computational complexity of agenda control in sequential voting systems. A special type of control that introduces clones of alternatives was studied by Elkind et al. (2010a). The setting where the briber can use multiple types of bribery/control simultaneously was studied by Faliszewski et al. (2011a). On the negative side, Faliszewski et al. (2011b) showed that if the voters' preferences are single-peaked, then for many common voting rules the manipulation and control problems are in P.

• **How can we analyze voters' incentives and strategic behavior?** This topic will be discussed in Section 3.2.

• **How can we design novel voting rules when the set of alternatives has a combinatorial structure, and is exponentially large?** This topic will be discussed in Chapter 8.

2.3.2 Other Major Topics in Computational Social Choice

Besides Computational Voting Theory, researchers in Computational Social Choice have also studied the following topics.

• **Fair division**, a.k.a. *cake cutting* (Steinhaus, 1948), aims at providing a good allocation of resources that satisfies some desired properties. The most desired property is *envy-freeness*, which states that in the allocation, no agent would prefer the resources allocated to any other agent. Fair division is closely related to Multiagent Resource Allocation (Chevaleyre et al., 2006). For indivisible goods, Lipton et al. (2004) obtained approximability and inapproximability results for envy-free allocations. For one heterogeneous divisible good, Procaccia (2009) proved a lower bound on the number of steps that must be used in any envy-free cake-cutting algorithm. Chen et al. (2010), and Mossel and Tamuz (2010) introduced truthfulness in cake-cutting, and proposed several cake-cutting algorithms that are truthful, envy-free, and also satisfy some other desired properties. Caragiannis et al. (2011) studied the cake-cutting setting where agents' valuation functions are *piecewise uniform with minimum length*. In such settings envy-freeness does not imply *proportionality*, and Caragiannis et al. proposed allocation algorithms that approximate the proportionality. Cohler et al. (2011) proposed an algorithm that computes the "optimal" envy-free allocation, that is, an envy-free allocation that has the highest social welfare among all envy-free allocations.

• **Judgement aggregation.** In judgement aggregation, a group of agents (judges) need to aggregate their opinions over several interrelated binary propositions. Recently, judgement aggregation has attracted much attention in Economics and Political Science (List and Puppe, 2009). It differs from combinatorial voting in the sense that in judgement aggregation, the aggregated values of the propositions must be consistent, while in combinatorial voting there is no such requirement. The best-known motivating example for the study of judgement aggregation is called the "doctrinal paradox" (Chapman, 1998), which is illustrated in the following example.

Example 2.3.1. *Suppose there are three judges who want to decide whether a defendant is liable. They use the majority rule to aggregate their opinions over three binary propositions: (1) $\mathcal{P}$, which is true if and only if the defendant did the action X, (2) $\mathcal{Q}$, which is true if and only if the defendant intended to do X, and finally (3) $\mathcal{R}$, which is true if and only if the judge thinks that the defendant is liable. Suppose all judges agree that the defendant is liable if and only if he did X and intended to do X (that is, $\mathcal{R} \leftrightarrow (\mathcal{P} \wedge \mathcal{Q})$). Consider the following three judges' judgments and the majority aggregation for each proposition.*

Table 2.2: The doctrinal paradox.

	$\mathcal{P}$	$\mathcal{Q}$	$\mathcal{R} \leftrightarrow (\mathcal{P} \wedge \mathcal{Q})$	$\mathcal{R}$
Judge 1	T	T	T	T
Judge 2	T	F	T	F
Judge 3	F	T	T	F
Majority	T	T	T	F

All judges' valuation over these propositions are consistent. However, the proposition-wise aggregations of the judges are not consistent ($\mathcal{P} = T$, $\mathcal{Q} = T$, and $\mathcal{R} \leftrightarrow \mathcal{P} \wedge \mathcal{Q} = T$ imply that $\mathcal{R} = T$). The message behind the example is that: under the majority rule, the judges come to the conclusion that the defendant did X and intended to do X, but they also agree that he is not liable, which contradicts the rule "anyone who did X and intended to do X is liable".

One of the first papers that considers computational aspects of judgement aggregation is Endriss et al. (2010a). They investigated the computational complexity of checking whether the set of propositions (called the *agenda*) satisfies some axioms, and showed that if these axioms are satisfied, then any judgement aggregation rule that satisfies some desired properties will never produce an inconsistent outcome. Later, Endriss et al. (2010b) investigated the computational complexity of computing the collective judgement as well as deciding whether an agent can influence the outcome by misreporting her valuation of the propositions. The latest work in this line of research is by Grandi and Endriss (2011), who proposed a general framework for aggregating binary variables under constraints, and obtained a general definition of paradoxes in this framework. This framework includes combinatorial voting and judgement aggregation as special cases.

2.4 Summary

In this Chapter we recalled the voting setting, notation that is used throughout this dissertation, and definitions of some common voting rules and axiomatic properties. We also gave a very brief introduction to some other major research directions in Computational Social Choice.

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.*

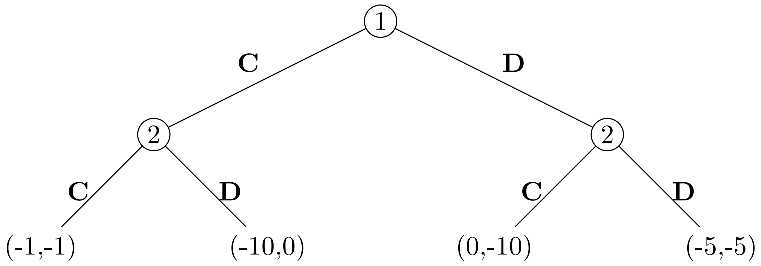


Figure 3.1: An extensive-form game.

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.

4

Computational Complexity of Unweighted Coalitional Manipulation

In this chapter, we will study the computational complexity of manipuation for three common voting rules. We will prove that the unweighted coalitional manipulation problem (Definition 3.1.2) is NP-complete for maximin (Section 4.1) and ranked pairs (Section 4.2), and we will give a polynomial-time algorithm for UCM for Bucklin (Section 4.3).

4.1 Manipulation for Maximin is NP-complete

In this section, we prove that the UCM problem for maximin is NP-complete. The proof uses a reduction from the TWO VERTEX DISJOINT PATHS IN DIRECTED ANTISYMMETRIC GRAPH problem, which is known to be NP-complete (Fortune et al., 1980).

Definition 4.1.1. *The* TWO VERTEX DISJOINT PATHS IN DIRECTED GRAPH *problem is defined as follows. We are given a directed graph G and two disjoint pairs of vertices (u, u') and (v, v'), where u, u', v, v' are all different from each other. We* *are asked whether there exist two directed paths $u \rightarrow u_1 \rightarrow \ldots \rightarrow u_{k_1} \rightarrow u'$ and $v \rightarrow v_1 \rightarrow \ldots \rightarrow v_{k_2} \rightarrow v'$ such that $u, u', u_1, \ldots, u_{k_1}, v, v', v_1, \ldots, v_{k_2}$ are all different from each other.*

Theorem 4.1.2. *The UCM problem for maximin is NP-complete, for any fixed number of manipulators $n' \geqslant 2$.*

Proof. It is easy to verify that the UCM problem for maximin is in NP. We now show that UCM is NP-hard, by giving a reduction from TWO VERTEX DISJOINT PATHS IN DIRECTED GRAPH.

Let the instance of TWO VERTEX DISJOINT PATHS IN DIRECTED GRAPH be denoted by $G = (\mathcal{V}, E)$, (u, u') and (v, v') where $\mathcal{V} = \{u, u', v, v', c_1, \ldots, c_{m-5}\}$. Without loss of generality, we assume that every vertex is reachable from u or v (otherwise, we can remove the vertex from the instance). We also assume that $(u, v') \notin E$ and $(v, u') \notin E$ (since such edges cannot be used in a solution). Let $G' = (\mathcal{V}, E \cup \{(v', u), (u', v)\})$, that is, G' is the graph obtained from G by adding (v', u) and (u', v).

We construct a UCM instance as follows.

Set of alternatives: $\mathcal{C} = \{c, u, u', v, v', c_1, \ldots, c_{m-5}\}$.

Alternative preferred by the manipulators: c.

Number of unweighted manipulators: any fixed number $n' \geqslant 2$.

Non-manipulators' profile: P^{NM} satisfying the following conditions:

1. For any $c' \neq c$, $D_{P^{NM}}(c, c') = -4n'$.
2. $D_{P^{NM}}(u, v') = D_{P^{NM}}(v, u') = -4n'$.
3. For any $(s, t) \in E$ such that $D_{P^{NM}}(t, s)$ is not defined above, we let $D_{P^{NM}}(t, s) = -2n' - 2$.

4. For any $s, t \in \mathcal{C}$ such that $D_{P^{NM}}(t, s)$ is not defined above, we let $|D_{P^{NM}}(t, s)| = 0$.

The existence of such a P^{NM}, whose size is polynomial in m, is guaranteed by Lemma 2.2.3.

We can assume without loss of generality that each manipulator ranks c first. Therefore, for any $c' \neq c$,

$$D_{P^{NM} \cup P^{M}}(c, c') = -3n' \tag{4.1}$$

We are now ready to show that there exists P^M such that Maximin$(P^{NM} \cup P^M) = c$ if and only if there exist two vertex disjoint paths from u to u' and from v to v' in G. First, we prove that if there exist such paths in G, then there exists a profile P^M for the manipulators such that Maximin$(P^{NM} \cup P^M) = c$.

Let $u \to u_1 \to \cdots \to u_{k_1} \to u'$ and $v \to v_1 \to \cdots \to v_{k_2} \to v'$ be two vertex disjoint paths. Further, let

$$\mathcal{V}' = \{u, u', v, v', u_1, \ldots, u_{k_1}, v_1, \ldots, v_{k_2}\} \quad .$$

Then, because any vertex is reachable from u or v in G, there exists a connected subgraph G^* of G' (which still includes all the vertices) in which $u \to u_1 \to \cdots \to u_{k_1} \to u' \to v \to v_1 \to \cdots \to v_{k_2} \to v' \to u$ is the only cycle. In other words, such a subgraph G^* can obtained by possibly removing some of the edges of G'. Therefore, by arranging the vertices of $\mathcal{V} \backslash \mathcal{V}'$ according to the direction of the edges of G^*, we can obtain a linear order O over $\mathcal{V} \backslash \mathcal{V}'$ with the following property: for any $t \in \mathcal{V} \backslash \mathcal{V}'$, it holds that either

1. there exists $s \in \mathcal{V} \backslash \mathcal{V}'$ such that $s \succ_O t$ and $(s, t) \in E$, or
2. there exists $s \in \mathcal{V}'$ such that $(s, t) \in E$.

We define P^M by letting $n' - 1$ manipulators vote the following.

$$c \succ u \succ u_1 \succ \cdots \succ u_{k_1} \succ u' \succ v \succ v_1 \succ \ldots \succ v_{k_2} \succ v' \succ O$$

We also let the remaining manipulator vote the following.

$$c \succ v \succ v_1 \succ \cdots \succ v_{k_2} \succ v' \succ u \succ u_1 \succ \ldots \succ u_{k_1} \succ u' \succ O$$

Then, we have the following calculations:

$$\begin{aligned} D_{P^{NM} \cup P^M}(u, v') &= -4n' + (n' - 1) - 1 \\ &= -3n' - 2 < -3n' \quad , \\ \text{and} \quad D_{P^{NM} \cup P^M}(v, u') &= -4n' + 1 - (n' - 1) \\ &= -5n' + 2 < -3n' \quad . \end{aligned}$$

Moreover, for any $t \in \mathcal{C} \backslash \{c, u, v\}$, there exists $s \in \mathcal{C} \backslash \{c\}$ such that $(s, t) \in E$ and $D_{P^M}(t, s) = -n'$, which means that

$$\begin{aligned} D_{P^{NM} \cup P^M}(t, s) &= -2n' - 2 - n' = -3n' - 2 \\ &< -3n' \end{aligned}$$

It now follows from Equation (4.1) thatMaximin$(P^{NM} \cup P^M) = c$.

Next, we prove that if there exists a profile P^M for the manipulators such that Maximin$(P^{NM} \cup P^M) = c$, then there exist two vertex disjoint paths from u to u' and from v to v'.

We define a function $f : \mathcal{V} \to \mathcal{V}$ such that $D_{P^{NM} \cup P^M}(t, f(t)) < -3n'$. Indeed, such a function exists since Maximin$(P^{NM} \cup P^M) = c$. Hence, for any $t \neq c$ there must exist s such that

$$D_{P^{NM} \cup P^M}(t, s) < -3n'$$

Moreover, s must be a parent of t in G'. If there exists more than one such s, define $f(t)$ to be any one of them.

It follows that if $(t, f(t))$ is neither (u, v') or (v, u'), then $(f(t), t) \in E$ and $D_{P^M}(t, f(t)) = -n'$, which means that $f(t) \succ t$ in each vote of P^M; otherwise, if $(t, f(t))$ is (u, v') or (v, u'), then $D_{P^M}(t, f(t)) \leqslant n' - 2$, which means that $f(t) \succ t$ in at least one vote of P^M.

Now, since $|\mathcal{V}| = m - 1$ is finite, there must exist $l_1 < l_2 \leqslant m$ such that $f^{l_1}(u) = f^{l_2}(u)$. That is,

$$f^{l_1}(u) \to f^{l_1+1}(u) \to \cdots \to f^{l_2-1}(u) \to f^{l_2}(u)$$

is a cycle in G'. We assume that for any $l_1 \leqslant l'_1 < l'_2 < l_2$, $f^{l'_1}(u) \neq f^{l'_2}(u)$. Now we claim that (v', u) and (u', v) must be both in the cycle, because

1. if neither of them is in the cycle, then in each vote of P^M, we must have

   $$f^{l_2}(u) \succ f^{l_2-1}(u) \succ f^{l_1}(u) = f^{l_2}(u) \quad ,$$

   which contradicts the assumption that each vote is a linear order;

2. if exactly one of them is in the cycle—without loss of generality, $f^{l_1}(u) = v, f^{l_1+1}(u) = u'$—then in at least one of the votes of P^M, we must have

   $$f^{l_2}(u) \succ f^{l_2-1}(u) \succ \ldots \succ f^{l_1}(u) = f^{l_2}(u) \quad ,$$

   which contradicts the assumption that each vote is a linear order.

Without loss of generality, let us assume that $f^{l_1}(u) = u, f^{l_1+1}(u) = v'$, $f^{l_3}(u) = v, f^{l_3+1}(u) = u'$, where $l_3 \leqslant l_2 - 2$. We immediately obtain two vertex disjoint paths:

$$u = f^{l_1}(u) = f^{l_2}(u) \to f^{l_2-1}(u) \to \ldots \to f^{l_3+1}(u) = u' \quad ,$$

and $v = f^{l_3}(u) \to f^{l_3-1}(u) \to \ldots \to f^{l_1+1}(u) = v'$. Therefore, UCM for maximin is NP-complete. □

Notice that the NP-completeness of UCM implies the NP-hardness of UCO for maximin.

4.2 Manipulation for Ranked Pairs is NP-complete

In this section, we prove that UCM for ranked pairs is NP-complete (even for a single manipulator) by giving a reduction from 3SAT.

Definition 4.2.1. *The* 3SAT *problem is defined as follows: Given a set of variables* $X = \{x_1, \ldots, x_q\}$ *and a formula* $Q = Q_1 \wedge \ldots \wedge Q_t$ *such that*

1. *for all* $1 \leqslant i \leqslant t$, $Q_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$, *and*
2. *for all* $1 \leqslant i \leqslant t$ *and* $1 \leqslant j \leqslant 3$, $l_{i,j}$ *is either a variable* $x \in X$, *or the negation of a variable (i.e.,*$\neg x$ *where* $x \in X$*),*

we are asked whether the variables can be set to true or false so that Q *is true.*

Theorem 4.2.2. *The UCM problem for ranked pairs is NP-complete, even when there is only one manipulator.*

Proof. It is easy to verify that UCM for ranked pairs are in NP. We first prove that UCM is NP-complete. Given an instance of 3SAT, we construct a UCM instance as follows. Without loss of generality, we assume that for any variable x, x and $\neg x$ appears in at least one clause, and none of the clauses contain both x and $\neg x$.

Set of alternatives: $\mathcal{C} = \{c, Q_1, \ldots, Q_t, Q'_1, \ldots, Q'_t\} \bigcup \{x_1, \ldots, x_q, \neg x_1, \ldots, \neg x_q\}$ $\bigcup \{Q_{l_{1,1}}, Q_{l_{1,2}}, Q_{l_{1,3}}, \ldots, Q_{l_{t,1}}, Q_{l_{t,2}}, Q_{l_{t,3}}\} \bigcup \{Q_{\neg l_{1,1}}, Q_{\neg l_{1,2}}, Q_{\neg l_{1,3}}, \ldots, Q_{\neg l_{t,1}}, Q_{\neg l_{t,2}}, Q_{\neg l_{t,3}}\}$.

Alternative preferred by the manipulator: c.

Number of unweighted manipulators: $n' = 1$.

Tie-breaking mechanism: We recall that in ranked pairs, we first use the parallel-universe tie-breaking to select multiple winners, then use a fixed-order tie-breaking mechanism the select the unique winner. In the fixed-order tie-breaking, c is ranked in the bottom position.

Non-manipulators' profile: P^{NM} satisfying the following conditions.

1. For any $i \leqslant t$, $D_{PNM}(c, Q_i) = 30$,$D_{PNM}(Q'_i, c) = 20$; for any $x \in \mathcal{C} \backslash \{Q_i, Q'_i : 1 \leqslant i \leqslant t\}$, $D_{PNM}(c, x) = 10$.

2. For any $j \leqslant q$, $D_{PNM}(x_j, \neg x_j) = 20$.

3. For any $i \leqslant t, j \leqslant 3$,

   - if $l_{i,j} = x_k$ for some $k \leqslant q$, then $D_{PNM}(Q_i, Q^i_{x_k}) = 30$, $D_{PNM}(Q^i_{x_k}, x_k) = 30$, $D_{PNM}(\neg x_k, Q^i_{\neg x_k}) = 30$, $D_{PNM}(Q^i_{\neg x_k}, Q'_i) = 30$;
   - if $l_{i,j} = \neg x_k$ for some $k \leqslant q$, then $D_{PNM}(Q_i, Q^i_{\neg x_k}) = 30$, $D_{PNM}(Q^i_{x_k}, x_k) = 30$, $D_{PNM}(\neg x_k, Q^i_{\neg x_k}) = 30$, $D_{PNM}(Q^i_{x_k}, Q'_i) = 30$, $D_{PNM}(Q^i_{\neg x_k}, Q^i_{x_k}) = 20$.

4. For any $x, y \in \mathcal{C}$, if $D_{PNM}(x, y)$ is not defined in the above steps, then $D_{PNM}(x, y) = 0$.

For example, when $Q_1 = x_1 \vee \neg x_2 \vee x_3$, D_{PNM} is illustrated in Figure 4.1.

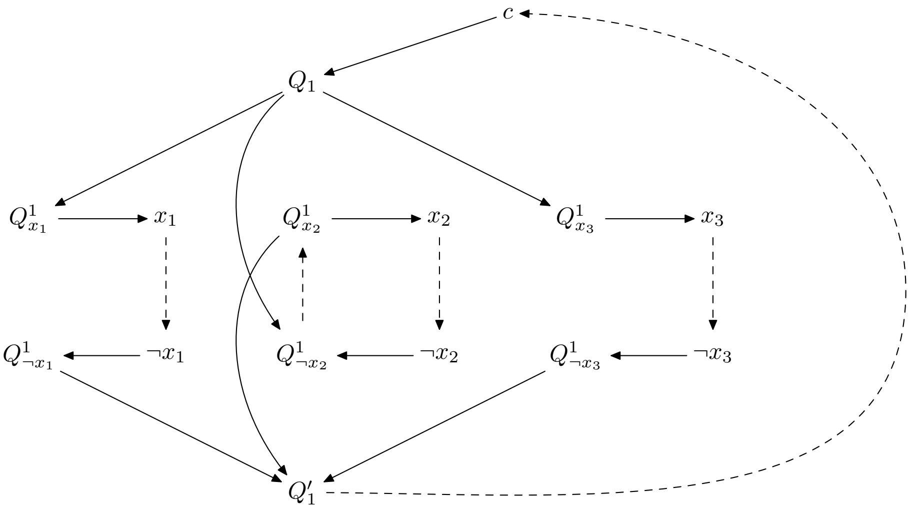


FIGURE 4.1: D_{PNM} for $Q_1 = x_1 \vee \neg x_2 \vee x_3$.

In Figure 4.1, for any vertices v_1, v_2, if there is a solid edge from v_1 to v_2, then $D_{PNM}(v_1, v_2) = 30$; if there is a dashed edge from v_1 to v_2, then $D_{PNM}(v_1, v_2) = 20$;

if there is no edge between v_1 and v_2 and $v_1 \neq c$, $v_2 \neq c$, then $D_{P^{NM}}(v_1, v_2) = 0$; for any x such that there is no edge between c and x, $D_{P^{NM}}(c, x) = 10$.

The existence of such a P^{NM} is guaranteed by Lemma 2.2.3, and the size of P^{NM} is in polynomial in t and q.

First, we prove that if there exists an assignment v of truth values to X so that Q is satisfied, then there exists a vote R_M for the manipulator such that $\text{RP}(P^{NM} \cup \{R_M\}) = c$. We construct R_M as follows.

- Let c be on the top of R_M.
- For any $k \leqslant q$, if $v(x_k) = \top$ (that is, x_k is true), then $x_k \succ_{R_M} \neg x_k$, and for any $i \leqslant t, j \leqslant 3$ such that $l_{i,j} = \neg x_k$, let $Q^i_{x_k} \succ_{R_M} Q^i_{\neg x_k}$.
- For any $k \leqslant q$, if $v(x_k) = \bot$ (that is, x_k is false), then $\neg x_k \succ_{R_M} x_k$, and for any $i \leqslant t, j \leqslant 3$ such that $l_{i,j} = \neg x_k$, let $Q^i_{\neg x_k} \succ_{R_M} Q^i_{x_k}$.
- The remaining pairs of alternatives are ranked arbitrarily.

If $x_k = \top$, then $D_{P^{NM} \cup \{R_M\}}(x_k, \neg x_k) = 21$, and for any $i \leqslant t, j \leqslant 3$ such that $l_{i,j} = \neg x_k$, $D_{P^{NM} \cup \{R_M\}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 19$. It follows that no matter how ties are broken when applying ranked pairs to $P^{NM} \cup \{R_M\}$, if $x_k = \top$, then $x_k \succ \neg x_k$ in the final ranking. This is because for any $l_{i,j} = \neg x_k$, $D_{P^{NM} \cup \{R_M\}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 19 < 21 = D_{P^{NM} \cup \{R_M\}}(x_k, \neg x_k)$, which means that before trying to fix $x_k \succ \neg x_k$, there is no directed path from $\neg x_k$ to x_k.

Similarly if $x_k = \bot$, then $D_{P^{NM} \cup \{R_M\}}(x_k, \neg x_k) = 19$, and for any $i \leqslant t, j \leqslant 3$ such that $l_{i,j} = \neg x_k$, $D_{P^{NM} \cup \{R_M\}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 21$. It follows that if $x_k = \bot$, then $\neg x_k \succ x_k$, and for any $i \leqslant t, j \leqslant 3$ such that $l_{i,j} = \neg x_k$, $Q^i_{\neg x_k} \succ Q^i_{x_k}$ in the final ranking. This is because $Q^i_{\neg x_k} \succ Q^i_{x_k}$ will be fixed before $x_k \succ \neg x_k$.

Because Q is satisfied under v, for each clause Q_i, at least one of its three literals is true under v. Without loss of generality, we assume $v(l_{i,1}) = \top$. If $l_{i,1} = x_k$, then before trying to add $Q'_i \succ c$, the directed path $c \to Q_i \to Q_{x_k} \to x_k \to \neg x_k \to Q_{\neg x_k} \to Q'_i$ has already been fixed. Therefore, $c \succ Q'_i$ in the final ranking, which means that for any alternatives x in $\mathcal{C} \backslash \{c, Q_1, \ldots, Q_t, Q'_1, \ldots, Q'_t\}$, $c \succ x$ in the final ranking because $D_{P^{NM} \cup \{R_M\}}(c, x) > 0$. Hence, c is the unique winner of $P^{NM} \cup \{R_M\}$ under ranked pairs.

Next, we prove that if there exists a vote R_M for the manipulator such that $\mathrm{RP}(P^{NM} \cup \{R_M\}) = c$, then there exists an assignment v of truth values to X such that Q is satisfied. We construct the assignment v so that $v(x_k) = \top$ if and only if $x_k \succ_{R_M} \neg x_k$, and $v(x_k) = \bot$ if and only if $\neg x_k \succ_{R_M} x_k$. We claim that $v(Q) = \top$. If, on the contrary, $v(Q) = \bot$, then there exists a clause (Q_1, without loss of generality) such that $v(Q_1) = \bot$. We now construct a way to fix the pairwise rankings such that c is not the winner for ranked pairs, as follows. For any $j \leqslant 3$, if there exists $k \leqslant q$ such that $l_{i,j} = \neg x_k$, then $x_k \succ_{R_M} \neg x_k$ because $v(\neg x_k) = \bot$. Therefore, $D_{P^{NM} \cup R_M}(x_k, \neg x_k) = 21$. Then, after trying to add all pairs $x \succ x'$ such that $D_{P^{NM} \cup R_M}(x, x') > 21$ (that is, all solid directed edges in Figure 4.1), it follows that $x_k \succ \neg x_k$ can be added to the final ranking. We choose to add $x_k \succ \neg x_k$ first, which means that $Q^1_{x_k} \succ Q^1_{\neg x_k}$ in the final ranking (otherwise, we have $Q^1_{\neg x_k} \succ Q^1_{x_k} \succ x_k \succ \neg x_k \succ Q^1_{\neg x_k}$, which is a contradiction).

For any $j \leqslant 3$, if there exists $k \leqslant q$ such that $l_{i,j} = x_k$, then $\neg x_k \succ_{R_M} x_k$ because $v(x_k) = \bot$. Therefore, $D_{P^{NM} \cup R_M}(x_k, \neg x_k) = 19$. We note that after trying to add all pairs $x \succ x'$ such that $D_{P^{NM} \cup R_M}(x, x') > 19$, $Q^1_{x_k} \not\succ Q^1_{\neg x_k}$. We recall that for any $j \leqslant 3$, if there exists $k \leqslant q$ such that $l_{i,j} = \neg x_k$, then $Q^1_{\neg x_k} \not\succ Q^1_{x_k}$. Hence, it follows that $Q'_1 \succ c$ is consistent with all pairwise rankings added so far. Then, since $D_{P^{NM} \cup R_M}(Q'_1, c) \geqslant 19$, if $Q'_1 \succ c$ has not been added, we choose to add it first of all pairwise rankings of alternatives $x \succ x'$ such that $D_{P^{NM} \cup R_M}(x, x') = 19$, which means that $Q'_1 \succ c$ in the final ranking—in other words, c is not at the top in the final ranking. Therefore, c is not the unique winner, which contradicts the assumption that $\mathrm{RP}(P^{NM} \cup \{R_M\}) = c$. □

Similarly, we can prove that when k is a constant greater than one, UCM for ranked pairs remain NP-complete.

Theorem 4.2.3. *The UCM problem for ranked pairs is NP-complete, for any fixed number of manipulators $n' \geqslant 2$.*

Proof. The proof is similar to that of Theorem 4.2.2. We let P^{NM} satisfy the following conditions.

1. For any $i \leqslant t$, $D_{P^{NM}}(c, Q_i) = 30n'$,$D_{P^{NM}}(Q'_i, c) = 22n' - 2$; for any $x \in \mathcal{C} \backslash \{Q_i, Q'_i : 1 \leqslant i \leqslant t\}$, $D_{P^{NM}}(c, x) = 10n'$.

2. For any $j \leqslant q$, $D_{P^{NM}}(x_j, \neg x_j) = 22n' - 2$.

3. For any $i \leqslant t, j \leqslant 3$, if $l_{i,j} = x$, then $D_{P^{NM}}(Q_i, Q^i_x) = 30n'$, $D_{P^{NM}}(Q^i_x, x) = 30n'$, $D_{P^{NM}}(\neg x, Q^i_{\neg x}) = 30n'$, $D_{P^{NM}}(Q^i_{\neg x}, Q'_i) = 30n'$; if $l_{i,j} = \neg x$, then $D_{P^{NM}}(Q_i, Q^i_{\neg x}) = 30n'$, $D_{P^{NM}}(Q^i_x, x) = 30n'$, $D_{P^{NM}}(\neg x, Q^i_{\neg x}) = 30n'$, $D_{P^{NM}}(Q^i_x, Q'_i) = 30n'$, $D_{P^{NM}}(Q^i_{\neg x}, Q^i_x) = 20n'$.

4. For any $y, y' \in \mathcal{C}$, if $D_{P^{NM}}(y, y')$ is not defined in the above steps, then $D_{P^{NM}}(y, y') = 0$.

First, if there exists an assignment v of truth values to X so that Q is satisfied, then we let R_M be defined as in the proof for Theorem 4.2.2. It follows that $\mathrm{RP}(P^{NM} \cup \{n'R_M\}) = c$ (all the manipulators can vote R_M).

Next, if there exists a profile P^M for the manipulators such that $\mathrm{RP}(P^{NM} \cup P^M) = c$, then we construct the assignment v so that $v(x) = \top$ if $x \succ_V \neg x$ for all $V \in P^M$, and $v(x) = \bot$ if $\neg x \succ_V x$ for all $V \in P^M$; the values of all the other variables are assigned arbitrarily. Then by similar reasoning as in the proof for Theorem 4.2.2, we know that Q is satisfied under v. □

4.3 A Polynomial-time Algorithm for Manipulation for Bucklin

In this section, we present a polynomial-time algorithm for the UCM problem for Bucklin.

For any alternative $x \in \mathcal{C}$, any natural number $d \in \mathbb{N}$, and any profile P, let $B(x, d, P)$ denote the number of times that x is ranked among the top d alternatives in P. The idea behind the algorithm is as follows. Let d_{min} denote the Bucklin score of x in P, that is, d_{min} is the minimal depth so that the favorite alternative c is ranked among the top d_{min} alternatives in more than half of the votes (when all of the manipulators rank c first). Then, we simply check if there is a way to assign the manipulators' votes so that none of the other alternatives is ranked among the top d_{min} alternatives in more than half of the votes. In other words, the order of the alternatives is not crucial, only their membership in the set of d_{min} top-ranked alternatives is relevant.

Algorithm 1.

Input. A UCM instance $(Bucklin, P^{NM}, c, n')$, where $\mathcal{C} = \{c, c_1, \dots, c_{m-1}\}$.[1]

Stage 0.

0.1 Calculate the Bucklin score d_{min} such that

$$B(c, d_{min}, P^{NM}) + n' > \frac{1}{2}(|P^{NM}| + n')$$

0.2 If there exists $c' \in C$, $c' \neq c$ such that

$$B(c', d_{min}, P^{NM}) > \frac{1}{2}(|P^{NM}| + n') \quad , \tag{4.2}$$

then output that there is no successful manipulation.

[1] This algorithm works for the fixed-order tie-breaking mechanism where c is ranked in the bottom position. Similar algorithms can be designed for other fixed-order tie-breaking mechanisms.

Aside. Notice that d_{min} is defined under the assumption that all the manipulators rank c first. Consider an alternative $c' \neq c$ that satisfies the condition in Equation (4.2). Such an alternative is ranked in the top d_{min} positions of half the votes $P^{NM} \cup P^{M}$, regardless of P^{M}. Hence, c cannot be a unique winner.

Stage 1.

1.1 For every $c' \in C \backslash \{c\}$, let

$$d_{c'} = \left\lfloor \frac{1}{2}(|P^{NM}| + n') \right\rfloor - B(c', d_{min}, P^{NM}) \quad ,$$

and let $k_{c'} = \min\{d_{c'}, n'\}$.

1.2 If

$$\sum_{c' \neq c} k_{c'} < (d_{min} - 1)n' \quad , \tag{4.3}$$

then output that there is no successful manipulation.

Aside. $k_{c'}$ is the number of times that we can place c' in the first d_{min} positions of the votes of P^{M}, without compromising the victory of c. In particular, $k_{c'}$ cannot be greater than n'.

Notice that there are exactly $(d_{min} - 1)n'$ problematic positions to fill, since c is ranked first by all the manipulators. Now, if the condition in Equation (4.3) is satisfied, for any P^{M} there must be an alternative c' that appears too many times in the first d_{min} positions, that is, $k_{c'} < B(c', d_{min}, P^{M})$. Since $B(c', d_{min}, P^{M}) \leqslant n'$, we have in particular that $k_{c'} < n'$, hence it must hold that $k_{c'} = d_{c'}$. It follows that

$$\begin{aligned} & B(c', d_{min}, P^{NM} \cup P^{M}) \\ =& B(c', d_{min}, P^{NM}) + B(c', d_{min}, P^{M}) \\ >& B(c', d_{min}, P^{NM}) + d_{c'} \\ =& \left\lfloor \frac{1}{2}(|P^{NM}| + n') \right\rfloor \end{aligned}$$

Therefore, c cannot be a unique winner.

Stage 2. Construct P^M by assigning the alternatives to the first d_{min} positions of the votes in a way that for every $t = 1, \dots, m-1$,

$$B(c_t, d_{min}, P^M) \leqslant k_{c_t} \tag{4.4}$$

Complete the rest of the votes arbitrarily. Return P^M as a successful manipulation.

Aside. Given that (4.3) does not hold, it is clearly possible to construct P^M such that (4.4) holds for every $c' \neq c$. Moreover, this can be done in polynomial time, e.g., by enumerating the alternatives and placing each alternative in the next position in $k_{c'}$ of the votes of the manipulators, until the crucial positions are filled.

Now, for every $t = 1, \dots, m-1$ it holds that

$$\begin{aligned} B(c_t, d_{min}, P^{NM} \cup P^M) &\leqslant B(c_t, d_{min}, P^{NM}) + k_{c_t} \\ &\leqslant \frac{1}{2}(|NM| + n') \quad , \end{aligned}$$

which implies that $Bucklin(P^{NM} \cup P^M) = c$.

We have obtained the following result.

Theorem 4.3.1. *Algorithm 1 correctly decides the UCM problem in polynomial time.*

It is easy to see that the tractability of UCM for Bucklin implies that UCO can be solved in polynomial time as well.

4.4 Summary

In this chapter, we investigated the computational complexity of the UCM and UCO problems for the maximin, ranked pairs, and Bucklin rules. The UCM problem is NP-complete under the maximin rule for any fixed number (at least two) of manipulators. The UCM problem is also NP-complete under the ranked pairs rule; in this case, the hardness holds even if there is only a single manipulator, similarly to second-order Copeland (Bartholdi et al., 1989a) and STV (Bartholdi and Orlin, 1991). Finally, we gave a polynomial-time algorithm for the UCM problem under the Bucklin rule.

It should be noted that all of our NP-hardness results, as well as the ones mentioned in the introduction, are *worst-case* results. Hence, there may still be an efficient algorithm that can find a beneficial manipulation for *most* instances. Indeed, nearly a dozen recent papers suggest that finding manipulations is easy with respect to some typical distributions on preference profiles. We will see some of them in the next chapter.

5

Computing Manipulations is "Usually" Easy

We have seen in the last chapter that computing a manipulation is NP-complete for maximin and ranked pairs. In particular, the coalitional manipulation problem is NP-complete for ranked pairs even for one manipulator. This property was only shown previously for STV and second-order Copeland. In Table 3.1 we observe that computational complexity can serve as a barrier for many common voting rules when there are two or more manipulators. In this chapter, we will prove that computational complexity is not a strong barrier against manipulation for almost all common voting rules. This argument will be supported by two approaches. In Section 5.2 we pursue the "frequency of manipulability" approach, that is, the votes are randomly generated i.i.d. according to some distribution over all linear orders. We will show that with a high probability the UCM problem (Definition 3.1.2) is computationally trivial. In Section 5.4 we pursue an approximation approach. More precisely, we focus on approximating the UCO problem (Definition 3.1.3), and propose an algorithm that approximates UCO with an additive error that only depends on the number of alternatives (but not on the number of voters) for all positional scoring rules.

Instead of proving the results one by one for common voting rules, we take unified approaches. In Section 5.1 we introduce a general framework called *generalized scoring rules*, and then characterize the frequency of manipulability for any generalized scoring rule in Section 5.2. To show how general this class of voting rules is, we give a concise axiomatic axiomatization in Section 5.3. In Section 5.4 we will design an approximation algorithm that works for any positional scoring rule, in light of a novel relationship between UCO and a scheduling problem.

5.1 Generalized Scoring Rules

A generalized scoring rule (GSR) associates a vector of k real numbers with every vote, for some k that depends on (but is not necessarily equal to) m. The decision that the rule makes is based only on the sum of these vectors. Even more specifically, the decision is based only on comparisons among the components in this sum. That is, if we know, for every $i, j \in \{1, \ldots, k\}$, whether the ith component in the sum is larger than the jth component, the jth is larger than the ith, or they are the same, then we know enough to determine the winner. Sometimes, the components can be partitioned so that the decision only depends on comparisons within elements of the partition, which will be helpful.

Let $k \in \mathbb{N}$, and let $\mathscr{K} = \{K_1, \ldots, K_q\}$ be a partition of $K = \{1, \ldots, k\}$. That is, for any $i \leqslant q$, $K_i \subseteq K$, $K = \bigcup_{l=1}^{q} K_l$, and for any $i, j \leqslant q$, $i \neq j$, $K_i \cap K_j = \emptyset$. We say that two vectors of length k are equivalent with respect to a partition if, within each element of the partition, they agree on which components are larger.

Definition 5.1.1. *Let $\mathscr{K}$ be a partition of K. For any $a, b \in \mathbb{R}^k$, we say that a and b are* equivalent with respect to $\mathscr{K}$*, denoted by $a \sim_{\mathscr{K}} b$, if for any $l \leqslant q$, any $i, j \in K_l$, $a_i \geqslant a_j \Leftrightarrow b_i \geqslant b_j$ (where a_i denotes the ith component of the vector a,* etc.*).*

For two partitions $\mathscr{K} = \{K_1, \ldots, K_q\}$ and $\mathscr{K}' = \{K'_1, \ldots, K'_p\}$, $\mathscr{K}'$ is a *refinement* of $\mathscr{K}$ if for any $l \leqslant q$, any $l' \leqslant p$, $K'_{l'} \cap K_l$ is either $K'_{l'}$ or $\emptyset$. That is, $\mathscr{K}'$

is obtained from $\mathscr{K}$ by partitioning the sets in $\mathscr{K}$. In this case, we say that $\mathscr{K}$ is *coarser* than $\mathscr{K}'$, and $\mathscr{K}'$ is *finer* than $\mathscr{K}$.

Proposition 5.1.2. *For any partitions $\mathscr{K}$, $\mathscr{K}'$ such that $\mathscr{K}'$ is a refinement of $\mathscr{K}$, and any $a, b \in \mathbb{R}^k$, if $a \sim_{\mathscr{K}} b$, then $a \sim_{\mathscr{K}'} b$.*

We note that $\{K\}$ (the partition that only contains K itself) is the coarsest partition.

Definition 5.1.3. *Let $\mathscr{K}$ be a partition of K. A function $g : \mathbb{R}^k \to \mathcal{C}$ is* compatible with $\mathscr{K}$ *if for any $a, b \in \mathbb{R}^k$, $a \sim_{\mathscr{K}} b \Rightarrow g(a) = g(b)$.*

That is, for any mapping g that is compatible with $\mathscr{K}$, $g(a)$ is determined (only) by comparisons within each K_l, $l \leqslant q$. Namely, we do not need to compare components across different elements of the partition.

Now we are ready to define generalized scoring rules.

Definition 5.1.4. *Let $k \in \mathbb{N}$, $f : L(\mathcal{C}) \to \mathbb{R}^k$ and $g : \mathbb{R}^k \to \mathcal{C}$, where g is compatible with $\mathscr{K}$. f and g determine the* generalized scoring rule *$GS(f, g)$ as follows. For any profile of votes $V_1, \ldots, V_n \in L(\mathcal{C})$, $GS(f, g)(V_1, \ldots, V_n) = g(\sum_{i=1}^n f(V_i))$. We say that $GS(f, g)$ is* of order k, *and* compatible with $\mathscr{K}$.

From Proposition 5.1.2 we know that for any partitions $\mathscr{K}$, $\mathscr{K}'$ such that $\mathscr{K}'$ is a refinement of $\mathscr{K}$, $GS(f, g)$ is compatible with $\mathscr{K}'$, then $GS(f, g)$ is also compatible with $\mathscr{K}$. Given a profile P of votes, we use $f(P)$ as shorthand for $\sum_{V \in P} f(V)$. We will call $f(P)$ the *total generalized score vector*. By definition, any unweighted generalized scoring rule satisfies *anonymity* (that is, every voter is treated equally) and *homogeneity* (that is, if we add any number of copies of the profile to the profile, the winner does not change). Any generalized scoring rule is compatible with the partition $\{K\}$. Nevertheless, being compatible with $\{K\}$ is not vacuous: if we modified the definition so that g is not required to be compatible with any partition, then any anonymous voting rule would belong to the resulting class of rules. If a generalized scoring rule is compatible with a partition, this effectively means that, within each element of the partition, the scores are of the same "type," so that we can compare them.

We now illustrate how general the class of generalized voting rules is by showing how some standard rules belong to the class. Many other rules can also be shown to belong to the class.

Proposition 5.1.5. *All positional scoring rules, Copeland, STV, maximin, ranked pairs, and Bucklin are generalized scoring rules.*

Proof of Proposition 5.1.5: We explicitly give $k, f, g, \mathscr{K}$ for each of these rules. In the remainder of the proof, the number of alternatives is fixed to be m. Let $V \in L(\mathcal{C})$ be a vote, and let P be a profile of votes. To simplify the construction, we will not specify how ties are broken when we describe these rules as generalized scoring rules. It is easy to incorporate the tie-breaking mechanism to define the g function for all these voting rules.

Positional scoring rules: Suppose the scoring vector for the rule is $\vec{s}_m = (s_m(1), \ldots, s_m(m))$. The total generalized score vector will simply consist of the total scores of the individual alternatives. Let

- $k_{\vec{s}_m} = m$.
- $f_{\vec{s}_m}(V) = (\vec{s}_m(V, c_1), \ldots, \vec{s}_m(V, c_m))$.
- $g_{\vec{s}_m}(f_{\vec{v}}(P)) = \arg\max_i (f_{\vec{v}}(P))_i$.
- $\mathscr{K}_{\vec{s}_m} = \{K\}$.

Copeland: For Copeland, the total generalized score vector will consist of the scores in the pairwise elections. Let

- $k_{Copeland} = m(m-1)$; the components are indexed by pairs (i,j) such that $i,j \leqslant m$, $i \neq j$.

- $(f_{Copeland}(V))_{(i,j)} = \begin{cases} 1 & \text{if } c_i \succ_V c_j \\ 0 & \text{otherwise} \end{cases}$

- $g_{Copeland}$ selects the winner based on $f_{Copeland}(P)$ as follows. For each pair $i \neq j$, if $(f_{Copeland}(P))_{(i,j)} > (f_{Copeland}(P))_{(j,i)}$, then add 1 point to i's Copeland score; if $(f_{Copeland}(P))_{(j,i)} > (f_{Copeland}(P))_{(i,j)}$, then add 1 point to j's Copeland score; if tied, then add 0.5 to both i's and j's Copeland scores. The winner is the alternative that gets the highest Copeland score.

- $q_{Copeland} = \frac{m(m-1)}{2}$ (we recall that q is the number of elements in the partition). The elements of the partition are indexed by (i,j), $i < j$. For any $l = (i,j)$, $i < j$, let $K_l = \{(i,j),(j,i)\}$. Let $\mathscr{K}_{Copeland} = \{K_l : l = (i,j), i < j\}$.

STV: For STV, we will use a total generalized score vector with many components. For every proper subset S of alternatives, for every alternative c outside of S, there is a component in the vector that contains the number of times that c is ranked first if all of the alternatives in S are removed. Let

- $k_{STV} = \sum_{i=0}^{m-1} \binom{m}{i}(m-i)$; the components are indexed by (S,j), where S is a proper subset of $\mathcal{C}$ and $j \leqslant m, c_j \notin S$.

- $(f_{STV}(V))_{(S,j)} = 1$, if after removing S from V, c_j is at the top; otherwise, let $(f_{STV}(V))_{(S,j)} = 0$.

- g_{STV} selects the winner based on $f_{STV}(P)$ as follows. In the first round, find $j_1 = \arg\min_j((f_{STV}(P))_{(\varnothing,j)})$. Let $S_1 = \{c_{j_1}\}$. Then, for any $2 \leqslant i \leqslant m-1$, define S_i recursively as follows: $S_i = S_{i-1} \cup \{j_i\}$, where $j_i = \arg\min_j(f_{STV}(P)_{(S_{i-1},j)})$; finally, the winner is the unique alternative in $\mathcal{C} - S_{m-1}$.

- $q_{STV} = 2^m - 1$. The elements of the partition are indexed by the $S \subset \mathcal{C}$. For any $S \subset \mathcal{C}$, let $K_S = \{(S, j) : c_j \notin S\}$. Let $\mathscr{K}_{STV} = \{K_S : S \subset \mathcal{C}\}$.

Maximin: For maximin, we use the same total generalized score vector as for Copeland, that is, the vector of all scores in pairwise elections. Let

- $k_{maximin} = m(m-1)$; the components are indexed by pairs (i, j) such that $i, j \leqslant m$, $i \neq j$.

- $(f_{maximin}(V))_{(i,j)} = \begin{cases} 1 & \text{if } c_i \succ_V c_j \\ 0 & \text{otherwise} \end{cases}$

- $g_{maximin}(f_{maximin}(P))$ is the c_i such that for any $i' \leqslant m$, $i' \neq i$, there exists $j' < m$, $j' \neq i'$ such that for any $j \leqslant m$, $j \neq i$, we have $f_{maximin}(P)_{(i,j)} > (f_{maximin}(P))_{(i',j')}$.

- $\mathscr{K}_{maximin} = \{K\}$.

Ranked pairs: We use the same total generalized score vector as for Copeland and maximin, that is, the vector of all scores in pairwise elections. Let

- $k_{rp} = m(m-1)$; the components are indexed by pairs (i, j) such that $i, j \leqslant m$, $i \neq j$.

- $(f_{rp}(V))_{(i,j)} = \begin{cases} 1 & \text{if } c_i \succ_V c_j \\ 0 & \text{otherwise} \end{cases}$

- g_{rp} selects the winner based on $f_{rp}(P)$ as follows. In each step, we consider a pair of alternatives c_i, c_j that we have not previously considered; specifically, we choose the remaining pair with the highest $(f_{rp}(P))_{(i,j)}$. We then fix the order $c_i > c_j$, unless this contradicts previous orders that we fixed (that is, it violates transitivity). We continue until we have considered all pairs of alternatives. The alternative at the top of the ranking wins.

- $\mathscr{K}_{rp} = \{K\}$.

Bucklin: For Bucklin, the total generalized score vector will have one component for every combination of an alternative and a position; this component contains the number of times that that alternative is ranked either in that position or in a higher position. We only need to consider positions from 1 through $m-1$. Let

- $k_{Bucklin} = 2m(m-1)$; the components are indexed by $(i,j)^1$ and $(i,j)^2$, $i \leqslant m-1$, $j \leqslant m$.
- $f_{Bucklin}(V)_{(i,j)^1} = 1$ and $f_{Bucklin}(V)_{(i,j)^2} = 0$ if c_j is ranked among the top i alternatives in V; otherwise $f_{Bucklin}(V)_{(i,j)^1} = 0$ and $f_{Bucklin}(V)_{(i,j)^2} = 1$.
- $g_{Bucklin}(f_{Bucklin}(P))$ is the c_j such that there exists $i \leqslant m$, $(i,j)^1 > (i,j)^2$, and for any $j' \neq j$, $(i,j')^2 \geqslant (i,j)^1$.
- $q_{Bucklin} = m-1$. For any $l \leqslant m-1$, let $K_l = \{(l,j)^1, (l,j)^2 : j \leqslant m\}$. Let $\mathscr{K}_{Bucklin} = \{K_l : l \leqslant m\}$.

□

We have shown that STV is a generalized scoring rule in the proof. In fact, we can generalize this and show that *any* multiround run-off process where in each round, alternatives are eliminated according to a generalized scoring rule (to be precise, a correspondence) must itself be a generalized scoring rule. For example, for STV, the voting rule that only eliminates one alternative (the alternative that has the lowest plurality score among all remaining alternatives) is used in every round. As another example, for Baldwin's rule, a reverse version of Borda that only eliminates one alternative (the alternative that has the lowest Borda score among all remaining alternatives) is used in every round. The proof can be found in the appendix of Xia and Conitzer (2008b), and is omitted here.

5.2 Frequency of Manipulability for Generalized Scoring Rules

Let π be a probability distribution over $L(\mathcal{C})$ that is positive everywhere. For any $n^* \in \mathbb{N}$, let ϕ_{π,n^*} be the distribution over profiles of n^* voters in which each vote is drawn i.i.d. according to π. Given a manipulation instance (r, P^{NM}, c, n'), if there is only one possible winner, then we say that this manipulation instance is *closed*; otherwise we say this manipulation instance is *open* (Procaccia and Rosenschein, 2007a).

Definition 5.2.1. *A manipulation instance* (r, P^{NM}, c, n') *is* closed *if for any profiles* P_1^M, P_2^M *for the manipulators,* $r(P^{NM} \cup P_1^M) = r(P^{NM} \cup P_2^M)$*. An instance is* open *if it is not closed.*

We note that in the above definition, whether a UCM instance is open or closed does not depend on the choice of c. That is, for any $c, c' \in \mathcal{C}$, (r, P^{NM}, c, n') is open (respectively, closed) if and only if (r, P^{NM}, c', n') is open (respectively, closed). Procaccia and Rosenschein (2007a) have shown that, suppose the following four conditions are satisfied.

1. The rule is a positional scoring rule,
2. the number of manipulators $|M|$ is $o(\sqrt{n})$,
3. the votes are drawn independently, and
4. there exists $d > 0$ such that for each vote's distribution, the variance of the difference in scores for any pair of alternatives is at least d.

Then, when $n \to \infty$, the probability that a weighted manipulation instance is open is 0. In this section, we generalize this result to generalized scoring rules; in addition, we characterize the rate of convergence to 0. However, unlike Procaccia and

Rosenschein, we do assume that votes are drawn i.i.d.; this is needed to obtain the convergence rate. Hence, strictly speaking, our result is not a generalization of their result. We can also obtain a strict generalization of Procaccia and Rosenschein's results to generalized scoring rules, but without proving a convergence rate; we will not do so in this paper.

5.2.1 Conditions under Which Coalitional Manipulability is Rare

In this section, we study the probability that a manipulation instance is open when there are $O(n^\alpha)$ $(0 \leqslant \alpha < \frac{1}{2})$ manipulators, and the nonmanipulator votes are drawn i.i.d. Let $n^* = |P^{NM}|$ denote the number of nonmanipulators. Then, n is the total number of voters, $n^* + n'$ (nonmanipulators and manipulators). We will prove that for any generalized scoring rule, this probability is $O(\frac{1}{\sqrt{n}})$. Let $T(r, m, n, \pi, n')$ denote this probability. That is, let c be an arbitrary alternative,

$$T(r, m, n, \pi, n') = Pr_{P^{NM} \sim \phi_{\pi,n^*}}\{(r, P^{NM}, c, n') \text{ is open}\}$$

Lemma 5.2.2. *Let $N \in \mathbb{N}$. Let $Y_1, \ldots, Y_N$ be i.i.d. random variables with $E(Y_1) < \infty$, $E((Y_1 - E(Y_1))^2) > 0$, and $E(|Y_1 - E(Y_1)|^3) < \infty$. Let $Y = \sum_{\zeta=1}^{N} Y_\zeta$. For any constant $0 \leqslant p < \frac{1}{2}$ that does not depend on N, and any function $f(N)$ that is $\Omega(1)$, we have that $Pr(|Y| \leqslant f(N))$ is $O(\frac{f(N)}{\sqrt{N}})$.*

Proof of Lemma 5.2.2: Let $\Phi(x)$ be the cumulative distribution function of the standard normal distribution $N(0, 1)$. Let $\sigma^2 = E((Y_1 - E(Y_1))^2)$, $\rho = E(|Y_1 - E(Y_1)|^3)$. Then we have:

$$\begin{aligned} &Pr(|Y| < f(N)) \\ =&Pr(-\frac{E(Y_1)N}{\sigma\sqrt{N}} - \frac{f(N)}{\sigma\sqrt{N}} < \frac{Y - E(Y_1)N}{\sigma\sqrt{N}} < -\frac{E(Y_1)N}{\sigma\sqrt{N}} + \frac{f(N)}{\sigma\sqrt{N}}) \end{aligned}$$

Then by the Berry-Esséen theorem (Durrett, 1991),

$$\begin{aligned}
&Pr(|Y| < f(N)) \\
<&\Phi(-\frac{E(Y_1)N}{\sigma\sqrt{N}} + \frac{f(N)}{\sigma\sqrt{N}}) - \Phi(-\frac{E(Y_1)N}{\sigma\sqrt{N}} - \frac{f(N)}{\sigma\sqrt{N}}) + \frac{C\rho}{\sigma^3\sqrt{N}} \\
=&\int_{-\frac{E(Y_1)N}{\sigma\sqrt{N}}-\frac{f(N)}{\sigma\sqrt{N}}}^{-\frac{E(Y_1)N}{\sigma\sqrt{N}}+\frac{f(N)}{\sigma\sqrt{N}}} N(0,1)(x)dx + \frac{C\rho}{\sigma^3\sqrt{N}} \\
<&\frac{2f(N)}{\sigma\sqrt{N}} \times \frac{1}{\sqrt{2\pi}} + \frac{C\rho}{\sigma^3\sqrt{N}}
\end{aligned}$$

which is $O(\frac{f(N)}{\sqrt{N}})$, because C is a constant that does not depend on N and $f(N) = \Omega(1)$. □

Theorem 5.2.3. *Let $r = GS(f, g)$ be a generalized scoring rule of order k. For any $m \in \mathbb{N}$, any constant $0 \leqslant \alpha < \frac{1}{2}$, and any constant h (where both m and h do not depend on n), there exists a constant $t_{m,\alpha,h} > 0$ (that does not depend on n) such that if $n' \leqslant hn^{\alpha}$, then*

$$T(r, m, n, \pi, n') \leqslant t_{m,\alpha,h} n^{\alpha-\frac{1}{2}}$$

Proof of Theorem 5.2.3: We recall that each vote is drawn i.i.d. according to π. For any pair $i_1, i_2 \leqslant k$, $i_1 \neq j_2$, and any $t > 0$, let

$$R(i_1, i_2, t, \pi, n') = Pr\{|(f(P^{NM}))_{i_1} - (f(P^{NM}))_{i_2}| \leqslant t\}$$

We recall that $(f(P^{NM}))_i$ is the ith component of $f(P^{NM})$. In other words, $R(i_1, i_2, t, \pi, n')$ is the probability of profiles of nonmanipulators' votes P^{NM} such that the difference between the i_1th component and the i_2th component of $f(P^{NM})$ is no more than t, when each vote is drawn i.i.d. according to π. We also recall that $n^* = |P^{NM}|$. Let $Y_1^{i_1,i_2}, \ldots, Y_{n^*}^{i_1,i_2}$ be n^* i.i.d. random variables, where the distribution for each $Y_\zeta^{i_1,i_2}$ is the same as the distribution for $(f(V))_{i_1} - (f(V))_{i_2}$, where V is drawn according to π. That is, for any $V \in L(\mathcal{C})$, with probability $\pi(V)$, $Y_1^{i_1,i_2}$ takes value $(f(V))_{i_1} - (f(V))_{i_2}$. Let $Y^{i_1,i_2} = \sum_{\zeta=1}^{n^*} Y_\zeta$.

Let $v_{max} = \max\limits_{i \leqslant k, V \in L(\mathcal{C})} (f(V))_i$. That is, v_{max} is the maximum component of all score vectors corresponding to a single vote. We note that v_{max} is a constant that does not depend on n. We also note that since n' is $O(n^\alpha)$ and $\alpha < \frac{1}{2}$, it must be that n^* is $\Omega(n)$, so that n is $O(n^*)$, $v_{max}hn^\alpha$ is $O((n^*)^\alpha)$. Therefore, by Lemma 5.2.2 (in which we let $N = n^*$), we know that $Pr(|Y^{i_1,i_2}| \leqslant v_{max}hn^\alpha)$ is $O(\frac{v_{max}hn^\alpha}{\sqrt{n^*}}) = O((n^*)^{\alpha - \frac{1}{2}})$, so it is $O(n^{\alpha - \frac{1}{2}})$. Hence, there exists a constant t_{i_1,i_2} such that

$$Pr(|Y^{i_1,i_2}| \leqslant v_{max}hn^\alpha) < t_{i_1,i_2} n^{\alpha - \frac{1}{2}}$$

We let $t_{max} = \max_{i,j \leqslant k, i \neq j} t_{i,j}$. If a manipulation instance is open, then there exists a profile P^M for the manipulators such that $GS(f,g)(P^M \cup P^{NM}) \neq GS(f,g)(P^{NM})$, which means that $f(P^M \cup P^{NM}) \nsim f(P^{NM})$. In this case there must exist i, j, $i \neq j$, such that $Pr(|(f(P^{NM}))_i - (f(P^{NM}))_j| \leqslant v_{max}n^*) \leqslant v_{max}hn^\alpha$. Therefore,

$$T(GS(f,g), m, n, \pi, n') \leqslant \sum_{1 \leqslant i < j \leqslant m} R(i, j, v_{max}hn^\alpha, \pi, n')$$

We note that $R(i, j, v_{max}hn^\alpha, \pi, n') = Pr(|Y^{i,j}| \leqslant v_{max}hn^\alpha)$. Therefore, we have

$$T(GS(f,g), m, n, \pi, n') \leqslant \sum_{i \neq j} R(i, j, v_{max}hn^\alpha, \pi, n')$$

$$\leqslant \sum_{i \neq j} t_{i,j} n^{\alpha - 1} \leqslant \frac{k(k-1)}{2} t_{max} n^{\alpha - \frac{1}{2}}$$

Let $t_{m,\alpha,h} = \frac{k(k-1)}{2} t_{max}$. We know that $t_{m,\alpha,h}$ is a constant that does not depend on n.

(End of the proof for Theorem 5.2.3.) □

From Proposition 5.1.5 and Theorem 5.2.3, we obtain the following corollary.

Corollary 5.2.4. *Let r be any positional scoring rule, Copeland, STV, maximin, ranked pairs, or Bucklin. For any $m \in \mathbb{N}$, any constant $0 \leqslant \alpha < \frac{1}{2}$, and any constant h (where m, α, and h do not depend on n), there exists a constant $t_{m,\alpha,h} > 0$ (that does not depend on n) such that if $n' \leqslant hn^{\alpha}$, then*

$$T(r, m, n, \pi, n') \leqslant t_{m,\alpha,h} n^{\alpha - \frac{1}{2}}$$

A profile is said to be *tied* if a single additional voter can change the outcome. By letting $\alpha = 0$ and $h = 1$ in Theorem 5.2.3, we have that for any generalized scoring rule and any fixed m, the number of tied profiles is $O(\frac{1}{\sqrt{n}})$.

5.2.2 Conditions under which Coalitions of Manipulators are All-Powerful

Let us consider a positional scoring rule and a distribution over nonmanipulator votes. Furthermore, let us consider each alternative's expected score; let $\mathcal{C}_{max}$ be the set of alternatives with the highest expected score. Procaccia and Rosenschein (2007a) have shown that, suppose the following conditions hold.

1. The number of manipulators is in both $\omega(\sqrt{n})$ and $o(n)$, and
2. votes are drawn i.i.d.

Then, the probability that the manipulators can make any alternative in $\mathcal{C}_{max}$ win converges to 1 as $n \to \infty$. Hence, assuming $|\mathcal{C}_{max}| > 1$, the probability that the instance is open converges to 1 (however, if $|\mathcal{C}_{max}| = 1$, it converges to 0).

In this section, we prove a similar result for generalized scoring rules; in addition, we characterize the rate of convergence to 0. (In fact, in this case, Procaccia and Rosenschein also characterize this rate—for positional scoring rules.)

Specifically, in this section, we study the case where the number of manipulators is $\Omega(n^{\alpha})$ ($\frac{1}{2} < \alpha < 1$) and $o(n)$, the votes are drawn i.i.d. according to π, and a generalized scoring rule is used. We provide a sufficient condition under which the manipulators can make any alternative in a particular set of alternatives win with probability $1 - O(e^{-\Omega(n^{2\alpha-1})})$. (We need the $o(n)$ assumption for a technical reason. If $n' = \Theta(n)$, then obviously the probability that the manipulators are all-powerful is higher.)

Definition 5.2.5. *π is* compatible with *$\mathscr{K}$ w.r.t. f, if, for $V \sim \pi$, for any $l \leqslant q$, any $i, j \in K_l$ ($i \neq j$), $E((f(V))_i) = E((f(V))_j)$.*

That is, π is compatible with $\mathscr{K}$ w.r.t. f if within each element of the partition $\mathscr{K}$, the expectation of the components of $f(V)$ are the same (where V is drawn according to π).

Given $GS(f, g)$, it will be useful to have a profile P such that for some partition $\mathscr{K}$ that $GS(f, g)$ is compatible with, the components of $f(P)$ within each K_l ($l \leqslant q$) are all different. The next definition makes this precise.

Definition 5.2.6. *For any $GS(f, g)$ compatible with $\mathscr{K}$, a profile P is said to be* distinctive *w.r.t. $GS(f, g)$ and $\mathscr{K}$ if for each $l \leqslant q$ and each pair $i, j \in K_l$, $i \neq j$, $(f(P))_i \neq (f(P))_j$.*

The next definition concerns the set of alternatives that can be made to win using a distinctive profile.

Definition 5.2.7. *For any $GS(f, g)$ compatible with $\mathscr{K}$, let $W_{\mathscr{K}}(f, g)$ be a subset of the alternatives defined as follows.*

$$W_{\mathscr{K}}(f, g) = \{GS(f, g)(P) : P \text{ is distinctive w.r.t. } GS(f, g) \text{ and } \mathscr{K}\}$$

For any profile P^M of manipulators and any alternative c, we define $T(m, n, \pi, c, P^M) = Pr(GS(f, g)(P^M \cup P^{NM}) = c)$. That is, given a profile of votes P^M of the manipulators, $T(m, n, \pi, c, P^M)$ is the probability that the winner of the profile $P^M \cup P^{NM}$ is

c, when the number of alternatives is m, the number of voters is n, and the nonmanipulators' votes P^{NM} are drawn i.i.d. according to π. Now we are ready to present the theorem.

Theorem 5.2.8. *Let $GS(f,g)$ be a generalized scoring rule that is compatible with $\mathscr{K}$. Let $\pi_{\mathscr{K}}$ be a distribution over $L(\mathcal{C})$ such that $\pi_{\mathscr{K}}$ is compatible with $\mathscr{K}$ w.r.t. f. For any $m > 0$, there exist constants $t_m > 0$ and $u_m > 0$ (neither of which depend on n) such that for any constant $h > 0$ (that does not depend on n) and any alternative $c \in W_{\mathscr{K}}(f,g)$, if the number of manipulators is at least hn^α ($\dfrac{1}{2} < \alpha < 1$) (as well as $o(n)$), then there exists a coalitional manipulation P^M such that*

$$T(m,n,\pi_{\mathscr{K}},c,P^M) > 1 - t_m e^{-u_m n^{2\alpha-1}}$$

Theorem 5.2.8 states that when the number of alternatives is held fixed, if the number of manipulators is large ($\Omega(n^\alpha)$ for $\alpha > \frac{1}{2}$, as well as $o(n)$) then for any alternative $c \in W_{\mathscr{K}}(f,g)$, there exists a manipulation P^M such that when the nonmanipulators' votes are drawn i.i.d. according to $\pi_{\mathscr{K}}$, then c is the winner with a probability of $1 - O(e^{-\Omega(n^{2\alpha-1})})$.

Proof of Theorem 5.2.8: Let $n' \geqslant hn^\alpha$. If $W_{\mathscr{K}}(f,g) = \varnothing$, then Theorem 5.2.8 vacuously holds. So we assume that $W_{\mathscr{K}}(f,g) \neq \varnothing$. For each $c \in W_{\mathscr{K}}(f,g)$, we associate c with a distinctive profile (w.r.t. f and $\mathscr{K}$), denoted by P_c^*, such that $c = GS(f,g)(P_c^*)$. We recall that P_c^* is distinctive if and only if for each $l \leqslant q$ and each pair $i,j \in K_l$, $i \neq j$, $(f(P_c^*))_i \neq (f(P_c^*))_j$. Let

$$d_{min} = \min_{l \leqslant q, i,j \in K_l, i \neq j, c \in W_{\mathscr{K}}(f,g)} (|(f(P_c^*))_i - (f(P_c^*))_j|)$$

That is, d_{min} is the minimal difference between any two components within the same element of $\mathscr{K}$ of $f(P_c^*)$, taken over all $c \in W_{\mathscr{K}}(f,g)$. Since $|W_{\mathscr{K}}(f,g)| < m$ (which does not depend on n), and P_c^* is distinctive, we know that $d_{min} > 0$ and does not depend on n. Let $p_{max} = \max_{c \in C} |P_c^*|$. That is, for all $c \in W_{\mathscr{K}}(f, g)$, the number of votes in P_c^* is no more than p_{max}. We note that p_{max} does not depend on n.

For any $c \in C$, define a profile of the manipulator votes P_c^M as follows. P_c^M consists of two parts:

1. $\lfloor \frac{n'}{|P_c^*|} \rfloor P_c^*$, and

2. an arbitrary profile for the remaining $n' - \lfloor \frac{n'}{|P_c^*|} \rfloor |P_c^*|$ votes.

That is, P_c^M consists mostly of $\lfloor \frac{n'}{|P_c^*|} \rfloor$ copies of P_c^*; the remaining votes (at most $|P_c^*|$) are chosen arbitrarily. We note that $|P_c^*|$ is a constant that does not depend on n, so that the second part becomes negligible when $n \to \infty$.

The next claim provides a lower bound on the difference between any two components of $f(P_c^M)$.

Claim 5.2.1. *There exists a constant d_c that does not depend on n such that the minimum difference between components of $f(P_c^M)$ is at least $d_c n^\alpha$.*

Proof of Claim 5.2.1: Since the minimal difference between any two components of P_c^* is at least d_{min}, the minimal difference between any two components of $f(P_c^M)$ is at least $\lfloor \frac{n'}{|P_c^*|} \rfloor d_{min}$. We note that the number of arbitrarily assigned votes in P_c^M is no more than $|P_c^*|$, and the difference between any two components in a vote is no more than v_{max}. Therefore the minimal difference between any two components of $f(P^M)$ is at least

$$\lfloor \frac{n'}{|P_c^*|} \rfloor d_{min} - v_{max}|P_c^*| \geqslant (\frac{n'}{p_{max}} - 1) d_{min} - v_{max} p_{max}$$

Note that this number is $\Omega(n^\alpha)$ because p_{max}, d_{min}, and v_{max} are constants that do not depend on n, and n' is $\Omega(n^\alpha)$. Therefore, there exists a d_c that does not depend on n such that the minimal difference between any two components of $f(P_c^M)$ is at least $d_c n^\alpha$.

(End of the proof for Claim 5.2.1.) □

The next lemma is known as *Chernoff's inequality* (Chernoff, 1952).

Lemma 5.2.9 (Chernoff's inequality)**.** *Let $N \in \mathbb{N}$. Let $Y_1, \ldots, Y_N$ be N i.i.d. random variables with variance σ^2. Let $Y = \sum_{\zeta=1}^{N} Y_\zeta$. For any $0 \leqslant l \leqslant 2\sqrt{N}\sigma$, $Pr(|Y - E(Y)| \geqslant l\sqrt{N}\sigma) \leqslant 2e^{-l^2/4}$.*

For any profile P^{NM} for the nonmanipulators, any $i_1, i_2 \leqslant k$, $i_1 \neq i_2$, let $D(P^{NM}, i_1, i_2) = |(f(P^{NM}))_{i_1} - (f(P^{NM}))_{i_2}|$. The next claim states that if each vote of P^{NM} is drawn i.i.d. according to $\pi_{\mathscr{K}}$, then for any different i_1, i_2 within the same element K_l of the partition $\mathscr{K}$, the probability that the difference between the i_1th and the i_2th component of $f(P^{NM})$ is larger than $d_c n^\alpha$ is $O(e^{-\Omega(n^{2\alpha-1})})$.

Claim 5.2.2. *For any $l \leqslant q$ and any $i_1, i_2 \in K_l$ ($i_1 \neq i_2$), there exists a constant $d_{c,i_1,i_2} > 0$ that does not depend on n such that*

$$Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) \leqslant 2e^{-d_{c,i_1,i_2} n^{2\alpha-1}}$$

Proof of Claim 5.2.2: Let $Y_1^{i_1,i_2}, \ldots, Y_{n^*}^{i_1,i_2}$ be n^* i.i.d. random variables such that the distribution for each $Y_\zeta^{i_1,i_2}$ is the same as the distribution for $(f(V))_{i_1} - (f(V))_{i_2}$, where V is drawn according to π. That is, for any $V \in L(\mathcal{C})$, with probability $\pi(V)$, $Y_1^{i_1,i_2}$ takes value $(f(V))_{i_1} - (f(V))_{i_2}$. Let $Y^{i_1,i_2} = \sum_{\zeta=1}^{|NM|} Y_\zeta^{i_1,i_2}$. Then, $Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) = Pr(Y^{i_1,i_2} > d_c n^\alpha)$.

Since $\pi_{\mathscr{K}}$ is compatible with $\mathscr{K}$, for any $l \leqslant q$, $i_1, i_2 \in K_l$, we know that $E((f(V))_{i_1}) = E((f(V))_{i_2})$, where V is drawn according to π. Therefore, $E(Y_1^{i_1,i_2}) =$

0. Let $\sigma^2_{i_1,i_2}$ be the variance of $Y_1^{i_1,i_2}$. We note that σ_{i_1,i_2} does not depend on n. If $\sigma^2_{i_1,i_2} = 0$, then for any $V \in L(\mathcal{C})$, $(f(V))_{i_1} = (f(V))_{i_2}$ (because for any $V \in L(\mathcal{C})$, $\pi_{\mathscr{K}}(V) > 0$), which means that $W_{\mathscr{K}}(f,g) = \varnothing$. This contradicts the assumption that $W_{\mathscr{K}}(f,g) \neq \varnothing$. Hence $\sigma^2_{i_1,i_2} > 0$. Since $n' = o(n)$, $n^* = \Omega(n)$, and for sufficiently large n we have $\frac{d_c n^\alpha}{\sigma_{i_1,i_2}\sqrt{n^*}} \leqslant 2\sigma_{i_1,i_2}\sqrt{n^*}$. Therefore, we can use Lemma 5.2.9 (in which we let $N = n^*$) to bound $Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha)$ above as follows.

$$\begin{aligned}
& Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) \\
=& Pr(|Y^{i_1,i_2}| > d_c n^\alpha) \\
=& Pr(|Y^{i_1,i_2}| > \frac{d_c n^{\alpha-\frac{1}{2}}}{\sigma_{i_1,i_2}\sqrt{n^*}} \times \sigma_{i_1,i_2}\sqrt{n^*}) \\
\leqslant& 2e^{-(\frac{d_c n^{\alpha-\frac{1}{2}}}{\sigma_{i_1,i_2}\sqrt{n^*}})^2/4} && \text{(Lemma 5.2.9)} \\
\leqslant& 2e^{-\frac{d_c^2}{4\sigma^2_{i_1,i_2}} n^{2\alpha-1}} && (n^* \leqslant n)
\end{aligned}$$

We note that $\frac{d_c^2}{4\sigma^2_{i_1,i_2}}$ is a constant that does not depend on n. Therefore, there exists $u_{c,i_1,i_2} > 0$ such that $Pr(D(P^{NM}, i_1, i_2) > d_c n^\alpha) \leqslant 2e^{-u_{c,i_1,i_2} n^{2\alpha-1}}$.

(End of the proof for Claim 5.2.2.) □

Let $u_c = \min_{l \leqslant q, i,j \in K_l, i \neq j} u_{c,i,j}$. Then $u_c > 0$ and is a constant (that does not depend on n). We note that for any P^{NM}, if $(P^{NM} \cup P_c^M) \not\sim_{\mathscr{K}} P_c^M$, then there exists $l \leqslant q$, $i,j \in K_l$, $i \neq j$, such that $|(f(P^{NM}))_i - (f(P^{NM}))_j| > |(f(P_c^M))_i - (f(P_c^M))_j| > d_c n^\alpha$.

Therefore, we can bound the probability of $(P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^M$ below as follows.

$$\begin{aligned}
&Pr((P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^M) \\
=&1 - Pr((P^{NM} \cup P_c^M) \nsim_{\mathscr{K}} P_c^M) \\
\geqslant&1 - Pr((\exists l \leqslant q)(\exists i, j \in K_l) D(P^{NM}, i, j) > d_c n^\alpha) \\
\geqslant&1 - \sum_{l \leqslant q} \sum_{i,j \in K_l, i \neq j} Pr(D(P^{NM}, i, j) > d_c n^\alpha) \\
\geqslant&1 - \sum_{l \leqslant q} \sum_{i,j \in K_l, i \neq j} 2e^{-u_{c,i,j} n^{2\alpha-1}} \\
\geqslant&1 - \sum_{l \leqslant q} \sum_{i,j \in K_l, i \neq j} 2e^{-u_c n^{2\alpha-1}} \geqslant 1 - \frac{m(m-1)}{2} \times 2e^{-u_c n^{2\alpha-1}}
\end{aligned}$$

When n is sufficiently large, $P_c^M \sim_{\mathscr{K}} P_c^*$. Therefore, we know that there exists a constant $t_c > 0$ (that does not depend on n) such that $Pr((P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^*) \geqslant 1 - t_c e^{-u_c n^{2\alpha-1}}$. Hence

$$\begin{aligned}
&T(m, n, \pi_{\mathscr{K}}, c, P^M) \\
\geqslant&Pr((P^{NM} \cup P_c^M) \sim_{\mathscr{K}} P_c^*) \\
\geqslant&1 - t_c e^{-u_c n^{2\alpha-1}}
\end{aligned}$$

(End of the proof for Theorem 5.2.8.) □

5.2.3 All-Powerful Manipulators in Common Rules

We already showed how Theorem 5.2.3, which states a condition under which manipulability is rare, can be applied to common voting rules in Corollary 5.2.4. We have not yet done so for Theorem 5.2.8, and we will do so in this section.[1] Specifically, we prove that if the number of alternatives is fixed, then for any positional scoring rule, Copeland, STV, ranked pairs, and maximin, if the number of manipulators is

[1] Except for Bucklin.

$\Omega(n^{\alpha})$ $(\alpha > \frac{1}{2})$ and $o(n)$, and the nonmanipulators' votes are drawn i.i.d. according to the uniform distribution, then for any alternative c, there exists a coalitional manipulation that will make c win with a probability of $1 - O(e^{-\Omega(n^{2\alpha-1})})$.

The next theorem provides a necessary and sufficient condition for $W_{\mathscr{K}}(f,g)$ to be nonempty.

Theorem 5.2.10. *Let $G(f,g)$ be compatible with $\mathscr{K}$. $W_{\mathscr{K}}(f,g) \neq \emptyset$ if and only if for any $l \leqslant q$, any $i,j \in K_l$, $i \neq j$, there exists a vote $V \in L(\mathcal{C})$ such that $(f(V))_i \neq (f(V))_j$*

Proof of Theorem 5.2.10: First we prove the "if" part. Suppose that for any $l \leqslant q$, any $i,j \in K_l$, $i \neq j$, there exists a vote $V \in L(\mathcal{C})$ such that $(f(V))_i \neq (f(V))_j$. For any $l \leqslant q$, let

$$h_{l,max} = \max_{i,j \in K_l, V \in L(\mathcal{C})} \{|(f(V))_i - (f(V))_j|\},$$

$$h_{l,min} = \min_{i,j \in K_l, V \in L(\mathcal{C})} \{|(f(V))_i - (f(V))_j| : |(f(V))_i - (f(V))_j| > 0\}$$

That is, $h_{l,max}$ is the maximum difference between any two components within K_l, for any $f(V)$; $h_{l,min}$ is the minimum *positive* difference between any two components within K_l, for any $f(V)$. Then, for any $l \leqslant q$, $h_{l,max} \geqslant h_{l,min} > 0$. Let h be a natural number such that for any $l \leqslant q$, $h > \frac{h_{l,max}}{h_{l,min}} + 1$. Suppose $L(\mathcal{X}) = \{L_1, \ldots, L_{m!}\}$. Then, let $P = \sum_{s=1}^{m!} h^{m!-s} L_s$. We now show that P is distinctive w.r.t. $GS(f,g)$ and $\mathscr{K}$.

For any $l \leqslant q$, any $i,j \in K_l$, let t be the minimum natural number such that

$(f(L_t))_i \neq (f(L_t))_j$. W.l.o.g. let $(f(L_t))_i > (f(L_t))_j$. Then

$$\begin{aligned}
&(f(P))_i - (f(P))_j \\
=&\sum_{s=1}^{m!} h^{m!-s}((f(L_s))_i - (f(L_s))_j) \\
=&h^{m!-t}((f(L_t))_i - (f(L_t))_j) + \sum_{s=t+1}^{m!} h^{-s}((f(L_s))_i - (f(L_s))_j) \\
\geqslant&h^{m!-t}h_{l,min} - \sum_{s=t+1}^{m!} h^{m!-s}h_{l,max} \\
=&h^{m!-t}(h_{l,min} - \frac{1}{h}\frac{1-\frac{1}{h^{m!-t}}}{1-\frac{1}{h}}h_{l,max}) \\
>&h^{m!-t}(h_{l,min} - \frac{1}{h-1}h_{l,max}) \\
>&0
\end{aligned}$$

The last inequality holds because $h > \frac{h_{l,max}}{h_{l,min}} + 1$. Therefore, we know that for any $l \leqslant q$, any $i, j \in K_l$, $i \neq j$, $(f(P))_i \neq (f(P))_j$. Hence, P is distinctive w.r.t. $GS(f, g)$ and $\mathscr{K}$, completing the proof of the "if" part.

Now we prove the "only if" part. Suppose there exist $l \leqslant q$, $i, j \in K_l$ such that for any $V \in L(\mathcal{C})$, $(f(V))_i = (f(V))_j$. Then, for any profile P, $(f(P))_i = (f(P))_j$, which means that P is not distinctive w.r.t. $GS(f, g)$ and $\mathscr{K}$. Therefore $W_{\mathscr{K}}(f, g) = \varnothing$, completing the proof of the "only if" part.

(End of the proof for Theorem 5.2.10.) □

Now we show how the conditions in Theorem 5.2.8 are satisfied for any positional scoring rule, STV, Copeland, maximin, and ranked pairs, when the nonmanipulator votes are drawn from the uniform distribution.

Proposition 5.2.11. *Let π_u be the uniform distribution. For any rule r that is a positional scoring rule, Copeland, STV, maximin, or ranked pairs, let k_r, $GS(f_r, g_r)$*

and $\mathscr{K}_r$ be defined as in Proposition 5.1.5. Then, π_u is compatible with $\mathscr{K}_r$, and for any $l \leqslant q_r$ and any $i, j \leqslant K_l$ ($i \neq j$), there exists a vote $V \in L(\mathcal{C})$ such that $(f_r(V))_i \neq (f_r(V))_j$.

Proof of Proposition 5.2.11: We verify the condition in Theorem 5.2.10 for the common voting rules mentioned in the proposition by simple calculation, w.r.t. the GSR-formulation mentioned in the proof of Proposition 5.1.5.

positional scoring rule with scoring vector $\vec{v}$: for any $i \leqslant m$,

$$E_{V \sim \pi_u}((f_{\vec{v}}(V))_i) = \frac{\sum_{j=1}^{m} v(j)}{m}$$

Copeland, maximin, or ranked pairs: for any $i \leqslant m$, $j \leqslant m$, $i \neq j$,

$$E_{V \sim \pi_u}((f_r(V))_{(i,j)}) = \frac{1}{2}$$

STV: for any (S, j) such that $S \subset \mathcal{C}$, $|S| = i$, $c_j \notin S$,

$$E_{V \sim \pi_u}((f_{STV}(V))_{(S,j)}) = \frac{1}{m - i}$$

It left us to show, for each of these voting rules, and for any two given components (that lie within the same element of the partition), the vote that makes these two components different.

positional scoring rule with scoring vector $\vec{v}$: for any $i, j \leqslant m$, $i \neq j$, let V be the vote that ranks c_i at the top and c_j at the bottom; then, $(f_{\vec{v}}(V))_i = v(1) \neq v(m) = (f_{\vec{v}}(V))_j$.

Copeland, maximin, or ranked pairs: for any $i_1, i_2 \leqslant m$, $j_1, j_2 \leqslant m$, $i_1 \neq j_1$, $i_2 \neq j_2$, and $(i_1, j_1) \neq (i_2, j_2)$, let V be any vote in which $c_{i_1} \succ_V c_{j_1}$ and $c_{j_2} \succ_V c_{i_2}$. Because $(i_1, j_1) \neq (i_2, j_2)$, such a V exists. Then,

$$(f_r(V))_{(i_1,j_1)} = 1 \neq 0 = (f_r(V))_{(i_2,j_2)}$$

STV: for any $S \subset \mathcal{C}$, $j_1 \neq j_2$ such that $c_{j_1} \notin S$, $c_{j_2} \notin S$, let V be the vote in which c_{j_1} is at the top. Then $(f_{STV}(V))_{(S,j_1)} = 1 \neq 0 = (f_{STV}(V))_{(S,j_2)}$.

(End of the proof for Proposition 5.2.11.) □

By combining Proposition 5.2.11 and Theorem 5.2.10, we know that for any of the rules in Proposition 5.2.11, there exists a distinctive profile; hence, $W_{\mathscr{K}_r}(f, g)$ is nonempty (some alternative will win under the distinctive profile, without any tie). Also, all of these rules are neutral (they treat every alternative in the same way) when restricted to profiles that do not cause a tie, so if $W_{\mathscr{K}_r}(f, g)$ is nonempty, it must be that $W_{\mathscr{K}_r}(f, g) = \mathcal{C}$.

Corollary 5.2.12. *Let π_u be the uniform distribution over $L(\mathcal{C})$. For any rule r that is a positional scoring rule, Copeland, STV, maximin, or ranked pairs, if the number of manipulators is $\Omega(n^\alpha)$ ($\frac{1}{2} < \alpha \leqslant 1$) as well as $o(n)$, then for any $c \in \mathcal{C}$, there exists a coalitional manipulation P^M such that the probability that $r(P^M \cup P^{NM}) = c$ is $1 - O(e^{-\Omega(n^{2\alpha-1})})$.*

5.3 An Axiomatic Characterization for Generalized Scoring Rules

We have explicitly shown in the proof of Proposition 5.1.5 that a variety of common rules fall into the category of GSRs. However, we did not give any formal result about the generality of this class of rules. The apparent wide applicability of GSRs makes this class potentially interesting from the perspective of other problems in computational social choice. Indeed, some such uses are quite obvious. GSRs map every vote to a vector of scores (which are not necessarily associated with alternatives), and the outcome of the rule is based strictly on the sum of these vectors. As a result, the votes of a subset of the electorate can be summarized completely by the sum of their score vectors.[2] In fact, the definition of GSRs is even more restrictive: the final

[2] The problem of summarizing the votes of a subelectorate was introduced and studied (Chevaleyre et al., 2009; Xia and Conitzer, 2010a).

outcome only depends on direct comparisons among the components of the summed score vector. For example, the outcome may depend on a comparison between the first component and the third component of the summed vector; then, it does not matter (for this comparison) whether these components are 42 and 50, respectively, or 101 and 967, because in both cases component 1 is smaller. Because of this, the GSR framework is also useful for preference elicitation, specifically, for determining whether enough information has been elicited from the voters to declare the winner. In particular, if it becomes clear that the remaining (not yet elicited) information about the voters' preferences can no longer change any of the comparisons in scores, then we can terminate elicitation.

In Social Choice, *axiomatic* characterizations of voting rules are important because they give us deeper insight into rules, and can often be used to prove important results about rules. For GSRs, having an axiomatic characterization is especially important in order to know how the frequency-of-manipulability result for large number of manipulators (Theorem 5.2.8), which are negative results for the agenda of making manipulation computationally hard, might be circumvented. Axiomatic characterization of voting rules is a common topic in the social choice literature. For two alternatives, the majority rule has been characterized in May (1952). Young (1975) characterized positional scoring correspondences (that is, the voting correspondences that select all alternatives that have the highest total scores) by consistency, neutrality, and anonymity. Here we say that a voting correspondence r^c satisfies consistency, if for any pair of profiles P_1, P_2, if $r^c(P_1) \cap r^c(P_2) \neq \varnothing$, then $r^c(P_1 \cup P_2) = r^c(P_1) \cap r^c(P_2)$. When r' is a voting rule, that is, it always select a unique winner, this consistency coincides with the consistency defined in Section 2.2. In this section we will only consider voting rules.

In this section, we introduce a new axiomatic property for voting rules, which we call *finite local consistency.* A voting rule satisfies finite local consistency if the set of all profiles can be partitioned into finitely many parts, such that the voting rule is consistent within each part. The minimum number of parts for a rule is the *degree of consistency* for the rule. For example, a consistent rule has degree of consistency 1. We then characterize generalized scoring rules by anonymity, homogeneity, and finite local consistency, and show that the order of a GSR (that is, the dimension of the score vector) is related to the degree of consistency of the rule. It follows that *Dodgson's rule* is not a GSR, because it does not satisfy homogeneity (Brandt, 2009).

5.3.1 Finite Local Consistency

In this subsection, we formally define *finite local consistency.*

Definition 5.3.1. *Let S be a set of profiles. r is* locally consistent on S *if for any $P_1, P_2 \in S$ with $r(P_1) = r(P_2)$, we have $P_1 \cup P_2 \in S$ and $r(P_1 \cup P_2) = r(P_1) = r(P_2)$.*

Definition 5.3.2. *For any natural number t, a voting rule r is* t-consistent *if there exists a partition $\{S_1, \ldots, S_t\}$ of all profiles such that for all $i \leqslant t$, r is locally consistent within S_i. A voting rule r is* finitely locally consistent *if it is t-consistent for some natural number t.*

We emphasize that in this definition, a rule is defined for a fixed number m of alternatives, but for profiles of arbitrarily many voters. Later, we will show that some common rules are finitely locally consistent for *every* $m \in \mathbb{N}$; however, in those cases, t depends on m, which is allowed, as long as t is finite. We note that this finiteness condition is important: for *any* voting rule, there exists a partition that has infinitely many elements such that the voting rule is locally consistent, simply by letting each profile be an element by itself.

The *degree of consistency* of a voting rule r (for a particular m) is the smallest number of elements in a locally consistent partition of profiles. That is, the degree of consistency of r is t if r is t-consistent, and for any $t' < t$, r is not t'-consistent. (We note that the partition corresponding to this lowest t is not necessarily unique.) The degree of consistency can be seen as an approximation to consistency: the lower the degree of consistency of a voting rule, the more "consistent" it is, and 1-consistency is equivalent to the standard definition of consistency. We will be interested in the exact degree of consistency (rather than just whether it is finite or not), because, as we will show, this degree is related to the order of a GSR equivalent to the rule, which in turn is important for the summarization and elicitation problems that we mentioned in the introduction.

5.3.2 Finite local consistency characterizes generalized scoring rules

We now present our main result of this section. Let $\mathcal{P}(k)$ be the number of *total preorders* over k elements, that is, the total number of ways to rank k elements, allowing for ties.

Theorem 5.3.3. *r is a generalized scoring rule if and only if r is anonymous, homogenous, and finitely locally consistent. Moreover, for any t-consistent voting rule r, there exists a GSR of order $(\frac{t(t-1)m(m-1)}{4})m!+1$ that is equivalent to r; conversely, for any GSR $GS(f,g)$ of order k, there exists a $\mathcal{P}(k)$-consistent voting rule r that is equivalent to $GS(f,g)$.*[3]

Proof of Theorem 5.3.3: We prove the "if" part by a geometrical representation of a voting rule that is anonymous and homogenous, similarly to Young (1975). Let $L(\mathcal{C}) = \{l_1, \ldots, l_{m!}\}$ be the set of all linear orders over $\mathcal{C}$. Let r be an anonymous and homogenous voting rule, so that profiles can be represented as multisets of votes. Hence, there is a one-to-one correspondence between the set of all profiles and the set of all points in $\mathbb{N}^{m!}$: any profile $P = \sum_{x=1}^{m!} w_x l_x$, $w_x \in \mathbb{N}$ is associated with the point $\vec{p} = (w_1, \ldots, w_{m!})$, that is, $\vec{p} \in \mathbb{N}^{m!}$, and for any $j \leqslant m!$, the jth component

[3] The $\mathcal{P}(k)$ bound can be improved if more information about the structure of the GSR is taken into account. For the sake of simplicity, we omit further discussions of it.

of $\vec{p}$ is exactly the number of voters whose preferences are l_j in P. Therefore, r can also be seen as a mapping from $\mathbb{N}^{m!}$ to $\mathcal{C}$, defined as follows: for any $\vec{p} \in \mathbb{N}^{m!}$, $r(\vec{p}) = r(P)$, where P is the profile that $\vec{p}$ corresponds to. In the remainder of the proof, we will not distinguish between the point $\vec{p}$ and the profile P. Also, because r is homogenous, the domain of r can be extended to ${\mathbb{Q}_{\geqslant 0}}^{m!}$ (vectors of nonnegative rationales) in the following way. For any $\vec{p} \in {\mathbb{Q}_{\geqslant 0}}^{m!}$, let $h \in \mathbb{N}$ be such that $h\vec{p} \in \mathbb{N}^{m!}$; then, let $r(\vec{p}) = r(h\vec{p})$. (This is well defined because by homogeneity, the choice of h does not matter.)

Because r is t-consistent, there exists a partition $(S_1, \ldots, S_t)$ of $\mathbb{N}^{m!}$ such that r is locally consistent within each S_i. We note that $\vec{p} \in S_i$ implies $h\vec{p} \in S_i$ for each $h \in \mathbb{N}$, because each S_i must be closed under the union of vectors that produce the same result, and we can take the union of h vectors $\vec{p}$. Now, for any $i \leqslant t$, we define $S_i^{\mathbb{Q}} = \{q\vec{p} : q \in \mathbb{Q}_{\geqslant 0}, \vec{p} \in S_i\}$. It follows that ${\mathbb{Q}_{\geqslant 0}}^{m!} = \bigcup_{i=1}^{t} S_i^{\mathbb{Q}}$, and for any $i_1 \neq i_2$, $S_{i_1}^{\mathbb{Q}} \cap S_{i_2}^{\mathbb{Q}} = \{0\}$. For any $i \leqslant t$, any $j \leqslant m$, we define $S_i^j = S_i^{\mathbb{Q}} \cap r^{-1}(c_j)$. That is, S_i^j is the set of points (equivalently, profiles) in $S_i^{\mathbb{Q}}$ whose winner is c_j. It follows that for any $\vec{p}_1, \vec{p}_2 \in S_i^j \cap \mathbb{N}^{m!}$, we have $\vec{p}_1 + \vec{p}_2 \in S_i^j$; for any $\vec{p} \in S_i^j$, any $q \in \mathbb{Q}_{\geqslant 0}$, we must have $q\vec{p} \in S_i^j$. For any $S \subseteq {\mathbb{R}_{\geqslant 0}}^{m!}$, we say that S is $\mathbb{Q}$-*convex* if for any $\lambda \in \mathbb{Q} \cap [0,1]$, any $\vec{p}_1, \vec{p}_2 \in S$, we have $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S$. We say a $\mathbb{Q}$-convex set S is a $\mathbb{Q}$-*convex cone*, if for any $q \in \mathbb{Q}_{\geqslant 0}$, any $\vec{p} \in S$, we have $q\vec{p} \in S$.

Claim 5.3.1. *For any $i \leqslant t$, any $j \leqslant m$, S_i^j is a $\mathbb{Q}$-convex cone.*

Proof. For any $q_1, q_2 \in \mathbb{Q}_{\geqslant 0}$, any $\vec{p}_1, \vec{p}_2 \in S_i^j$, there exists $T \in \mathbb{N}$ such that $Tq_1\vec{p}_1, Tq_2\vec{p}_2 \in \mathbb{N}^{m!}$. We note that $Tq_1\vec{p}_1, Tq_2\vec{p}_2 \in S_i^j$, which implies that $Tq_1\vec{p}_1 + Tq_2\vec{p}_2$ is also in S_i^j. It follows that $q_1\vec{p}_1 + q_2\vec{p}_2 = \frac{1}{T}(Tq_1\vec{p}_1 + Tq_2\vec{p}_2) \in S_i^j$. □

For any $S \subseteq {\mathbb{R}_{\geqslant 0}}^{m!}$, we let $\mathrm{conv}(S)$ be the *convex hull* of S in ${\mathbb{R}_{\geqslant 0}}^{m!}$. That is, $\mathrm{conv}(S) = \{\sum_{i=1}^{h} \alpha_i\vec{p}_i : h = 1, 2, \ldots, \sum_{i=1}^{h} \alpha_i = 1, (\forall i \leqslant h)\ \alpha_i > 0, \alpha_i \in \mathbb{R}, \vec{p}_i \in S\}$.

Lemma 5.3.4 (proved in Young (1975))**.** *$S \subseteq \mathbb{Q}^{m!}$ is $\mathbb{Q}$-convex if and only if $S = conv(S) \cap \mathbb{Q}^{m!}$.*

Let $d \in \mathbb{N}$, $S_1, S_2 \subseteq \mathbb{R}^d$, and for any $x \in \mathbb{R}$, let $\delta(x) = 1$ if $x > 0$, $\delta(x) = -1$ if $x < 0$, and $\delta(0) = 0$. We say that S_1 and S_2 are *separated* by a finite set of vectors $I = \{\vec{p}_1, \ldots, \vec{p}_o\}$, in which $\vec{p}_i \in \mathbb{R}^l$ for all $i \leqslant o$, if there exist two sets $O_1, O_2 \subseteq \{-1, 0, 1\}^I$ such that $O_1 \cap O_2 = \{0\}$, and for any $\vec{p} \in S_1$ $(\vec{p} \neq 0)$, we have $\delta(\vec{p}, I) = (\delta(\vec{p} \cdot \vec{p}_1), \ldots, \delta(\vec{p} \cdot \vec{p}_o)) \in O_1$; for any $\vec{p} \in S_2$ $(\vec{p} \neq 0)$, we have $(\delta(\vec{p} \cdot \vec{p}_1), \ldots, \delta(\vec{p} \cdot \vec{p}_o)) \in O_2$. In this case we also say that I separates S_1 from S_2 *via* O_1, O_2.

$S \subseteq \mathbb{R}^d$ is called an *affine space* if for any $\vec{p}_1, \vec{p}_2 \in S$, any $q_1, q_2 \in \mathbb{R}$, we have $q_1\vec{p}_1 + q_2\vec{p}_2 \in S$. For any $S' \subseteq \mathbb{R}^d$, we let aff$(S')$ denote the *affine extension* of S' as follows: aff$(S') = \{\sum_{i=1}^h \alpha_i \vec{p}_i : h = 1, 2, \ldots, (\forall i \leqslant h)\ \alpha_i \in \mathbb{R}, \vec{p}_i \in S'\}$. That is, aff$(S')$ is the smallest affine space in $\mathbb{R}^d$ that contains S'. We let *relint*(conv(S)) denote the *relative interior* of conv(S), defined as follows. *relint*(conv(S)) is the set of all vectors $\vec{p} \in \mathbb{R}^d$ such that there exists $\epsilon > 0$ such that $B(\vec{p}, \epsilon) \cap \text{aff}(S) \subseteq \text{conv}(S)$, where $B(\vec{p}, \epsilon)$ is the ball centered on $\vec{p}$ with radius ϵ.

Lemma 5.3.5. *Let $S \subseteq \mathbb{R}^{m!}$ be an affine space, and let $S_1, S_2 \subseteq S \cap {\mathbb{Q}_{\geqslant 0}}^{m!}$ be two $\mathbb{Q}$-convex cones such that $S_1 \neq S_2$, $S_1 \cap S_2 = \{0\}$. There exists a finite set of vectors $I \subseteq \mathbb{R}^{m!}$ that separates S_1 from S_2, and $|I| \leqslant dim(S)$.*

Proof. We prove the claim by induction on dim(S). When dim$(S) = 1$, it must be the case that one of S_1 and S_2 is $\{0\}$, and the other has an element $\vec{p}' \neq 0$. Without loss of generality, we let $S_1 = \{0\}$, $S_2 \neq \{0\}$. In this case, we let $I = \{\vec{p}'\}$, $O_1 = \{0\}$, and $O_2 = \{0, 1\}$.

Suppose Lemma 5.3.5 holds for dim$(S) \leqslant d$. Without loss of generality, we assume dim(aff$(S_1)) \geqslant$ dim(aff(S_2)). When dim$(S) = d + 1$, there are two cases.

Case 1: $\dim(\text{aff}(S_1)) = \dim(\text{aff}(S_2)) = d + 1$. In this case $S = \text{aff}(S_1) = \text{aff}(S_2)$. First we prove that $\text{relint}(\text{conv}(S_1)) \cap \text{relint}(\text{conv}(S_2)) = \varnothing$. If not, suppose $\vec{p} \in \text{relint}(\text{conv}(S_1)) \cap \text{relint}(\text{conv}(S_2))$. Let $\vec{p} = \sum_{j=1}^{h} \alpha_j \vec{p}_j$, where $\sum_{j=1}^{h} \alpha_j = 1$, for all $j \leqslant h$, $\vec{p}_j \in S_1$ and $\alpha_j \geqslant 0$, and $B(\vec{p}, \epsilon) \cap S \subseteq \text{conv}(S_1)$, $B(\vec{p}, \epsilon) \cap S \subseteq \text{conv}(S_2)$. There exist $\beta_j \in \mathbb{Q}_{\geqslant 0}$ $(j \leqslant h)$ such that $\vec{p}^* = \sum_{j=1}^{h} \beta_j \vec{p}_j \neq 0$, and the distance between $\vec{p}^*$ and $\vec{p}$ is less than ϵ (by setting the β_j sufficiently close to the α_j). We note that S_1 is $\mathbb{Q}$-convex, which means that $\vec{p}^* \in S_1$. It follows that $\vec{p}^* \in \text{conv}(S_2)$, because $\vec{p}^* \in B(\vec{p}, \epsilon) \cap S$. From Lemma 5.3.4 we have that $S_2 = \text{conv}(S_2) \cap {\mathbb{Q}_{\geqslant 0}}^{m!}$. Therefore, $\vec{p}^* \in \text{conv}(S_2) \cap {\mathbb{Q}_{\geqslant 0}}^{m!} = S_2$. This contradicts the assumption that $S_1 \cap S_2 = \{0\}$.

Because $\text{relint}(\text{conv}(S_1)) \cap \text{relint}(\text{conv}(S_2)) = \varnothing$, we apply the separating hyperplane theorem: there exists a hyperplane $H_{\vec{p}^*}$ characterized by $\vec{p}^* \in \mathbb{R}^{m!}$, such that for any $\vec{p}_1 \in S_1$, $\vec{p}_1 \cdot \vec{p}^* \leqslant 0$; for any $\vec{p}_2 \in S_2$, $\vec{p}_2 \cdot \vec{p}^* \geqslant 0$; and at least one of S_1 and S_2 is not contained in $H_{\vec{p}^*}$. We let $S' = S \cap H_{\vec{p}^*}$, and $S_1' = S_1 \cap S'$, $S_2' = S_2 \cap S'$. $H_{\vec{p}^*}$ does not contain S, so it follows that $\dim(S') < \dim(S) = d + 1$. Applying Lemma 5.3.5 on S', S_1', S_2' (using the induction assumption), there exists a set of vectors I' that separates S_1' from S_2' via O_1', O_2', $|I'| \leqslant d$. Let $I = \{\vec{p}^*\} \cup I'$ and $O_1 = \{\vec{a} \in \{-1, 0, 1\}^I : \vec{a}|_{\{\vec{p}^*\}} = -1 \vee (\vec{a}|_{\{\vec{p}^*\}} = 0 \wedge \vec{a}|_{I'} \in O_1')\}$ (here, for $J \subseteq I$, let $\vec{a}|_J$ be the components of $\vec{a}$ corresponding to the vectors in J). This works because for any $\vec{p} \in S_1$, either $\vec{p}$ is in the open halfspace $\{\vec{p}' : \vec{p}' \cdot \vec{p}^* < 0\}$, or $\vec{p}$ is in $S_1 \cap H_{\vec{p}^*}$. Similarly, let $O_2 = \{\vec{a} \in \{-1, 0, 1\}^I : \vec{a}|_{\{p^*\}} = 1 \vee (\vec{a}|_{\{p^*\}} = 0 \wedge \vec{a}|_{I'} \in O_2')\}$. It follows that I separates S_1 from S_2 via O_1, O_2, and $|I| = |I'| + 1 \leqslant d + 1$.

Case 2: $\dim(\text{aff}(S_2)) < d + 1$. If $\text{aff}(S_1) = \text{aff}(S_2)$, then let $S' = \text{aff}(S_1)$, $|S'| < d + 1$. Applying Lemma 5.3.5 on S', S_1, S_2 (by the induction assumption), we can conclude that there exists $I' \subseteq {\mathbb{Q}_{\geqslant 0}}^{m!}$ that separates S_1 from S_2, and $|I'| \leqslant d < d+1$. If $\text{aff}(S_1) \neq \text{aff}(S_2)$, then there exists a hyperplane $H_{\vec{p}^*}$ (orthogonal to $\vec{p}^*$) such that $0 \in H_{\vec{p}^*}$, $S_2 \subseteq H_{\vec{p}^*}$, and $S_1 \nsubseteq H_{\vec{p}^*}$ (because the intersection of all hyperplanes that contains S_2 is S_2). Let $S' = \text{aff}(S_2)$, and $S'_1 = S_1 \cap S'$. S' is an affine space whose dimension is $\dim(\text{aff}(S_2)) < d+1$. For any $\vec{p}_1, \vec{p}_2 \in S'_1$, any $\lambda \in \mathbb{Q}_{\geqslant 0}$, we have that $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S_1$ (because S_1 is $\mathbb{Q}$-convex), and $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S'$ (because S' is an affine space); hence, $\lambda\vec{p}_1 + (1-\lambda)\vec{p}_2 \in S'_1$. Therefore, S'_1 is a $\mathbb{Q}$-convex cone.

By applying Lemma 5.3.5 on S', S'_1, S_2 (using the induction assumption), there exists $I' \subset {\mathbb{Q}_{\geqslant 0}}^{m!}$ ($|I'| \leqslant d$) that separates S'_1 from S_2 via O'_1, O'_2. We let $I = I' \cup \{\vec{p}^*\}$; $O_1 = \{\vec{a} \in \{-1,0,1\}^I : \vec{a}|_{\{\vec{p}^*\}} \neq 0 \vee (\vec{a}|_{\{\vec{p}^*\}} = 0 \wedge \vec{a}|_{I'} \in O'_1)\}$. This works because for any $\vec{p} \in S_1$, either $\vec{p} \cdot \vec{p}^* \neq 0$ (meaning that $\vec{p}$ is not in S'), or $\vec{p} \cdot \vec{p}^* = 0$, and $\delta(\vec{p}, I') \in O'_1$ (meaning that $\vec{p}$ is in $S_1 \cap S'$). Similarly we define $O_2 = \{\vec{a} \in \{-1,0,1\}^I : \vec{a}|_{\{\vec{p}^*\}} = 0 \wedge a|_{I'} \in O'_2\}$. It follows that I separates S_1 from S_2, and $|I| = |I'| + 1 \leqslant d+1$. This completes the proof of Lemma 5.3.5. □

For any $i_1, i_2 \leqslant t$, $j_1, j_2 \leqslant m$, where either $i_1 \neq i_2$ or $j_1 \neq j_2$, $S_{i_1}^{j_1} \cap S_{i_2}^{j_2} = \{0\}$. (We recall that S_i^j is the set of points in $S_i^{\mathbb{Q}}$ whose winner is c_j.) From Lemma 5.3.5, there exists a finite set $I_{i_1j_1,i_2j_2}$ of vectors that separates $S_{i_1}^{j_1}$ from $S_{i_2}^{j_2}$ via $O^1_{i_1j_1,i_2j_2}, O^2_{i_1j_1,i_2j_2}$, where $|I_{i_1j_1,i_2j_2}| \leqslant m!$. Now we can define a corresponding generalized scoring rule, as follows.

- $k = |\bigcup_{(i_1,j_1)\neq(i_2,j_2)} I_{i_1j_1,i_2j_2}| + 1$, and the components are indexed by vectors in some $I_{i_1j_1,i_2j_2}$, and a 0 component (which is always 0). Because $|I_{i_1j_1,i_2j_2}| \leqslant m!$, we have $k \leqslant (\frac{t(t-1)m(m-1)}{4})m! + 1$.
- For any $(i_1,j_1) \neq (i_2,j_2)$, any $\vec{p} = (p_1, \dots, p_{m!}) \in I_{i_1j_1,i_2j_2}$, any $b \leqslant m!$, the $\vec{p}$ component of the generalized score vector given vote (ranking) l_b is $f(l_b) = p_b$. We note that for any profile $\vec{p} = (w_1, \dots, w_{m!})$, any $\vec{p}^* = (p^*_1, \dots, p^*_{m!}) \in I_{i_1j_1,i_2j_2}$, the $\vec{p}^*$ component of $f(\vec{p})$ is $\sum_{x=1}^{m!} w_x p^*_x = \vec{p} \cdot \vec{p}^*$.
- For any $\vec{a} \in {\mathbb{Q}_{\geqslant 0}}^k$ with $\vec{a} \neq 0$, $g(\vec{a}) = c_j$ if and only if there exists $i \leqslant t$ such that for any $i' \leqslant t$, $j' \leqslant m$, there exists $o \in O^1_{ij,i'j'}$ such that for any $\vec{p}^* \in I_{ij,i'j'}$, the following three conditions hold: (1) $\vec{a}|_{\vec{p}^*}$ is strictly larger than 0 (the value of the 0

component), if and only if $o|_{\vec{p}*} = 1$; (2) $\vec{a}|_{\vec{p}*}$ is equal to 0, if and only if $o|_{\vec{p}*} = 0$; and (3) $\vec{a}|_{\vec{p}*}$ is strictly smaller than 0, if and only if $o|_{\vec{p}*} = -1$. That is, $g(\vec{a}) = c_j$ if and only if there exists $i \leqslant t$ such that for any i', j', we always have $\vec{a} \notin S_{i'}^{j'}$ by using the set of separation vectors $I_{ij,i'j'}$. (That g is well defined will follow from the following argument.)

Next, we prove that $GS(f, g) = r$. For any profile $\vec{p} \in \mathbb{Q}_{\geqslant 0}{}^{m!}$, suppose $\vec{p} \in S_i^j$. For any $(i, j) \neq (i', j')$, since $\vec{p} \in S_i^j$, by using the separation vectors $I_{ij,i'j'}$ and $O^1_{ij,i'j'}, O^2_{ij,i'j'}$, $\vec{p}$ should be classified as "not in $S_{i'}^{j'}$". That is, there exists $o \in O^1_{ij,i'j'}$ such that for any $\vec{p}^* \in I_{ij,i'j'}$, $o|_{\vec{p}*} = \delta(\vec{p} \cdot \vec{p}^*)$; and for any $o' \in O^2_{ij,i'j'}$, there exists $\vec{p}^* \in I_{ij,i'j'}$ such that $o'|_{\vec{p}*} \neq \delta(\vec{p} \cdot \vec{p}^*)$. It follows that $GS(f, g)(\vec{p}) = c_j$.

The "only if" part is straightforward. For any total preorder $\mathcal{O}$ over $\{1, \ldots, k\}$, we let $S_{\mathcal{O}} = \{\vec{p} \in \mathbb{Q}_{\geqslant 0}{}^{m!} : f(\vec{p}) \sim \mathcal{O}\}$. For any $\vec{p}_1, \vec{p}_2 \in S_{\mathcal{O}}$, $f(\vec{p}_1 + \vec{p}_2) = f(\vec{p}_1) + f(\vec{p}_2) \sim \mathcal{O}$, so that $GS(f, g)(\vec{p}_1) = GS(f, g)(\vec{p}_2) = GS(f, g)(\vec{p}_1 + \vec{p}_2)$. Hence, $GS(f, g)$ is locally consistent within $S_{\mathcal{O}}$. It follows that $\{S_{\mathcal{O}}\}$ is a finitely locally consistent partition for the rule, of size $\mathcal{P}(k)$. □

We are not aware of any closed-form formula for $\mathcal{P}(k)$, though there exist recursive formulas. We now give a simple upper bound on $\mathcal{P}(k)$. Any total preorder V can be represented by a strict order $(c_{i_1} \succ c_{i_2} \succ \ldots \succ c_{i_m})$ and a string $\vec{s} = (s_1, \ldots, s_{m-1}) \in \{0, 1\}^{m-1}$, as follows: if $s_l = 0$ then $c_{i_l} \succ_V c_{i_{l+1}}$, and if $s_l = 1$ then $c_{i_l} \approx_V c_{i_{l+1}}$. This implies $\mathcal{P}(k) \leqslant k! 2^{k-1}$.

5.4 A Scheduling Approach for Positional Scoring Rules

So far in this chapter we have been focusing on characterizing the frequency of manipulability for common voting rules, in order to show that computational complexity is not a strong barrier against manipulation. In this section, we argue that computational complexity is not a strong barrier against manipulation from the viewpoint of approximation. The optimization problem we will look at in this section asks for the smallest total number (weight) of the manipulators that can make a given alternative win. This optimization problem severs as our basis for approximation, and has two dimensions: the first dimension concerns whether the votes are weighted or unweighted, and the second dimension concerns whether the manipulators' votes are divisible (that is, each manipulator can cast a convex combination of linear orders as her vote) or not. For example, when the voters are unweighted and are not allowed cast divisible votes, the problem is the UCO problem (Definition 3.1.3).

Our main contribution is the exploration of a surprising and fruitful connection between coalitional manipulation for positional scoring rules and scheduling. We demonstrate that some of work on the latter problem can be leveraged to obtain nontrivial algorithmic results for the former problem.

The intuition behind the reduction is as follows. The scheduling problem to which we reduce is that of scheduling on parallel machines where the goal is to minimize makespan. In the coalitional manipulation problem for a positional scoring rule with scoring vector $\vec{s}_m$, each manipulator j always ranks the coalition's preferred alternative c first, but must award $\vec{s}_m(i) \cdot w_j$ points to the alternative it ranks ith, where w_j is the manipulator's weight. For any $i \geqslant 2$, we define a machine for $\vec{s}_m(i)$; the larger $\vec{s}_m(i)$ is in relation to $\vec{s}_m(1)$, the slower the machine is. Furthermore, each alternative besides c is a job; the larger the gap between the score of this alternative and the score of c, the larger the job is. When a manipulator with weight w_j ranks an alternative in the ith position, it decreases the gap between c and this alternative by $(\vec{s}_m(1) - \vec{s}_m(i))w_j$ points, which, under the detailed reduction, is equivalent to processing the corresponding job on the $(i-1)$th slowest machine for w_j time units.

In addition to WCM (Definition 3.1.1), UCM (Definition 3.1.2), and UCO (Definition 3.1.3). In this section we also study the following problem for positional scoring rules.

Definition 5.4.1. *The* Coalitional Optimization for divisible votes (COd) *problem is defined as follows. An instance is a tuple* $(r, P^{NM}, \vec{w}^{NM}, c)$*, where* r *is a voting rule,* P^{NM} *is the non-manipulators' profile,* $\vec{w}^{NM}$ *represents the weights of* P^{NM}*, and* c *is the alternative preferred by the manipulators. We are asked to find the minimum* W^M *such that there exist a divisible vote* V^M *for one manipulator with weight* W^M*, such that*

$$r((P^{NM}, \{V^M\}), (\vec{w}^{NM}, W^M)) = c$$

In the remainder of this section, we assume that c is ranked in the top position in the fixed-order tie-breaking mechanism. We let WCMd, UCMd, UCOd denote the variants of WCM, UCM, UCO, respectively, in which votes are divisible.[4] We note that it is irrelevant whether the votes of the non-manipulators are divisible or not; what matters is whether the manipulators' votes are divisible.

In Section 5.4.1, we consider WCMd, which may be interesting in its own right, but mainly serves to prepare the ground for our results regarding WCM. We give a polynomial-time algorithm for WCMd under any positional scoring rule by reducing it to the well-studied scheduling problem known as $Q|pmtn|C_{max}$ (in which preemptions are allowed). This algorithm also solves COd.

In Section 5.4.2 we deal with the indivisible case (WCM), and augment the WCMd algorithm with a rounding technique. Based on existing results from the scheduling literature, we can assume that the scheduling solutions use relatively few preemptive break points. We then show that in the coalitional manipulation problem, we need at most one additional voter per preemptive break point. We obtain the following theorem, which is a somewhat weaker but far more generally applicable version of the main result of Zuckerman et al. regarding Borda (Zuckerman et al.,

[4] We do not need to define similar variant for COd, because it is not hard to see that any solution to a COd instance where the votes are divisible can be converted in polynomial time to a solution to the same instance where the votes are indivisible.

2009, Theorem 3.4).

THEOREM 5.4.10. *Algorithm 2 runs in polynomial time and*

1. *if the algorithm returns* false*, then there is no successful manipulation (even for the WCMd version of the instance);*

2. *otherwise, the algorithm returns a successful manipulation for a modified set of manipulators, consisting of the original manipulators plus at most $m-2$ additional manipulators, each with weight at most $W/2$, where W is the maximum weight of the manipulators.*

Crucially, in most settings of interest (e.g., political elections), the number of alternatives m is small compared to the number of voters, or even the number of manipulators. Moreover, WCM is NP-complete under scoring rules such as Borda and Veto, even when there are only three alternatives (Conitzer et al., 2007). Therefore, in many important scenarios, $m-2$ additional manipulators constitute a very small fraction of the total number of manipulators, that is, the algorithm gives a good "approximation" to WCM.

A direct implication of Theorem 5.4.10 is that in the unweighted case (UCM) our approximation algorithm always finds a manipulation with at most $m-2$ additional manipulators, if there exists one for the given instance. Put another way, the algorithm approximates UCO to an additive term of $m-2$.

In Section 5.5, we establish an "integrality gap," in the following sense: the optimal solution to UCO can require $m-2$ more manipulators than the optimal solution to UCOd (Theorem 5.5.3). Moreover, we show that there is a family of instances of UCO such that any algorithm that is based on rounding an optimal solution for COd requires $m-2$ more votes than the optimal UCO solution (Theorem 5.5.4). These results suggest that the analysis of the guarantees provided by our technique is tight.

5.4.1 Algorithms for WCMd and COd

In this section we present algorithms for WCMd and COd. We devise a polynomial-time algorithm that solves WCMd by reducing it to the scheduling problem known as $Q|pmtn|C_{max}$. This algorithm also solves COd exactly. In the next subsection (Section 5.4.2), we augment the algorithm for WCMd with a rounding technique, and obtain an approximation algorithm for WCM as a result. While our solution for WCMd may be interesting in its own right, its main purpose is to provide intuitions and techniques that are subsequently leveraged for approximating WCM.

We will show how to reduce WCMd/COd to the scheduling problem of *parallel uniform machines with preemption*, categorized as $Q|pmtn|C_{max}$ (see, for example, Brucker (2007) for the meaning of the notation). In an instance of $Q|pmtn|C_{max}$, we are given $\bar{n}$ jobs $\mathscr{J} = \{J_1, \dots J_{\bar{n}}\}$ and $\bar{m}$ machines $\mathcal{M} = \{M_1, \dots, M_{\bar{m}}\}$; each job J_i has a workload $p_i \in \mathbb{R}_+$, and the processing speed of machine M_i is $s^i \in \mathbb{R}_+$, that is, it will finish s^i amount of work in one unit of time. A *preemption* is an interruption of the job that is being processed on one machine (the job may be resumed later, not necessarily on the same machine). Preemptions are allowed in $Q|pmtn|C_{max}$. We are asked for the minimum makespan, i.e., the minimum time to complete all jobs, and an optimal schedule.

We first draw a natural connection between WCMd/COd under positional scoring rules and $Q|pmtn|C_{max}$. After counting the non-manipulators' votes only, each alternative will have a total non-manipulator score. For any $i \leqslant m-1$, we let p_i denote the gap between the non-manipulator score of c_i and the non-manipulator score of c (which is positive if the former is larger; the case where the gap is negative is trivial). In particular, the p_i's can be seen as the workload of $m-1$ jobs. We note that, without loss of generality, the manipulators will always rank c in the top position. Therefore, a manipulator vote (of weight 1) in which c_j is ranked in the

ith position decreases the gap between c_j and c by $\vec{s}_m(1) - \vec{s}_m(i)$ points.

We consider a set of $m-1$ machines $M_1, \ldots, M_{m-1}$ whose speeds are $\vec{s}_m(1) - \vec{s}_m(2), \ldots, \vec{s}_m(1) - \vec{s}_m(m)$, respectively. A ranking (a vote) is equivalent to an allocation of the $m-1$ jobs to machines: an alternative ranked i positions below c corresponds to a job allocated to the ith slowest machine. We can now see that the minimum makespan of the scheduling problem is the minimum total weight of the manipulators required to make c a winner, that is, the optimal solution to COd. For WCMd, the goal is to compute the votes for $\sum_{i=1}^{k} w_i$ "amount" of manipulators (since the votes are divisible, a problem instance with k manipulators with weights $\vec{w}$ is equivalent to a problem instance with a single manipulator whose weight is $\sum_{i=1}^{k} w_i$), such that the final total score of c is at least the final total score of any other alternative. This is equivalent to computing a schedule that completes all jobs within time at most $\sum_{i=1}^{k} w_i$.

Formally, for a WCMd instance $(\vec{s}_m = (\vec{s}_m(1), \ldots, \vec{s}_m(m)), P^{NM}, w^{NM},$
$c, k, (w_1, \ldots, w_k))$, we construct an instance of $Q|pmtn|C_{max}$ with $m-1$ jobs and $m-1$ machines (that is, $\bar{m} = \bar{n} = m-1$) as follows. For any $i \leqslant m-1$, we let $s^i = \vec{s}_m(1) - \vec{s}_m(i+1)$, $p_i = \max\{\vec{s}_m(P^{NM}, w^{NM}, c_i) - \vec{s}_m(P^{NM}, w^{NM}, c), 0\}$. We do not distinguish between alternative c_i and job J_i. This reduction is illustrated in the following example.

Example 5.4.2. Let $m = 4$, $\mathcal{C} = \{c, c_1, c_2, c_3\}$. The positional scoring rule is Borda (which corresponds to the scoring vector $(3, 2, 1, 0)$). The non-manipulators are unweighted (that is, their weights are 1), and their profile is
$P^{NM} = (V_1^{NM}, V_2^{NM}, V_3^{NM}, V_4^{NM})$, defined as follows.

$$V_1^{NM} = [c_1 \succ c \succ c_2 \succ c_3], \qquad V_2^{NM} = [c_2 \succ c_1 \succ c \succ c_3]$$

$$V_3^{NM} = [c_3 \succ c_2 \succ c_1 \succ c], \qquad V_4^{NM} = [c_1 \succ c_2 \succ c_3 \succ c]$$

We have that $s(P^{NM}, c) = 3$, $s(P^{NM}, c_1) = 9$, $s(P^{NM}, c_2) = 8$, $s(P^{NM}, c_3) = 4$.

Therefore, we construct a $Q|pmtn|C_{max}$ instance in which there are 3 machines M_1, M_2, M_3 whose speeds are $s^1 = 1, s^2 = 2, s^3 = 3$, corresponding to the 2nd, 3rd, and 4th position in the votes, respectively, and 3 jobs J_1, J_2, J_3, whose workloads are $p_1 = 6 = (9 - 3), p_2 = 5 = (8 - 3), p_3 = 1 = (4 - 3)$, respectively. □

Let $W_0 = 0$, $W = \max_{j \leqslant k} w_j$, and for any $1 \leqslant i \leqslant k$, $W_i = \sum_{j=1}^{i} w_j$. A schedule is usually represented by a *Gantt chart*, as illustrated in Figure 5.1. (We note that Figure 5.1 is not the solution to Example 5.4.2.)

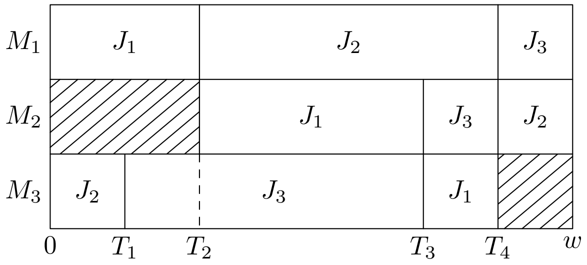


FIGURE 5.1: An example schedule. The machines are idle in shaded areas.

Let w be the minimum makespan for the $Q|pmtn|C_{max}$ instance constructed above, and let $f^* : \mathcal{M} \times [0, w] \to \mathscr{J} \cup \{I\}$ be an optimal solution to $Q|pmtn|C_{max}$, where I means that the machine is idle. If $w > W_k$, then there is no successful manipulation that makes c a winner. If $w \leqslant W_k$, we first extend the optimal solution f^* to make it fully occupy the whole time interval $[0, W_k]$; any way of allocating jobs to machines in the added time would suffice. Let f be the solution obtained in this way.

Given f, for any time $t \in [0, W_k]$, we say that t is a *preemptive break point* if there is a preemption at t—formally, there exists a machine M_i such that for some $\epsilon' > 0$, we have that for all $\epsilon \in [0, \epsilon']$, $f(M_i, t - \epsilon) \neq f(M_i, t + \epsilon)$, that is, the job being processed at time $t - \epsilon$ on M_i is different from the job being processed at time $t + \epsilon$. We let $B_f = \{T_1, \ldots, T_l\}$ denote the preemptive break points of f, where $0 < T_1 < T_2 < \ldots < T_l < W_k$. For example, the set of preemptive break points of the schedule in Figure 5.1 is $B_f = \{T_1, T_2, T_3, T_4\}$.

Example 5.4.3. The minimum makespan of the scheduling problem instance in Example 5.4.2 is $(6+5)/5 = 11/5$. An optimal schedule f is as follows.

$\mathbf{M_1}$: For any $0 \leqslant t \leqslant 11/5$, $f(M_1, t) = J_3$.

$\mathbf{M_2}$: For any $0 \leqslant t \leqslant 8/5$, $f(M_2, t) = J_2$; for any $8/5 < t \leqslant 11/5$, $f(M_2, t) = J_1$.

$\mathbf{M_3}$: For any $0 \leqslant t \leqslant 8/5$, $f(M_3, t) = J_1$; for any $8/5 < t \leqslant 11/5$, $f(M_3, t) = J_2$.

$t = 8/5$ is the only preemptive break point in this schedule. □

Any solution to the $Q|pmtn|C_{max}$ instance obtained from the reduction can be converted to a solution to WCMd in the following way. First, we assign jobs to all idle machines arbitrarily to ensure that at any time between 0 and W_k, no machines are idle and all jobs are allocated. Formally, we define $f' : \mathcal{M} \times [0, W_k] \rightarrow \mathscr{J}$ such that $\{f'(M_1, t), \ldots, f'(M_{m-1}, t)\} = \{J_1, \ldots, J_{m-1}\}$ for all t, and for any $M \in \mathcal{M}$ and $t \in [0, W_k]$, we have that if $f(M, t) \in \mathscr{J}$, then $f'(M, t) = f(M, t)$. For example, we can assign jobs to the shaded areas (which represent idle time) in the schedule in Figure 5.1 in the way illustrated in Figure 5.2.

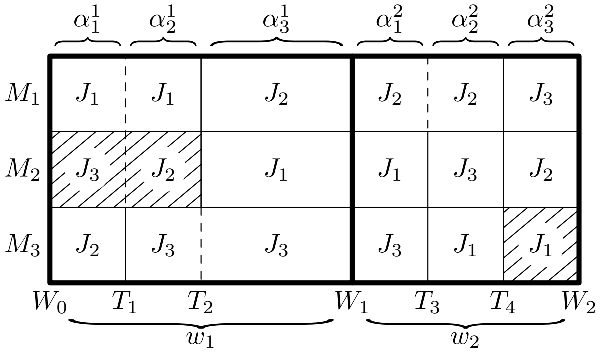


FIGURE 5.2: Conversion of an optimal schedule to a solution for WCMd.

Next, for any $1 \leqslant i \leqslant k$, we convert the schedule to the manipulators' votes in the natural way:

- If there are no preemption break points in (W_{i-1}, W_i), we let manipulator i vote for $c \succ f'(M_1, W_{i-1} + \epsilon) \succ f'(M_2, W_{i-1} + \epsilon) \succ \ldots \succ f'(M_{m-1}, W_{i-1} + \epsilon)$, where $\epsilon > 0$ is sufficiently small.

- If there are preemptive break points in (W_{i-1}, W_i), denoted by $T_a, T_{a+1}, \ldots, T_{a+b-1}$, then we let $V_1^i, \ldots, V_{b+1}^i$ denote the orders that correspond to the schedule at times $W_{i-1} + \epsilon, T_a + \epsilon, \ldots, T_{a+b-1} + \epsilon$, respectively. Let $\alpha_1^i = T_a - W_{i-1}$, $\alpha_2^i = T_{a+1} - T_a, \ldots, \alpha_{b+1}^i = W_i - T_{a+b-1}$. We let manipulator i vote for $\sum_{j=1}^{b+1} [\alpha_j^i / (W_i - W_{i-1})] \cdot V_j^i$.

Example 5.4.4. Suppose there are two manipulators whose weights w_1 and w_2 are illustrated in Figure 5.2. Manipulator 1 votes $[(1/4)(c \succ c_1 \succ c_3 \succ c_2) + (1/4)(c \succ c_1 \succ c_2 \succ c_3) + (1/2)(c \succ c_2 \succ c_1 \succ c_3)]$; manipulator 2 votes $[(1/3)(c \succ c_2 \succ c_1 \succ c_3) + (1/3)(c \succ c_2 \succ c_3 \succ c_1) + (1/3)(c \succ c_3 \succ c_2 \succ c_1)]$. □

On the basis of the exposition above we now refer the reader to Algorithm 1. The algorithm solves WCMd in three steps: 1. convert the WCMd instance to a $Q|pmtn|C_{max}$ instance; 2. apply a polynomial-time algorithm that solves $Q|pmtn|C_{max}$ (for example, the algorithm in Gonzalez and Sahni (1978)); 3. convert the solution to the scheduling instance to a solution to the WCMd instance. Algorithm 1 also solves COd, because the makespan w computed in Line 3 is the optimal solution to COd. It is easy to verify that the algorithm runs in polynomial time. To conclude, we have the following result.

Theorem 5.4.5. *Algorithm 1 solves WCMd and COd (exactly) in polynomial time.*

5.4.2 Algorithm for WCM

We now move on to the more difficult indivisible case. We first note that Algorithm 1 cannot be directly applied to WCM, because the manipulators' votes constructed in

```
Algorithm 1: compWCMd
1  $\forall i \leqslant m-1$, $s^i \leftarrow \vec{s}_m(1) - s_{i+1}$
2  $\forall i \leqslant m-1$, $p_i \leftarrow \max\{s(P^{NM}, w^{NM}, c_i) - s(P^{NM}, w^{NM}, c), 0\}$
3  Solve the $Q|pmtn|C_{max}$ instance (e.g., by the algorithm in Gonzalez and Sahni
   (1978)). Let $w$ and $f$ denote the minimum makespan and an extended
   optimal schedule; let $T_1, \ldots, T_l$ denote the preemptive break points.
4  if $w > W_k$ then
5      return false.
6  end
7  Let $f' : \mathcal{M} \times [0, W_k] \rightarrow \mathscr{J}$ be such that
   $\{f'(M_1, t), \ldots, f'(M_{m-1}, t)\} = \{J_1, \ldots, J_{m-1}\}$, and for any $M \in \mathcal{M}$, any
   $t \in [0, W_k]$, we have that if $f(M, t) \in \mathscr{J}$, then $f'(M, t) = f(M, t)$.
8  for $i = 1$ to $k$ do
9      Let $V_1^i = [c \succ f'(M_1, W_{i-1} + \epsilon) \succ \ldots \succ f'(M_{m-1}, W_{i-1} + \epsilon)]$
10     $j \leftarrow 2$
11     for each preemptive break point $T \in (W_{i-1}, W_i)$ (in order) do
12         Let $V_j^i = [c \succ f'(M_1, T + \epsilon) \succ \ldots \succ f'(M_{m-1}, T + \epsilon)]$
13         $j \leftarrow j + 1$
14     end
15     For any $j$, let $\alpha_j^i$ be the length of the $j$th interval in $[W_{i-1}, W_i]$ induced by
       the preemptive break points.
16     Let manipulator $i$ vote $\sum_j [\alpha_j^i / (W_i - W_{i-1})] \cdot V_j^i$, and add this vote to $P^M$
17 end
18 return $P^M$
```

Line 16 can be divisible. For any positional scoring rule, if there is a successful manipulation (in which all manipulators rank c in the top position), and we increase the weights of the manipulators, then c still wins the election. This property is known as *monotonicity in weights* (see Zuckerman et al. (2009) for a formal definition and the proof). Therefore, instead of having manipulator i cast the divisible vote $\sum_j [\alpha_j^i / (W_i - W_{i-1})] \cdot V_j^i$, we let her cast the indivisible vote V_{j*}^i, which is one of the V_j^i with the highest weight among all the V_j^i's constructed for manipulator i. In addition, for any $j \neq j^*$, we add one extra manipulator whose weight is α_j^i, and let the new manipulator vote V_j^i. It turns out that if we use a particular algorithm for the scheduling problem, then the solution will not require too many additional manipulators. This gives us Algorithm 2 for WCM.

Algorithm 2: compWCM

```
This algorithm is the same as Algorithm 1, except for the following two lines:
3  Use the algorithm in Gonzalez and Sahni (1978) to solve the scheduling
   problem
16 Let manipulator $i$ vote for $V^i_{j*}$, where for any $j \neq j^*$, $\alpha^i_{j*} \geqslant \alpha^i_j$; and for any
   $j \neq j^*$, we add a new manipulator whose weight is $\alpha^i_j$, and let her vote $V^i_j$
```

Example 5.4.6. Let the coalitional manipulation problem instance be the same as in Example 5.4.2. Suppose we have two manipulators whose weights are both 1; then, because the minimum makespan is $11/5 > 2$ (as observed in Example 5.4.3), there is no solution to the WCMd and WCM problem instances. The solution to the COd problem instance is 11/5.

Now suppose we have two manipulators, whose weights are $w_1 = 1$ and $w_2 = 6/5$, respectively. Let f be the optimal schedule defined in Example 5.4.3. A solution to the WCMd problem instance is obtained as follows. Manipulator 1 votes $[c \succ c_3 \succ c_2 \succ c_1]$, and manipulator 2 votes $[(1/2)(c \succ c_3 \succ c_2 \succ c_1) + (1/2)(c \succ c_3 \succ c_1 \succ c_2)]$. For WCM, the vote of manipulator 1 is the same, the vote of manipulator 2 is $[c \succ c_3 \succ c_2 \succ c_1]$, and there is one additional manipulator, whose weight is 3/5 and whose vote is $[c \succ c_3 \succ c_1 \succ c_2]$. □

Example 5.4.7. Suppose there are two manipulators whose weights are illustrated in Figure 5.2. The vote of manipulator 1 is $c \succ c_2 \succ c_1 \succ c_3$, and we introduce two new manipulators with weight $w_1/4$ whose votes are $c \succ c_1 \succ c_3 \succ c_2$ and $c \succ c_1 \succ c_2 \succ c_3$. The vote of manipulator 2 is $c \succ c_2 \succ c_1 \succ c_3$, and we introduce two new manipulators with weight $w_2/3$ whose votes are $c \succ c_2 \succ c_3 \succ c_1$ and $c \succ c_3 \succ c_2 \succ c_1$. Since $|B_f|$ (the number of preemptive break points) is 4, there are in total four additional manipulators. □

For any $j \neq j^*$, we must have $\alpha^i_j \leqslant (W_i - W_{i-1})/2 \leqslant W/2$ (recall that $W =$

$\max_{j \leqslant k} w_j$). Moreover, for any preemptive break point we introduce at most one extra manipulator. Therefore, we immediately have the following lemma that relates the number of the new manipulators to the number of preemptive break points.

Lemma 5.4.8. *If $w \geqslant W_k$, then there is no successful manipulation for WCMd (nor for WCM); otherwise, Algorithm 2 returns a manipulation with at most $|B_f|$ additional manipulators, each with weight at most $W/2$.*

Therefore, the smaller $|B_f|$ is, the fewer new manipulators are introduced by Algorithm 2. $|B_f|$ depends on which algorithm we use to solve $Q|pmtn|C_{max}$ in Line 3. In fact, there are many efficient algorithms that solve $Q|pmtn|C_{max}$. For example, $Q|pmtn|C_{max}$ can be solved in time $O(\bar{n}^2\bar{m})$ by a greedy algorithm (Brucker, 2007). At each time point t, the algorithm (called the *level algorithm*) assigns jobs to the machines in a way such that the greater the remaining workload of a job, the faster the machine it is assigned to.[5] However, this algorithm in some cases generates a schedule that has as many as $\bar{m}(\bar{m}-1)/2$ preemptive break points. Therefore, we turn to the algorithm by Gonzalez and Sahni (1978), which runs in time $O(\bar{n} + \bar{m}\log\bar{n})$ using at most $2(\bar{m}-1)$ preemptions. Gonzalez and Sahni also showed that this bound is tight. We note that one preemptive break point corresponds to at least two preemptions, and in the instances that were used to show that the $2(\bar{m}-1)$ bound is tight, $\bar{m}-1$ preemptive break points are required. Therefore, we immediately have the following lemma.

Lemma 5.4.9. *The number of preemptive break points in the solution obtained by the algorithm of Gonzalez and Sahni (1978) is at most $\bar{m}-1$. Furthermore, this bound is tight.*

We note that $\bar{m} = m - 1$. Hence, combining Lemma 5.4.8 and Lemma 5.4.9, we

[5] The greedy algorithm of Zuckerman et al. (2009) is effectively a discrete-time version of the level algorithm.

have the following theorem, which is our main result.

Theorem 5.4.10. *Algorithm 2 runs in polynomial time and*

1. *if the algorithm returns* false*, then there is no successful manipulation (even for the WCMd version of the instance);*

2. *otherwise, the algorithm returns a successful manipulation for a modified set of manipulators, consisting of the original manipulators plus at most $m - 2$ additional manipulators, each with weight at most $W/2$.*

5.5 Algorithms for UCM and UCO

We now consider the case where votes are unweighted. UCMd and UCOd can be solved using Algorithm 1. As for UCM/UCO, every manipulator's weight is one (so that $W = 1$), and we are only allowed to add new manipulators whose weight is also 1. We recall that increasing the weights of the manipulators never prevents c from winning. Therefore, in the context of UCM/UCO we use a slight modification of Algorithm 2, by adding one unweighted manipulator whenever Algorithm 2 proposes adding a weighted manipulator (whose weight can be at most 1/2).

Algorithm 3: compWCM

```
This algorithm is the same as Algorithm 1, except for the following two lines:
3  Use the algorithm in Gonzalez and Sahni (1978) to solve the scheduling
   problem.
16 Let manipulator $i$ vote for $V_1^i$; for any $j > 1$, we add a new manipulator who
   votes for $V_j^i$.
```

The following corollary immediately follows from Theorem 5.4.10.

Corollary 5.5.1. *For UCM, if Algorithm 3 returns* false*, then there is no successful manipulation; otherwise, Algorithm 3 returns a successful manipulation with at most $m - 2$ additional manipulators.*

Recall that Lines 1-3 of Algorithm 3 compute the minimum makespan w (the solution to COd) of the scheduling problem that is obtained from the UCM instance. It is easy to see that if votes are divisible then $\lceil w \rceil$ is the minimum number of unweighted manipulators required to make c win the election, that is, $\lceil w \rceil$ is the optimal solution to UCOd. Therefore, Algorithm 1 can easily be modified to yield an algorithm that solves UCOd. We further note that Algorithm 3 is an approximation algorithm for UCO, as the number of manipulators returned by Algorithm 3 is no more than $\lceil w \rceil + m - 2$. Put another way, Algorithm 3 returns a solution to UCO (with indivisible votes) that approximates the optimal solution to UCOd (with divisible votes) to an additive term of $m - 2$.

Generally, if there exists a successful manipulation, then Algorithm 3 returns a manipulation with additional manipulators. However, there are some special positional scoring voting rules under which UCM can always be solved exactly by Algorithm 1. Given $l \in \{1, \ldots, m-1\}$, the l-approval rule is the scoring rule where $\vec{s}_m(1) = \ldots = \vec{s}_m(l) = 1$ and $\vec{s}_m(l+1) = \ldots = \vec{s}_m(m) = 0$. For example, Plurality (with scoring vector $(1, 0, \ldots, 0)$) and Veto (with scoring vector $(1, \ldots, 1, 0)$) are 1-approval and $(m-1)$-approval, respectively. We note that UCM under any l-approval rule reduces to the scheduling problem in which all machines have the same speed. This corresponds exactly to the scheduling problem $P|pmtn|C_{max}$ in discrete time (that is, the preemptions are allowed only at integer time points), which has a polynomial-time algorithm: *Longest Remaining Processing Time first (LRPT)* Pinedo (2008). Therefore, if we modify Algorithm 3 by solving the reduced scheduling instance with LRPT, then we can solve UCM under any k-approval voting rule in polynomial time.[6] To summarize:

Corollary 5.5.2. *Let $l \in \{1, \ldots, m-1\}$. UCM/UCO for l-approval is in* P.

[6] The simple observation that UCM is in P for approval voting rules was also recently made by Andrew Lin (via personal communication), who employed a completely different (greedy) approach.

5.5.1 On The Tightness of The Results

We presently wish to argue that we have made the most of our technique. The next theorem states that the $m-2$ bound is tight in terms of the difference between the optimal solution to UCO and the optimal solution to UCOd under the same input. It also implies that Algorithm 3 is optimal in the sense that for any $q < m-2$, there is no approximation algorithm for UCO that always outputs a manipulation with at most q manipulators more than the optimal solution to UCOd. This result can be seen as a new type of integrality gap, which applies to our special flavor of rounding.

Theorem 5.5.3. *For any $m \geqslant 3$, there exists a UCO instance such that the (additive) gap between the optimal solution to UCOd and the optimal solution to UCO is $m-2$.*

Proof. For any $m \geqslant 3$, we let the scoring vector be $(m(m-1)(m-2)-1, \ldots, m(m-1)(m-2)-1, m(m-1)(m-2)-2, 0)$. Let $V = [c_1 \succ \ldots \succ c_{m-1} \succ c]$, and let π be the cyclic permutation on $\mathcal{C}\backslash\{c\}$, that is, $\pi : c_1 \rightarrow \ldots \rightarrow c_{m-1} \rightarrow c_1$. For any $i \leqslant m-1$, let V_i be the linear order over $\mathcal{C}$ in which c is ranked in position $(m-1)$, and $\pi^i(c_1) \succ_{V_i} \pi^i(c_2) \succ_{V_i} \ldots \succ_{V_i} \pi^i(c_{m-1})$. Let $P = (V, V_1, \ldots, V_{m-1})$, $P^{NM} = P \cup \pi(P) \cup \ldots \cup \pi^{m-2}(P)$. It follows that for any $i \leqslant m-1$, $s(P^{NM}, c_i) - s(P^{NM}, c) = (m-1)^2 - 1$. Let $V' = [c \succ c_1 \ldots \succ c_{m-1}]$; it can be verified that the divisible vote

$$\frac{1}{m-1}(V', \pi(V'), \pi^2(V'), \ldots, \pi^{m-2}(V'))$$

is sufficient to make c win, hence the optimal solution to UCOd is 1.

We next prove that the solution to UCO is $m-1$. Clearly the following profile is a successful manipulation.

$$(V', \pi(V'), \pi^2(V'), \ldots, \pi^{m-1}(V'))$$

Hence, it remains to show that the solution is at least $m-1$. For the sake of contradiction we assume that the solution is $m-2$, and P^M is the corresponding successful manipulation. Therefore, there must exist $i \leqslant m-1$ such that c_i is not ranked at the bottom of any of the votes of P^M. Therefore,

$$s(P^M, c) - s(P^M, c_i) \leqslant m - 2 < (m-1)^2 - 1,$$

which means that $s(P^{NM} \cup P^M, c) - s(P^{NM} \cup P^M, c_i) < 0$. This contradicts the assumption that P^M is a successful manipulation. □

We next ask the following natural question: is it possible to improve the rounding technique so that the algorithm achieves a better bound, relative to the optimal solution for the indivisible case? This is not ruled out by Theorem 5.5.3, since that theorem compares to the optimal UCOd solution rather than the optimal UCO solution. Nevertheless, the answer is negative, as long as all linear orders in an optimal solution to the COd problem appear in the output of the algorithm. We say that an approximation algorithm $\mathcal{I}$ for UCO is *based on COd* if for any UCO instance, there exists an optimal solution to COd such that every linear order that appears in that solution also appears in the output of $\mathcal{I}$ (as a fraction of the vote of a manipulator).

Theorem 5.5.4. *Let $\mathcal{I}$ be an approximation algorithm based on COd. For any $m \geqslant 3$, there exists a UCO instance such that the gap between the optimal solution to UCO and the output of $\mathcal{I}$ is $m-2$.*

Proof. For any $m \geqslant 3$, we construct an instance such that the solution to the UCO problem is 1, but at least $m-1$ linear orders appear in any optimal solution to the COd problem (so the gap is $m-2$).

We let the scoring vector be $(m+2, 1, 0, \ldots, 0)$. Let

$$V = [c \succ c_1 \succ \ldots \succ c_{m-1}],$$

and

$$V' = [c_{m-1} \succ c_1 \succ c \succ c_2 \succ \ldots \succ c_{m-2}].$$

Furthermore, let

$$\pi : c_1 \to c_2 \to \ldots \to c_{m-1} \to c_1,$$

and

$$\pi^* : c \to c_1 \to \ldots \to c_{m-1} \to c.$$

We define preference profiles by letting

$$P = (V', V, \pi^*(V), (\pi^*)^2(V), \ldots, (\pi^*)^{m-2}(V))$$

$$\text{and } P^{NM} = P \cup \pi(P) \cup \ldots \cup \pi^{m-2}(P).$$

We have that $s(P, c) = m + 2$, $s(P, c_1) = m + 4$, and for any $2 \leqslant i \leqslant m - 1$, $s(P, c_i) = m + 3$. Therefore, $s(P^{NM}, c) = (m + 2)(m - 1)$ and for any $2 \leqslant i \leqslant m - 1$, $s(P^{NM}, c_i) = (m + 3)(m - 1) + 1$. Therefore, for any $i \leqslant m - 1$, $s(P^{NM}, c_i) - s(P^{NM}, c) = m$. It follows that one manipulator suffices to make c the winner (by voting $c \succ c_1 \succ \ldots \succ c_{m-1}$).

On the other hand, the minimum weight for COd is $(m - 1)/m$, for example,

$$V^M = \frac{m-1}{m}(\frac{1}{m-1}V + \frac{1}{m-1}\pi(V) + \ldots + \frac{1}{m-1}\pi^{m-2}(V)).$$

In any manipulator's vote corresponding to the minimum total weight, every alternative except c must appear in the second position for a fraction of the vote. Therefore, any algorithm based on COd must output at least $m - 1$ linear orders. □

5.6 Summary

In this chapter, we extensively examined how strong computational complexity is as a barrier against manipulation. Most results in this chapter are negative. In Section 5.1 we showed that (roughly) for all generalized scoring rules, if the number of manipulators is $o(n^\alpha)$ for some $\alpha < 1/2$, then the probability that these manipulators can succeed goes to 0 as n goes to infinity; however, if the number of manipulators is

$\omega(n^\alpha)$ for some $\alpha > 1/2$, then the probability that these manipulators are all-powerful goes to 1 as n goes to infinity (except they cannot make alternatives against which the nonmanipulators are systematically biased win). We note that as n goes to infinity, $\sqrt{n}/n$ goes to zero.

This "dichotomy" result implies that when the total number of voters is large, even if the number of manipulators is very small compared to the number of nonmanipulators, the manipulators can still manipulate the winner with a high probability. We further gave an axiomatization in Section 5.3, which tells us how general the class of GSRs is—it is the class of all voting rules that satisfies anonymity, homogeneity, and finite local consistency.

Section 5.4 aimed at directly designing (approximation) algorithms for a number of coalitional manipulation problems for positional scoring rules. Built on top of a novel connection between coalitional manipulation problems and scheduling problems, we proposed polynomial-time algorithms that solve WCMd and COd. We also used these algorithms plus a rounding technique to obtain approximation algorithms for WCM, UCM, and UCO, with an additive error bound of $m-2$, which is tight in a sense.

Therefore, it seems that computational complexity is not a very strong barrier against manipulation. An obvious next step is to look for other ways to prevent manipulation. Note that one assumption made in all previous manipulation settings is that the manipulators have full information about the votes of the nonmanipulators. Therefore, a natural question to ask is: What if the manipulators do not have full information about the other voters' votes? The work in the next chapter is motivated by this question. We will study the case of one manipulator with limited information about other voters' votes. We will prove that restricting the information of the manipulator can effectively make a certain type of manipulation, which we call *dominating manipulation*, NP-hard. At one extreme, if the manipulator knows nothing, many voting rules are immune to dominating manipulations. These results seem very natural at a high level, but to obtain them, we need a formal model to analyze voters' strategic behavior.

6

Preventing Manipulation by Restricting Information

It was shown in the last chapter that computational complexity does not seem to be a very strong barrier against manipulation. Consequently, we need to look for new barriers. In this chapter we examine some preliminary ideas to prevent manipulation for the cases where there is one manipulator, by restricting the manipulator's information about the other voters' votes. We recall that in all previously studied manipulation problems, it is normally assumed that the manipulator has full information about the votes of the non-manipulators. The argument often given is that if it is NP-hard with full information, then it only can be at least as computationally difficult with partial information. However, when there is only one manipulator, computing a manipulation is in P for most common voting rules, including all positional scoring rules, Copeland, maximin, and voting trees (see Table 3.1). The only known exceptions are STV (Bartholdi and Orlin, 1991), ranked pairs (Xia et al., 2009), and Nanson's and Baldwin's rules (Narodytska et al., 2011). It is not clear whether it is computationally easy for a single manipulator to find a manipulation when she only has partial information for other rules.

In this chapter, we first model how one manipulator computes a manipulation based on partial information about the other votes. For example, the manipulator may know that some voters prefer one alternative to another, but might not be able to know all pairwise comparisons for all voters. We suppose the knowledge of the manipulator is described by an *information set* E. This is some subset of possible profiles of the non-manipulators which is known to contain the true profile. Given an information set and a pair of votes U and V, if for every profile in E, the manipulator is not worse off voting U than voting V, and there exists a profile in E such that the manipulator is strictly better off voting U, then we say that U *dominates* V. If there exists a vote U that dominates the true preferences of the manipulator then the manipulator has an incentive to vote untruthfully. We call this a *dominating manipulation.* If there is no such vote, then a cautious manipulator might have little incentive to vote strategically.

We are interested in whether a voting rule r is *immune* to dominating manipulations, meaning that a voter's true preferences are never dominated by another vote. If r is not immune to dominating manipulations, we are interested in whether r is *resistant*, meaning that computing whether a voter's true preferences are dominated by another vote U is NP-hard, or *vulnerable*, meaning that this problem is in P. These properties depend on both the voting rule and the form of the partial information. Interestingly, it is not hard to see that most voting rules are immune to manipulation when the partial information is just the current winner. For instance, with any majority consistent rule (for example, plurality), a risk averse manipulator will still want to vote for her most preferred alternative. This means that the chairman does not need to keep the current winner secret to prevent such manipulations. On the other hand, if the chairman lets slip more information, many rules stop being immune. With most scoring rules, if the manipulator knows the current scores, then the rule is no longer immune to such manipulation. For instance, when her most preferred alternative is too far behind to win, the manipulator might vote instead for a less preferred candidate who can win.

In this chapter, we focus on the case where the partial information is represented by a profile P_{po} of partial orders, and the information set E consists of all linear orders that extend P_{po}. The dominating manipulation problem is related to the *possible/necessary winner* problems, which I have briefly talked about in Section 1.6 and Section 2.3. We recall that in possible/necessary winner problems, we are given an alternative c and a profile of partial orders P_{po} that represents the partial information of the voters' preferences. We are asked whether c is the winner for *some* extension of P_{po} (that is, c is a *possible winner*), or whether c is the winner for *every* extension of P_{po} (that is, c is a *necessary winner*). We note that in the possible/necessary winner problems, there is no manipulator and P_{po} represents the chair's partial information about the votes. In dominating manipulation problems, P_{po} represents the partial information of the manipulator about the non-manipulators.

In the following sections, we start with the special case where the manipulator has complete information. In this setting the dominating manipulation problem reduces to the standard manipulation problem, and many common voting rules are vulnerable to dominating manipulation (from known results). When the manipulator has no information, we show that a wide range of common voting rules are immune to dominating manipulation. When the manipulator's partial information is represented by partial orders, our results are summarized in Table 6.1.

Our results are encouraging. For most voting rules r we study in this paper (except plurality and veto), hiding even a little information makes r resistant to dominating manipulation. If we hide all information, then r is immune to dominating manipulation. Therefore, limiting the information available to the manipulator

[1] All hardness results hold even when the number of undetermined pairs in each partial order is no more than a constant.

Table 6.1: Computational complexity of the dominating manipulation problems with partial orders, for common voting rules.

	DOMINATING MANIPULATION
STV	Resistant (Proposition 6.2.2)
Ranked pairs	Resistant (Proposition 6.2.2)
Borda	Resistant (Theorem 6.3.1)
Copeland	Resistant (Corollary 6.3.7)
Voting trees	Resistant (Corollary 6.3.7)
Maximin	Resistant (Theorem 6.3.8)
Plurality	Vulnerable (Algorithm 5)
Veto	Vulnerable (Similar to plurality)

appears to be a promising way to prevent strategic voting.

6.1 Framework for Manipulation with Partial Information

We now introduce the framework of this paper. In this chapter, we suppose there are $n-1 \geqslant 1$ non-manipulators and one manipulator to make notion easier. The information the manipulator has about the votes of the non-manipulators is represented by an *information set* E. The manipulator knows for sure that the profile of the non-manipulators is in E. However, the manipulator does not know exactly which profile in E it is. Usually E is represented in a compact way. Let $\mathcal{I}$ denote the set of all possible information sets in which the manipulator may find herself.

Example 6.1.1. *Suppose the voting rule is r.*

- *If the manipulator has no information, then the only information set is $E = \mathcal{F}_{n-1}$. Therefore $\mathcal{I} = \{\mathcal{F}_{n-1}\}$. Here we recall that $\mathcal{F}_{n-1}$ is the set of all $(n-1)$-profiles*
- *If the manipulator has complete information, then $\mathcal{I} = \{\{P\} : P \in \mathcal{F}_{n-1}\}$.*
- *If the manipulator knows the current winner (before the manipulator votes), then the set of all information sets the manipulator might know is $\mathcal{I} = \{E_1, E_2, \ldots, E_m\}$, where for any $i \leqslant m$, $E_i = \{P \in \mathcal{F}_{n-1} : r(P) = c_i\}$.*

Let V_M denote the true preferences of the manipulator. Given a voting rule r

and an information set E, we say that a vote U *dominates* another vote V, if for every profile $P \in E$, we have $r(P \cup \{U\}) \succeq_{V_M} r(P \cup \{V\})$, and there exists $P' \in E$ such that $r(P' \cup \{U\}) \succ_{V_M} r(P' \cup \{V\})$. In other words, when the manipulator only knows the voting rule r and the fact that the profile of the non-manipulators is in E (and no other information), voting U is a strategy that dominates voting V. We define the following two decision problems.

Definition 6.1.2. *Given a voting rule r, an information set E, the true preferences V_M of the manipulator, and two votes V and U, we are asked the following two questions.*

- *Does U dominate V? This is the* DOMINATION *problem.*
- *Does there exist a vote V' that dominates V_M? This is the* DOMINATING MANIPULATION *problem.*

We stress that usually E is represented in a compact way, otherwise the input size would already be exponentially large, which would trivialize the computational problems. Given a set $\mathcal{I}$ of information sets, we say a voting rule r is *immune* to dominating manipulation, if for every $E \in \mathcal{I}$ and every V_M that represents the manipulator's preferences, V_M is not dominated; r is *resistant* to dominating manipulation, if DOMINATING MANIPULATION is NP-hard (which means that r is not immune to dominating manipulation, assuming $\mathsf{P}\neq\mathsf{NP}$); and r is *vulnerable* to dominating manipulation, if r is not immune to dominating manipulation, and DOMINATING MANIPULATION is in P.

6.2 Manipulation with Complete/No Information

In this section we focus on the following two special cases: (1) the manipulator has complete information, and (2) the manipulator has no information. It is not hard to see that when the manipulator has complete information, DOMINATING MANIPULA-

TION coincides with the standard manipulation problem. Therefore, our framework of dominating manipulation is an extension of the traditional manipulation problem, and we immediately obtain the following proposition from the Gibbard-Satterthwaite theorem (Gibbard, 1973; Satterthwaite, 1975).

Proposition 6.2.1. *When $m \geqslant 3$ and the manipulator has full information, a voting rule satisfies non-imposition and is immune to dominating manipulation if and only if it is a dictatorship.*

The following proposition directly follows from the computational complexity of the manipulation problems for some common voting rules (Bartholdi et al., 1989a; Bartholdi and Orlin, 1991; Conitzer et al., 2007; Zuckerman et al., 2009; Xia et al., 2009).

Proposition 6.2.2. *When the manipulator has complete information, STV, ranked pairs, Nanson's and Baldwin's rules are resistant to* DOMINATING MANIPULATION*; all positional scoring rules, Copeland, voting trees, and maximin are vulnerable to dominating manipulation.*

Next, we investigate the case where the manipulator has no information. We obtain the following positive results.

Theorem 6.2.3. *When the manipulator has no information, any Condorcet consistent voting rule r is immune to dominating manipulation.*

Proof. For the sake of contradiction, let U dominates V_M. Because $U \neq V_M$, there exist two alternatives a and b such that $a \succ_{V_M} b$ and $b \succ_U a$. We prove the theorem in the following two cases.

Case 1: $n-1$ is even. For any j such that $1 \leqslant j \leqslant (n-1)/2$, we let $V_{2j-1} = [a \succ b \succ (\mathcal{C}\backslash\{a,b\})]$, where the alternatives in $\mathcal{C}\backslash\{a,b\}$ are ranked according to the ascending order of their subscripts; let $V_{2j} = [b \succ a \succ \text{Rev}(\mathcal{C}\backslash\{a,b\})]$. Here $\text{Rev}(\mathcal{C}\backslash\{a,b\})$ is the reverse of $\mathcal{C}\backslash\{a,b\}$. Let $P = (V_1, \ldots, V_{n-1x})$. It follows that a is the Condorcet winner for $P \cup \{V_M\}$ and b is the Condorcet winner for $P \cup \{U\}$. Because $a \succ_{V_M} b$, V_M is not dominated by U, which contradicts the assumption.

Case 2: $n-1$ is odd. For any j such that $1 \leqslant j \leqslant (n-2)/2$, we let $V_{2j-1} = [a \succ b \succ (\mathcal{C}\backslash\{a,b\})]$ and $V_{2j} = [b \succ a \succ \text{Rev}(\mathcal{C}\backslash\{a,b\})]$. Suppose $a = c_{i_1}$ and $b = c_{i_2}$. Let $V_{n-1} = \begin{cases} V_1 & \text{if } i_1 > i_2 \\ V_2 & \text{if } i_1 < i_2 \end{cases}$. Let $P = (V_1, \ldots, V_{n-1})$. It follows that a is the Condorcet winner for $P \cup \{V_M\}$ and b is the Condorcet winner for $P \cup \{U\}$, which contradicts the assumption. □

Theorem 6.2.4. *When the manipulator has no information, Borda is immune to dominating manipulation.*

Proof. For the sake of contradiction, let U dominates V_M. Because $U \neq V_M$, there exists $i^* \leqslant m$ such that $\text{Alt}(V_M, i^*) \neq \text{Alt}(U, i^*)$ and for every $i < i^*$, $\text{Alt}(V_M, i) = \text{Alt}(U, i)$. That is, i^* is the first position from the top where the alternatives in V_M and U are different. Let $c_{i_1} = \text{Alt}(V_M, i^*)$ and $c_{i_2} = \text{Alt}(U_M, i^*)$. We prove the theorem in the following three cases.

Case 1: $n-1$ is even. For any $i < i' \leqslant m$, let $V_M^{[i,i']}$ denote the sub-linear-order of V_M that starts at the ith position of V_M and ends at the i'th position of V_M. For any j such that $1 \leqslant j \leqslant (n-1)/2$, we let $V_{2j-1} = [V_M^{[i*,m]} \succ \text{Rev}(V_M^{[1,i*-1])}]$ and $V_{2j} = [\text{Rev}(V_M^{[i*,m]}) \succ \text{Rev}(V_M^{[1,i*-1])}]$. Let $P = (V_1, \ldots, V_{n-1})$. It follows that $\text{Borda}(P \cup \{V_M\}) = c_{i_1}$ and $\text{Borda}(P \cup \{U\}) = c_{i_2}$. We note that $c_{i_1} \succ_{V_M} c_{i_2}$, which contradicts the assumption.

Case 2: $n-1$ is odd and c_1 is ranked within top i^* positions in V_M. For any j such that $1 \leqslant j \leqslant (n-2)/2$, we let $V_{2j-1} = [c_1 \succ c_2 \succ \cdots \succ c_m]$ and $V_{2j} = [c_m \succ c_{m-1} \succ \cdots \succ c_1]$. Let $V_{n-1} = \text{Rev}(V)$ and $P = (V_1, \ldots, V_{n-1})$. It follows that $\text{Borda}(P \cup \{V\}) = c_1$ and $\text{Borda}(P \cup \{U\}) \neq c_1$, which contradicts the assumption.

Case 3: $n - 1$ is odd and c_1 is not ranked within top i^* positions in V_M. Let $V_1, \ldots, V_{n-2}$ be defined the same as in Case 2. Let $V' = [V_M^{[i^*,m]} \succ \text{Rev}(V_M^{[1,i^*-1]})]$. Let $U' = [U^{[i^*,m]} \succ \text{Rev}(U^{[1,i^*-1]})]$. It follows that $\text{Borda}(V', V_M) = c_{i_1}$. Let $a = \text{Borda}(V', U)$. If $a \neq c_{i_1}$, then $c_{i_1} \succ_{V_M} a$. This is because the alternatives ranked within top $i^* - 1$ positions in V_M gets exactly the average score in $\{V', U\}$, which means that in order for any of them to win, the scores of all alternative in $\{V', U\}$ must be the same. However, due to the tie-breaking mechanism, the winner is c_1, which contradicts the assumption that c_1 is not ranked within top i^* positions in V_M. Let $P' = (V_1, \ldots, V_{n-2}, V')$, we have that $\text{Borda}(P' \cup \{V_M\}) = c_{i_1} \succ_{V_M} a = \text{Borda}(P' \cup \{U\})$, which contradicts the assumption. If $a = c_{i_1}$, then $\text{Borda}(U', V_M) = \text{Borda}(V', U) = a = c_{i_1}$. Let $P^* = (V_1, \ldots, V_{n-2}, U')$. We have $\text{Borda}(P^* \cup \{V_M\}) = c_{i_1} \succ_{V_M} c_{i_2} = \text{Borda}(P^* \cup \{U\})$, which is a contradiction.

Therefore, the theorem is proved. □

Theorem 6.2.5. *When the manipulator has no information and $n \geqslant 6(m-2)+1$, any positional scoring rule is immune to dominating manipulation.*

Proof. For the sake of contradiction, let U dominates V_M. Let $c = \arg\max_{c*}\{\vec{s}_m(V_M, c^*) : \vec{s}_m(V_M, c^*) > \vec{s}_m(U, c^*)\}$. It follows that there exists an alternative c' such that $\vec{s}_m(V_M, c') < \vec{s}_m(V_M, c)$ and $\vec{s}_m(U, c') = \vec{s}_m(V_M, c)$. It follows that $s_m(V_M, c) > \vec{s}_m(V_M, c')$ and $\vec{s}_m(U, c') = \vec{s}_m(V_M, c) > \vec{s}_m(U, c)$.

We prove the theorem for the case where $c = c_1$ and $c' = c_2$. The other cases can be proved similarly. Let M_{m-2} denote the cyclic permutation such that $c_3 \to c_4 \to \cdots \to c_m \to c_3$. For any $k \in \mathbb{N}$ and any $c \in \mathcal{C} \backslash \{c_1, c_2\}$, we let $M_{m-2}^0(c) = c$ and $M_{m-2}^k(c) = M(M_{m-2}^{k-1}(c))$. Let $W = [c_1 \succ c_2 \succ c_3 \succ \cdots \succ c_m]$ and $W' = [c_2 \succ c_1 \succ c_3 \succ \cdots \succ c_m]$. Let P_1 denote the $6(m-2)$-profile that is composed of three copies of $\{W, W', M_{m-2}(W), M_{m-2}(W'), \ldots, M_{m-2}^{m-3}(W), M_{m-2}^{m-3}(W)\}$.

If $n-1$ is even, then let P be composed of P_1 plus $(n-1)/2 - 3(m-2)$ copies of $\{W, W'\}$. If $n-1$ is odd, then let W^* denote the a vote obtained from V_M by exchanging the positions of c and c' and let P be composed of $P_1 \cup \{W^*\}$ plus $\lfloor (n-1)/2 \rfloor - 3(m-2)$ copies of $\{W, W'\}$. Because $\vec{s}_m(1) > \vec{s}_m(m)$, we have that $r(P \cup \{V_M\}) = c_1$ and $r(P \cup \{U\}) = c_2$. We note that $c_1 \succ_{V_M} c_2$. Therefore, we obtain a contradiction, which means that V_M is not dominated. □

These results demonstrate that the information that the manipulator has about the votes of the non-manipulators plays an important role in determining strategic behavior. When the manipulator has complete information, many common voting rules are vulnerable to dominating manipulation, but if the manipulator has no information, then many common voting rules become immune to dominating manipulation.

6.3 Manipulation with Partial Orders

In this section, we study the case where the manipulator has partial information about the votes of the non-manipulators. We suppose the information is represented by a profile P_{po} composed of partial orders. That is, the information set is $E = \{P \in \mathcal{F}_n : P \text{ extends } P_{po}\}$. We note that the two cases discussed in the previous section (complete information and no information) are special cases of manipulation with partial orders. Consequently, by Proposition 6.2.1, when the manipulator's information is represented by partial orders and $m \geqslant 3$, no voting rule that satisfies non-imposition and non-dictatorship is immune to dominating manipulation. It also follows from Theorem 6.2.4 that STV and ranked pairs are resistant to dominating manipulation. The next theorem states that even when the manipulator only misses a tiny portion of the information, Borda becomes resistant to dominating manipulation.

Theorem 6.3.1. *DOMINATION and DOMINATING MANIPULATION with partial orders are NP-hard for Borda, even when the number of unknown pairs in each vote is no more than* 4.

Proof. We only prove that DOMINATION is NP-hard, via a reduction from EXACT COVER BY 3-SETS (X3C). The proof for DOMINATING MANIPULATION is similar to the proof of the NP-hardness of the possible winner problems under positional scoring rules in Xia and Conitzer (2011a).

In an X3C instance, we are given two sets $\mathcal{V} = \{v_1, \ldots, v_q\}$, $\mathcal{S} = \{S_1, \ldots, S_t\}$, where for any $j \leqslant t$, $S_j \subseteq \mathcal{V}$ and $|S_j| = 3$. We are asked whether there exists a subset $\mathcal{S}'$ of $\mathcal{S}$ such that each element in $\mathcal{V}$ is in exactly one of the 3-sets in $\mathcal{S}'$. We construct a DOMINATION instance as follows.

Alternatives: $\mathcal{C} = \{c, w, d\} \cup \mathcal{V}$, where d is an auxiliary alternative. Therefore, $m = |\mathcal{C}| = q + 3$. Ties are broken in the following order: $c \succ w \succ \mathcal{V} \succ d$.

Manipulator's preferences and possible manipulation: $V_M = [w \succ c \succ d \succ \mathcal{V}]$. We are asked whether $V = V_M$ is dominated by $U = [w \succ d \succ c \succ \mathcal{V}]$.

The profile of partial orders: Let $P_{po} = P_1 \cup P_2$, defined as follows.

First part (P_1) of the profile: For each $j \leqslant t$, We define a partial order O_j as follows.

$O_j = [w \succ S_j \succ d \succ \text{Others}] \backslash [\{w\} \times (S_j \cup \{d\})]$

That is, O_j is a partial order that agrees with $w \succ S_j \succ d \succ$ Others, except that the pairwise relations between (w, S_j) and (w, d) are not determined (and these are the only 4 unknown relations). Let $P_1 = \{O_1, \ldots, O_t\}$.

Second part (P_2) of the profile: We first give the properties that we need P_2 to satisfy, then show how to construct P_2 in polynomial time. All votes in P_2 are linear orders that are used to adjust the score differences between alternatives. Let

$P_1' = \{w \succ S_i \succ d \succ \text{Others} : i \leqslant t\}$. That is, P_1' ($|P_1'| = t$) is an extension of P_1 (in fact, P_1' is the set of linear orders that we started with to obtain P_1, before removing some of the pairwise relations). Let $\vec{s}_m = (m-1, \ldots, 0)$. P_2 is a set of linear orders such that the following holds for $Q = P_1' \cup P_2 \cup \{V\}$:

(1) For any $i \leqslant q$, $\vec{s}_m(Q, c) - \vec{s}_m(Q, v_i) = 1$, $\vec{s}_m(Q, w) - \vec{s}_m(Q, c) = 4q/3$.

(2) For any $i \leqslant q$, the scores of v_i and w, c are higher than the score of d in any extension of $P_1 \cup P_2 \cup \{V\}$ and in any extension of $P_1 \cup P_2 \cup \{U\}$.

(3) The size of P_2 is polynomial in $t + q$.

We now show how to construct P_2 in polynomial time. For any alternative $a \neq d$, we define the following two votes: $W_a = \{[a \succ d \succ \text{Others}], [\text{Rev(Others)} \succ a \succ d]\}$, where Rev(Others) is the reversed order of the alternatives in $\mathcal{C} \backslash \{a, d\}$. We note that for any alternative $a' \in \mathcal{C} \backslash \{a, d\}$, $\vec{s}_m(W, a) - \vec{s}_m(W, a') = 1$ and $\vec{s}_m(W, a') - \vec{s}_m(W, d) = 1$. Let $Q_1 = P_1' \cup \{V\}$. P_2 is composed of the following parts:

(1) $tm - \vec{s}_m(Q_1, c)$ copies of W_c.

(2) $tm + 4q/3 - \vec{s}_m(Q_1, w)$ copies of W_w.

(2) For each $i \leqslant q$, there are $tm - 1 - \vec{s}_m(Q_1, v_i)$ copies of W_{v_i}.

We next prove that V is dominated by U if and only if c is the winner in at least one extension of $P_{po} \cup \{V\}$. We note that for any $v \in \mathcal{V} \cup \{w\}$, the score of v in V is the same as the score of v in U. The score of c in U is lower than the score of c in V. Therefore, for any extension P^* of P_{po}, if $r(P^* \cup \{V\}) \in (\{w\} \cup \mathcal{V})$, then $r(P^* \cup \{V\}) = r(P^* \cup \{U\})$ (because d cannot win). Hence, for any extension P^* of P_{po}, voting U can result in a different outcome than voting V only if $r(P^* \cup V) = c$. If there exists an extension P^* of P_{po} such that $r(P^* \cup \{V\}) = c$, then we claim that the manipulator is strictly better off voting U than voting V. Let P_1^* denote the extension of P_1 in P^*. Then, because the total score of w is no more than the total score of c, w is ranked lower than d at least $\frac{q}{3}$ times in P_1^*. Meanwhile, for each $i \leqslant q$, v_i is not ranked higher than w more than one time in P_1^*, because otherwise the total score of v_i will be strictly higher than the total score of c. That is, the votes in P_1^* where $d \succ w$ make up a solution to the X3C instance. Therefore, the only possibility for c to win is for the scores of c, w, and all alternatives in $\mathcal{V}$ to be the same (so that c wins according to the tie-breaking mechanism). Now, we have $w = r(P^* \cup \{U\})$. Because $w \succ_{V_M} c$, the manipulator is better off voting U. It follows that V is dominated by U if and only if there exists an extension of $P_{po} \cup \{V\}$ where c is the winner.

The above reasoning also shows that V is dominated by U if and only if the X3C instance has a solution. Therefore, DOMINATION is NP-hard. For the DOMINATING MANIPULATION problem, we add to P_{po} a profile P_E defined as follows. For each $e \in \mathcal{V} \cup \{w\}$ and each $i \leqslant l-1$, we obtain a vote $V_{e,i}$ from V_M by exchanging the alternative ranked in the $(i+1)$th position and e, and then exchanging the alternative ranked in the ith position and d; let $O_{e,i}$ denote the partial order obtained from $V_{e,i}$ by removing $d \succ e$. Let M denote the following cyclic permutation $c \to w \to d \to \mathcal{V} \to A \to c$. Let P_E denote q copies of $\{O_{e,i}, M(V_{e,i}), M(V_{e,i})^2, \ldots, M^{l-1}(V_{e,i}) : e \in \mathcal{V} \cup \{w\}, i \leqslant l-1\}$. We note that in an extension P_E^* of P_E where the extension of $O_{e,i}$ is $V_{e,i}$, then the scores of the alternatives in P_E^* are the same.

For any vote W where there exists $v \in \mathcal{V}$ such that the score difference between w and v is different from the score difference between w and v in V_M, there must exists $v' \in \mathcal{V}$ such that the score difference between w and v' in W is strictly smaller than their score difference in V_M. Then, it is not hard to find an extension of P_{po} such that if the manipulator votes V_M, then w wins, and if the manipulator votes W, then v' wins, which means that V_M is not dominated by W. Therefore, if V_M is dominated by another W, then the score differences between w and the alternatives in $\mathcal{V}$ are the same across V_M and W. Following the same reasoning as for the DOMINATION problem, we conclude that DOMINATING MANIPULATION is NP-hard. □

Theorem 6.3.1 can be generalized to a class of scoring rules similar to the class of rules in Theorem 1 in Xia and Conitzer (2011a), which does not include plurality or veto. In fact, as we will show later, plurality and veto are vulnerable to dominating manipulation.

We now investigate the relationship to the possible winner problem in more depth. In a possible winner problem (r, P_{po}, c), we are given a voting rule r, a profile P_{po} composed of n partial orders, and an alternative c. We are asked whether there exists an extension P of P_{po} such that $c = r(P)$. Intuitively, both DOMINATION and DOMINATING MANIPULATION seem to be harder than the possible winner problem under the same rule. Next, we present two theorems, which show that for any WMG-based rule, DOMINATION and DOMINATING MANIPULATION are harder than two special possible winner problems, respectively.

We first define a notion that will be used in defining the two special possible winner problems. For any instance of the possible winner problem (r, P_{po}, c), we define its *WMG partition* $\mathbb{R} = \{R_{c'} : c' \in \mathcal{C}\}$ as follows. For any $c' \in \mathcal{C}$, let $R_{c'} = \{\text{WMG}(P) : P \text{ extends } P_{po} \text{ and } r(P) = c'\}$. That is, $R_{c'}$ is composed of all WMGs of the extensions of P_{po}, where the winner is c'. It is possible that for some $c' \in \mathcal{C}$, $R_{c'}$ is empty. For any subset $\mathcal{C}' \subseteq \mathcal{C}\backslash\{c\}$, we let $G_{\mathcal{C}'}$ denote the weighted majority graph where for each $c' \in \mathcal{C}'$, there is an edge $c' \rightarrow c$ with weight 2, and these are the only edges in $G_{\mathcal{C}'}$. We are ready to define the two special possible winner problems for WMG-based voting rules.

Definition 6.3.2. *Let d^* be an alternative and let $\mathcal{C}'$ be a nonempty subset of $\mathcal{C}\backslash\{c, d^*\}$. For any WMG-based voting rule r, we let $PW_1(d^*, \mathcal{C}')$ denote the set of possible winner problems (r, P_{po}, c) satisfying the following conditions:*

1. *For any $G \in R_c$, $r(G + G_{\mathcal{C}'}) = d^*$.*

2. *For any $c' \neq c$ and any $G \in R_{c'}$, $r(G + G_{\mathcal{C}'}) = r(G)$.*

3. For any $c' \in \mathcal{C}'$, $R_{c'} = \varnothing$.

We recall that R_c and $R_{c'}$ are elements in the WMG partition of the possible winner problem.

Definition 6.3.3. *Let d^* be an alternative and let $\mathcal{C}'$ be a nonempty subset of $\mathcal{C} \backslash \{c, d^*\}$. For any WMG-based voting rule r, we let $PW_2(d^*, \mathcal{C}')$ denote the problem instances (r, P_{po}, c) of $PW_1(d^*, \mathcal{C}')$, where for any $c' \in \mathcal{C} \backslash \{c, d^*\}$, $R_{c'} = \varnothing$.*

Theorem 6.3.4. *Let r be a WMG-based voting rule. There is a polynomial time reduction from $PW_1(d^*, \mathcal{C}')$ to* DOMINATION *with partial orders, both under r.*

Proof. Let (r, P_{po}, c) be a $\mathrm{PW}_1(d^*, \mathcal{C}')$ instance. We construct the following DOMINATION instance. Let the profile of partial orders be $Q_{po} = P_{po} \cup \{\mathrm{Rev}(d^* \succ c \succ \mathcal{C}' \succ \text{Others})\}$, $V = V_M = [d^* \succ c \succ \mathcal{C}' \succ \text{Others}]$, and $U = [d^* \succ \mathcal{C}' \succ c \succ \text{Others}]$. Let P be an extension of P_{po}. It follows that $\mathrm{WMG}(P \cup \{\mathrm{Rev}(d^* \succ c \succ \mathcal{C}' \succ \text{Others}), V\}) = \mathrm{WMG}(P)$, and $\mathrm{WMG}(P \cup \{\mathrm{Rev}(d^* \succ c \succ \mathcal{C}' \succ \text{Others}), U\}) = \mathrm{WMG}(P) + G_{\mathcal{C}'}$. Therefore, the manipulator can change the winner if and only if $\mathrm{WMG}(P) \in R_c$, which is equivalent to c being a possible winner. We recall that by the definition of $\mathrm{PW}_1(d^*, \mathcal{C}')$, for any $G \in R_c$, $r(G + G_{\mathcal{C}'}) = d^*$; for any $c' \neq c$ and any $G \in R_{c'}$, $r(G + G_{\mathcal{C}'}) = c'$; and $d^* \succ_V c$. It follows that V $(=V_M)$ is dominated by U if and only if the $\mathrm{PW}_1(d^*, \mathcal{C}')$ instance has a solution. □

Theorem 6.3.4 can be used to prove that DOMINATION is NP-hard for Copeland, maximin, and voting trees, even when the number of undetermined pairs in each partial order is bounded above by a constant. It suffices to show that for each of these rules, there exist d^* and $\mathcal{C}'$ such that $\mathrm{PW}_1(d^*, \mathcal{C}')$ is NP-hard. To prove this, we can modify the NP-completeness proofs of the possible winner problems for Copeland, maximin, and voting trees by Xia and Conitzer Xia and Conitzer (2011a).

Corollary 6.3.5. DOMINATION *with partial orders is* NP*-hard for Copeland, maximin, and voting trees, even when the number of unknown pairs in each vote is bounded above by a constant.*

Proof. **Copeland:** We tweak the reduction in the NP-completeness proof of PW w.r.t. Copeland (Xia and Conitzer, 2011a, Theorem 3) by letting $D(c, v) = 1$ for any alternative $v \in \mathcal{V}$ and use the tie-breaking mechanism where $w \succ c \succ$ Others. Let $d^* = w$, $\mathcal{C}' = B$, $V = U = [w \succ c \succ \mathcal{C}' \succ \text{Others}]$ and $W = [w \succ \mathcal{C}' \succ c \succ \text{Others}]$. It follows that the alternatives in B never wins the elections, and if c wins the election in an extension P of P_{po}, then the Copeland score of c is $8t + 1$ and the Copeland score of $w = 8t$. However, in the weighted majority graph $\text{WMG}(P) + G_{\mathcal{C}'}$, c loses to all alternatives in $\mathcal{C}'$ in their pairwise elections, which means that the Copeland score of c is $t + 1$. Consequently w is the winner. On the other hand, for any extension P where c is not the winner, w is the winner, and w is also the winner in the weighted majority graph $\text{WMG}(P) + G_{\mathcal{C}'}$. Therefore, the PW instance is a $\text{PW}_1(d^*, \mathcal{C}')$ instance.

Maximin: We tweak the reduction in the NP-completeness proof of PW w.r.t. maximin (Xia and Conitzer, 2011a, Theorem 5) by letting $D(w', w) = t$. Let $d^* = w$, $\mathcal{C}' = \{w'\}$, $V = U = [w \succ c \succ w' \succ \mathcal{V}]$ and $W = [w \succ w' \succ c \succ \mathcal{V}]$. We adopt the tie-breaking mechanism where $w \succ c \succ \mathcal{V} \succ w'$. It is easy to check that w' never wins the elections. If c wins the election in an extension P of P_{po}, then the minimum pairwise score of c is $-t+2$, and the minimum pairwise score of w and the alternatives in $\mathcal{V}$ are $-t$. We note that in the majority graph $\text{WMG}(P) + G_{\mathcal{C}'}$, the minimum pairwise score of c is $-t$ (against w'), which means that $r(\text{WMG}(P) + G_{\mathcal{C}'}) = w$. For any extension P of P_{po} such that $r(P) \neq c$, it easy to check that the winner is in $\{w\} \cup \mathcal{V}$, and the minimum pairwise scores of them are the same as in the weighted majority graph $\text{WMG}(P) + G_{\mathcal{C}'}$. Therefore, the PW instance is a $\text{PW}_1(d^*, \mathcal{C}')$ instance.

Voting trees: We tweak the reduction in the NP-completeness proof of PW w.r.t. voting trees (Xia and Conitzer, 2011a, Theorem 7) by letting $D(c, d) = 1$. Let $d^* = w$, $\mathcal{C}' = \{d\}$, $V = U = [w \succ c \succ d \succ \text{Others}]$ and $W = [w \succ d \succ c \succ \text{Others}]$. For any extension P of P_{po} where c wins, the winner for the weighted majority graph $\text{WMG}(P) + G_{\mathcal{C}'}$ is w, because c loses to d in the first round, and w beats any other alternatives (except c) in their pairwise elections. For any extension P of P_{po} where c does not win, the winner is w. Therefore, the PW instance is a $\text{PW}_1(d^*, \mathcal{C}')$ instance. □

Theorem 6.3.6. *Let r be a WMG-based voting rule. There is a polynomial-time reduction from $PW_2(d^*, \mathcal{C}')$ to* DOMINATING MANIPULATION *with partial orders, both under r.*

Proof. The proof is similar to the proof for Theorem 6.3.4. We note that d^* is the manipulator's top-ranked alternative. Therefore, if c is not a possible winner, then V $(= V_M)$ is not dominated by any other vote; if c is a possible winner, then V is dominated by $U = [w \succ \mathcal{C}' \succ c \succ \text{Others}]$. □

Similarly, we have the following corollary.

Corollary 6.3.7. DOMINATING MANIPULATION *with partial orders is* NP*-hard for Copeland and voting trees, even when the number of unknown pairs in each vote is bounded above by a constant.*

It is an open question if $\text{PW}_2(d^*, \mathcal{C}')$ with partial orders is NP-hard for maximin. However, we can directly prove that DOMINATING MANIPULATION is NP-hard for maximin by a reduction from X3C.

Theorem 6.3.8. DOMINATING MANIPULATION *with partial orders is* NP*-hard for maximin, even when the number of unknown pairs in each vote is no more than 4.*

Proof. We prove the hardness result by a reduction from X3C. Given an X3C instance $\mathcal{V} = \{v_1, \ldots, v_q\}$, $\mathcal{S} = \{S_1, \ldots, S_t\}$, where $q = t > 3$, we construct a DOMINATING MANIPULATION instance as follows.

Alternatives: $\mathcal{V} \cup \{c, w, w'\}$. Ties are broken in the order $w \succ \mathcal{V} \succ c \succ w'$.

First part P_1 of the profile: for each $i \leqslant t$, we start with the linear order $V_i = [w \succ S_i \succ c \succ (\mathcal{V} \backslash S_i) \succ w']$, and subsequently obtain a partial order O_i by removing the relations in $\{w\} \times (S_i \cup \{c\})$. For each $i \leqslant t$, we let O'_i be a partial order obtained from $V'_i = [w \succ v_i \succ \text{Others}]$ by removing $w \succ v_i$. We let O' be a partial order obtained from $V' = [w' \succ w \succ \text{Others}]$ by removing $w' \succ w$. Let P_1 be the profile composed of $\{O_1, \ldots, O_t\}$, 2 copies of $\{O'_1, \ldots, O'_t\}$, and 3 copies of O'. Let P'_1 denote the extension of P_1 that consists of $V_1, \ldots, V_t$, 2 copies of $\{V'_1, \ldots, V'_t\}$, and 3 copies of V'.

Second part P_2 of the profile: P_2 is defined to be a a set of linear orders such that the pairwise score differences of $P'_1 \cup P_2 \cup \{V\}$ satisfy:

(1) $D(w, c) = 2t + \frac{2q}{3}$, $D(w', w) = 2t + 6$, $D(w', c) = 2t$, and for all $i \leqslant q$, $D(w, v_i) = 2t + 4$ and $D(v_i, w') = 4(t + q)$.

(2) $D(l, r) \leqslant 1$ for all other pairwise scores not defined in (1).

Manipulator's preferences: $V_M = [w \succ \mathcal{V} \succ c \succ w']$.

We note that in any extension of $P_1 \cup P_2$, after the manipulator changes her vote from V_M to $[w \succ \mathcal{V} \succ w' \succ c]$, the only change made to the weighted majority graph is that the weight on $w \rightarrow c$ increases by 2. Since w' never wins in any extension, if c does not win when the manipulator votes for V_M, then the winner does not change after the manipulator changes her vote to $[w \succ \mathcal{V} \succ w' \succ c]$. It follows from the proof of Theorem 6.3.4, Corollary 6.3.5, and Theorem 5 in Xia and Conitzer (2011a) that if the X3C instance has a solution, then V_M is dominated by $U = [w \succ \mathcal{V} \succ w' \succ c]$.

Suppose that the X3C instance does not have a solution, we next show that V_M is not dominated by any vote.

For the sake of contradiction, suppose the X3C instance does not have a solution and V_M is dominated by a vote U. There are following cases.

Case 1: There exist $v_i \in \mathcal{V}$ such that $w \succ_V v_i$ and $v_i \succ_U w$. We let P^* be the extension of $P_1 \cup P_2$ obtained from $P_1' \cup P_2$ as follows. (1) Let $w \succ w'$ in 3 extensions of O' (we recall that there are $q > 3$ copies of O' in P_1). (2) Let $v_i \succ w$ in 2 extensions of O_i'. It is easy to check that in P^*, the minimum pairwise score of w is $-2t$ (via w') and the minimum pairwise score of v_i is $-2t$ (via w). Therefore, due to the tie-breaking mechanism, w wins. However, if the manipulator changes her vote from V_M to U, then the minimum pairwise score of w at most $-2t$ and the minimum pairwise score of v_i is at least $-2t + 2$, which means that v_i wins. We note that $w \succ_V v_i$. This contradicts the assumption that U dominates V_M.

Case 2: $w \succ_W v_i$ for each $v_i \in \mathcal{V}$. By changing her vote from V_M to U, the manipulator might reduce the minimum score of U by 2, increase the minimum score of c by 2, or increase the minimum score of w' by 2. Therefore, by changing her vote to U, the manipulator would either make no changes, make w lose, or make c win (we note that w' is not winning anyway). In each of these three cases the manipulator is not better off, which means that U does not dominate V_M. This contradicts the assumption. □

For plurality and veto, there exist polynomial-time algorithms for both DOMINATION and DOMINATING MANIPULATION. Given an instance of DOMINATION, denoted by (r, P_{po}, V_M, V, U), we say that U is a *possible improvement* of V, if there exists an extension P of P_{po} such that $r(P \cup \{U\}) \succ_{V_M} r(P \cup \{V\})$. It follows that U dominates V if and only if U is a possible improvement of V, and V is not a possible improvement of U. We first introduce an algorithm (Algorithm 4) that checks whether U is a possible improvement of V for plurality.

Let c_{i*} (resp., c_{j*}) denote the top-ranked alternative in V (resp., U). We will check whether there exists $0 \leqslant l \leqslant n$, $d, d' \in \mathcal{C}$ with $d' \succ_{V_M} d$, and an extension P^* of P_{po}, such that if the manipulator votes for V, then the winner is d, whose plurality score in P^* is l, and if the manipulator votes for U, then the winner is d'. We note that if such d, d' exist, then either $d = c_{i*}$ or $d' = c_{j*}$ (or both hold). To this end, we solve multiple maximum-flow problems defined as follows.

Let $\mathcal{C}' \subset \mathcal{C}$ denote a set of alternatives. Let $\vec{e} = (e_1, \ldots, e_m) \in \mathbb{N}^m$ be an arbitrary vector composed of m natural numbers such that $\sum_{i=1}^{m} e_i \geqslant n$. We define a maximum-flow problem $F_{\mathcal{C}'}^{\vec{e}}$ as follows.

Vertices: $\{s, O_1, \ldots, O_n, c_1, \ldots, c_m, y, t\}$.

Edges:

- For any O_i, there is an edge from s to O_i with capacity 1.
- For any O_i and c_j, there is an edge $O_i \to c_j$ with capacity 1 if and only if c_j can be ranked in the top position in at least one extension of O_i.
- For any $c_i \in \mathcal{C}'$, there is an edge $c_i \to t$ with capacity e_i.
- For any $c_i \in \mathcal{C} \backslash \mathcal{C}'$, there is an edge $c_i \to y$ with capacity e_i.
- There is an edge $y \to t$ with capacity $n - \sum_{c_i \in \mathcal{C}'} e_i$.

For example, $F_{\{c_1, c_2\}}^{\vec{e}}$ is illustrated in Figure 6.1.

It is not hard to see that $F_{\mathcal{C}'}^{\vec{e}}$ has a solution whose value is n if and only if there exists an extension P^* of P_{po}, such that (1) for each $c_i \in \mathcal{C}'$, the plurality of c_i is exactly e_i, and (2) for each $c_{i'} \notin \mathcal{C}'$, the plurality of $c_{i'}$ is no more than $e_{i'}$. Now, for any pair of alternatives $d = c_i, d' = c_j$ such that $d' \succ_{V_M} d$ and either $d = c_{i*}$ or $d' = c_{j*}$, we define the set of *admissible maximum-flow problems* A_{Plu}^{l} to be the set

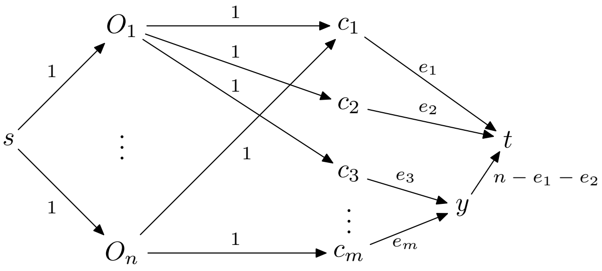


FIGURE 6.1: $F^{\vec{e}}_{\{c_1,c_2\}}$.

of maximum flow problems $F^{\vec{e}}_{c_i,c_j}$ where $e_i = l$, and if $F^{\vec{e}}_{c_i,c_j}$ has a solution, then the manipulator can improve the winner by voting for U. More precisely, we define A^l_{Plu} as follows.

- If $i = i^*$ and $j \neq j^*$, then let $e_i = l$, $e_j = l + 1 - \delta(c_j, c_i)$, and $e_{j*} = \min(l + 1 - \delta(j^*, i), e_j - 1 - \delta(j^*, j))$. For any $c_{i'} \in \mathcal{C} \backslash \{c_i, c_j, c_{j*}\}$, we let $e_{i'} = \min(l + 1 - \delta(i', i), e_j - \delta(i', j))$. Let $A^l_{\text{Plu}} = \{F^{\vec{e}}_{\{c_i,c_j\}}\}$.

- If $i \neq i^*$ and $j = j^*$, then let $e_i = l$, $e_j = l - \delta(c_j, c_i)$, and $e_{i*} = \min(l - 1 - \delta(i^*, i), e_j + 1 - \delta(i^*, j))$. For any $c_{i'} \in \mathcal{C} \backslash \{c_i, c_j, c_{i*}\}$, we let $e_{i'} = \min(l - \delta(i', i), e_j + 1 - \delta(i', j))$. Let $A^l_{\text{Plu}} = \{F^{\vec{e}}_{\{c_i,c_j\}}\}$.

- If $i = i^*$ and $j = j^*$, then we define A^l_{Plu} as follows.
  - Let $e_i = l$, $e_j = l + 1 - 2\delta(c_j, c_i)$. For any $c_{i'} \in \mathcal{C} \backslash \{c_i, c_j\}$, we let $e_{i'} = \min(l + 1 - \delta(i', i), e_j + 1 - \delta(i', j))$.
  - Let $e'_i = e'_j = l$. For any $c_{i'} \in \mathcal{C} \backslash \{c_i, c_j\}$, we let $e'_{i'} = \min(l + 1 - \delta(i', i), e_j + 1 - \delta(i', j))$. Let $\vec{e'} = (e'_1, \ldots, e'_m)$.
  - Let $A^l_{\text{Plu}} = \{F^{\vec{e}}_{\{c_i,c_j\}}, F^{\vec{e'}}_{\{c_i,c_j\}}\}$.

Algorithm 4 solves all maximum-flow problems in A^l_{Plu} to check whether U is a possible improvement of V.

```
Algorithm 4: PossibleImprovement($V$,$U$)
Let $c_{i^*} = \mathrm{Alt}(V,1)$ and $c_{j^*} = \mathrm{Alt}(U,1)$.
for any $0 \leqslant l \leqslant n$ and any pair of alternatives $d = c_i, d' = c_j$ such that
$d' \succ_{V_M} d$ and either $d = c_{i^*}$ or $d' = c_{j^*}$ do
    Compute $A^l_{\mathrm{Plu}}$.
    for each maximum-flow problem $F^{\vec{e}}_{\mathcal{C}'}$ in $A^l_{Plu}$ do
        if $\sum_{c_i \in \mathcal{C}'} e_i \leqslant n$ and the value of maximum flow in $F^{\vec{e}}_{\mathcal{C}'}$ is $n$ then
            Output that the $U$ is a possible improvement of $V$, terminate the
            algorithm.
        end
    end
end
Output that $U$ is not a possible improvement of $V$.
```

A^l_{Plu} The algorithm for DOMINATION (Algorithm 5) runs Algorithm 4 twice to check whether U is a possible improvement of V, and whether V is a possible improvement of U.

```
Algorithm 5: Domination
if PossibleImprovement($V$,$U$)="yes" and PossibleImprovement($U$,$V$)="no"
then
    Output that $V$ is dominated by $U$.
end
else
    Output that $V$ is not dominated by $U$.
end
```

The algorithm for DOMINATING MANIPULATION for plurality simply runs Algorithm 5 $m-1$ times. In the input we always have that $V = V_M$, and for each alternative in $\mathcal{C} \backslash \{\mathrm{Alt}(V,1)\}$, we solve an instance where that alternative is ranked first in U. If in any step V is dominated by U, then there is a dominating manipulation; otherwise V is not dominated by any other vote. The algorithms for DOMINATION and DOMINATING MANIPULATION for veto are similar.

6.4 Summary

We have shown in this chapter that for many common voting rules, restricting the manipulator's information about the other voters' votes is an effective way to make dominating manipulation computationally hard, or even impossible. Analysis of manipulation with partial information provides insight into what needs to be kept confidential in an election. For instance, in a plurality or veto election, revealing (perhaps unintentionally) part of the preferences of non-manipulators may open the door to strategic voting.

In Chapter 4, 5, and 6 we have seen some recent work and discussions on using computational complexity as a barrier against manipulation. However, a more important question that should be asked is: Why should we even try to prevent manipulation and other types of strategic behavior? In the next chapter, we will show that indeed, the strategic behavior of the voters can lead to extremely undesirable outcomes, in a type of voting games which we call *Stackelberg voting games*.

7

Stackelberg Voting Games

Using computational complexity to protect elections from manipulation, bribery, control, and other types of strategic behavior is one of the major topics of Computational Social Choice. This raises the following fundamental question: Why should we prevent voters' strategic behavior? Of course we may answer this question by arguing that people should be sincere in voting due to ethical, sociological, political, or even divine reasons. However, after all, the most important objective of voting is to select a "good" alternative, especially in multi-agent systems. Therefore, we would prefer to give an answer that is similar to the following: we want to prevent voters' strategic behavior because it might lead to undesirable outcomes.

Showing evidence for this answer in the voting setting is not as simple as it may seem to be. One approach is to consider the game where all voters vote at the same time, and study the equilibria of this simultaneous-move voting game. Unfortunately, even in a complete-information setting where all voters' preferences are common knowledge, this leads to an extremely large number of equilibria, many of them bizarre. For example, as we have seen in an example in Section 3.2, in a plurality election with the lexicographic tie-breaking mechanism, it may be the case that all voters' true preferences are Obama≻Clinton≻McCain. Nevertheless, the profile where all three voters vote for McCain≻Clinton≻Obama is a Nash equilibrium. This equilibrium is quite robust, because voting for Obama is a waste, given that nobody else is expected to vote for Obama and some votes went to Clinton. There has been some work exploring different solution concepts in simultaneous-move voting games—e.g., Farquharson (1969) and Moulin (1979)—but in some sense, the equilibrium selection issue in the above example is inherent in settings where voters vote simultaneously.

In many practical situations, the voters vote one after another, and the later voters know the votes cast by the earlier voters. For example, consider online systems that allow users to rate movies or other products. We consider the setting where the voters vote one after another in this chapter to overcome the equilibrium selection problem. We assume that voters' preferences over the alternatives are strict; we also make a complete-information assumption that the voters' preferences are common knowledge (among the voters themselves, though not necessarily to the election organizer).[1] This results in an extensive-form game of perfect information that can be solved by backward induction. In sharp contrast to the simultaneous-move setting, this results in a unique outcome (winning alternative). We refer to this game as a *Stackelberg voting game*.

Our main theoretical results will be shown in Section 7.2. As a corollary to our main theorem, which is quite technical but very general, we will show that for any voting rule r that satisfies the majority criterion (see Section 2.2 for the definition), no matter how many voters there are, there always exists a profile such that the backward-induction winner (i.e., the unique winner in all SPNE) of the Stackelberg

[1] While this is clearly a simplifying assumption, it approximates the truth in many settings, and with this assumption we do not need to specify prior distributions over preferences. Also, naturally, our negative results still apply to more general models, including models allowing for incomplete information, so long as the complete-information setting is a special case.

voting game that uses r is ranked within the bottom two positions in all voters' true preferences, with only two exceptions. This result is quite negative, because it says that if we allow voters to vote strategically, then sometimes the outcome is almost the *worst* outcome for all but two voters. Therefore, to some extent we are showing an ordinal *price-of-anarchy (PoA)* (Koutsoupias and Papadimitriou, 1999). The PoA is the ratio of the optimal social welfare over the worst social welfare in equilibrium outcomes. In fact, in the settings where social welfare is not well-defined, it is even not clear how the PoA should be defined. Fortunately, the paradoxes we will show are clearly very negative results.

Similar to the "worst case vs. typical case" debate about the results on hardness of manipulation, here again we can ask how often the paradoxes happen. To answer this question, we will pursue an empirical approach in Section 7.4. We will use the techniques developed in Section 7.3 to run simulations to compare the backward-induction winner to a benchmark outcome—namely, the alternative that would win if all voters voted truthfully. Our experimental results show that, surprisingly, more voters prefer the backward-induction outcome over the truthful outcome on average. Therefore, it seems that on average the backward-induction outcome is not too undesirable.

The idea of modeling a voting process in which voters vote one after another as an extensive-form game is not new. Sloth (1993) studied elections with two alternatives, as well as settings with more alternatives where a pairwise decision between two options is made at every stage. She relates the outcomes of this process to the multistage sophisticated outcomes of the game (McKelvey and Niemi, 1978; Moulin, 1979). In the extensive-form games studied by Dekel and Piccione (2000), multiple voters can vote simultaneously in each stage. They compare the equilibrium outcomes of these games to the outcomes of the *symmetric equilibria* of their simultaneous counterparts. Battaglini (2005) studies how these results are affected by the possibility of abstention and a small cost of voting.

Our approach is significantly different from the previous approaches in several aspects. First, the prior work focuses mostly on the case of two alternatives or, in the case of multiple alternatives, on particular voting procedures; in contrast, we consider general (anonymous) voting rules with any number of alternatives, and correspondingly derive very general paradoxes. Second, we show some paradoxes to illustrate that the strategic behavior of the voters sometimes leads to very undesirable outcomes. Third, we also study how the backward-induction outcome can be efficiently computed, and we use these algorithmic insights in simulations to evaluate the quality of the Stackelberg voting game's outcome "on average."

Desmedt and Elkind (2010) simultaneously and independently studied a similar setting in which voters vote sequentially under the plurality rule, and showed several different types of paradoxes. In their model, voters are allowed to abstain, and voting comes at a small cost. They assume random tie-breaking and therefore need to consider expected utilities, while in our model studied in this chapter, voters' preferences are ordinal.

7.1 Stackelberg Voting Game

We now consider the strategic Stackelberg voting game. We use a complete-information assumption: all the voters' preferences are common knowledge. Given this assumption, for any voting rule r, the process where voters vote in sequence can be modeled as an extensive-form game of perfect information. In Section 3.2 we gave the formal definition of simultaneous-move voting games, and mentioned that extensive-form voting games can be defined similarly. Here I will be more specific. The game has n stages. In stage j ($j \leqslant n$), voter j chooses an action from $L(\mathcal{C})$. Each leaf of the tree is associated with an outcome, which is the winner for the profile consisting of the votes that were cast to reach this leaf.

Because the voters' preferences are linear orders (which implies that there are no ties), we can solve the game by backward induction, which results in a unique outcome. We note that this requires only ordinal preferences, that is, we do not need to define utilities. The backward-induction process works as follows. First, for any subprofile of votes by the voters 1 through $n-1$ (that is, any node that is the parent of leaves), there will be a nonempty subset of alternatives that n can make win by casting some vote. She will pick her most preferred one. Now, because we can predict what voter n will do, we take voter $(n-1)$'s perspective: for any subprofile of votes by the voters 1 through $n-2$, there will be a nonempty subset of alternatives that voter $n-1$ can make win by casting some vote (taking into account how voter n will act). She will pick her most preferred one; etc. We continue this process all the way to the root of the tree; the outcome there is called the *backward-induction outcome*.

As noted above, only the ordinal preferences of the voters matter; that is, a voter's preferences correspond to a member of $L(\mathcal{C})$. While votes and preferences both lie in the same set $L(\mathcal{C})$, we must be careful to distinguish between them, because in this context, a voter will sometimes cast a vote that is different from her true preferences. Nevertheless, we can use $P \in \mathcal{F}_n$ to denote a profile of preferences, as well as a profile of votes. For a given voting rule r, let $r(P)$ be the outcome if the *votes* are P; let $SG_r(P)$ be the backward-induction outcome if the *true preferences* are P.[2]

7.2 Paradoxes

In this section, we investigate whether the strategic behavior described above will lead to undesirable outcomes. It turns out that it can. Our main theorem is a

[2] Of course, because it is a function from profiles of linear orders to alternatives, SG_r can also be interpreted as a voting rule, though there is a significant risk of confusion in doing so. We note that even if r is anonymous, SG_r (as a voting rule) is not necessarily anonymous (the order of the voters matters).

general result that applies to many common voting rules. We will show that, for such a rule, there exists a profile that has two types of paradox associated with it in the backward-induction outcome: first, the winner loses all but one of its pairwise elections; second, the winner is ranked somewhere in the bottom two positions in almost every voter's true preferences. For the second type of paradox, we will show that the number of exceptions (voters who rank the winner higher) is closely related to a parameter called the *domination index*. The domination index of a voting rule r that satisfies non-imposition is the smallest number q such that any coalition of $\lfloor n/2 \rfloor + q$ voters can make any given alternative win (no matter how the remaining voters vote) under r. We note that the domination index is always well defined for any rule that satisfies non-imposition, and is at least 1.

Definition 7.2.1. *For any voting rule r that satisfies non-imposition, and any $n \in \mathbb{N}$, we let the* domination index $DI_r(n)$ *be the smallest number q such that for any alternative c, and for any subset of $\lfloor n/2 \rfloor + q$ voters, there exists a profile P for these voters, such that for any profile P' for the remaining voters, $r(P \cup P') = c$.*

The domination index DI_r is closely related to the *anonymous veto function* $\mathrm{VF}_r : \{1, \ldots, n\} \to \{0, \ldots, m\}$ (Definition 10.4 in Moulin (1991)), defined as follows. $\mathrm{VF}_r(i)$ is the largest number $j \leqslant m - 1$ such that any coalition of i voters can veto any subset (that is, make sure that none of the alternatives in the set is the winner) of no more than j alternatives. We note that the domination index $\mathrm{DI}_r(n)$ for a voting rule r is the smallest number q such that $\mathrm{VF}_r(\lfloor n/2 \rfloor + q) = m - 1$ (that is, any coalition of size $\lfloor n/2 \rfloor + q$ can veto any set of $m - 1$ alternatives).

The next proposition gives bounds on the domination index for some common voting rules.

Proposition 7.2.2. *For any positional scoring rule r, $DI_r \leqslant \lfloor n/2 \rfloor - \lfloor n/m \rfloor$. $DI_r(n) = 1$ for any voting rule r that satisfies the majority criterion (Section 2.2), including* *any rule that satisfies the Condorcet criterion (Section 2.2), plurality, plurality with runoff, Bucklin, and STV.*

The next lemma provides a sufficient condition for an alternative not to be the backward-induction winner. It says that if there is a coalition of size $k \geqslant \lfloor n/2 \rfloor + \mathrm{DI}_r(n)$ who all prefer c to d, and another condition holds, then d cannot win.[3] For any alternative $c \in \mathcal{C}$ and any $V \in L(\mathcal{C})$, we let $\mathrm{Up}(c, V)$ denote the set of all alternatives that are ranked higher than c in V.

Lemma 7.2.3. *Let P be a profile. An alternative d is not the winner $SG_r(P)$ if there exists another alternative c and a sub-profile $P_k = (V_{i_1}, \ldots, V_{i_k})$ of P that satisfies the following conditions: 1. $k \geqslant \lfloor n/2 \rfloor + DI_r(n)$, 2. $c \succ d$ in each vote in P_k, 3. for any $1 \leqslant j_1 < j_2 \leqslant k$, $Up(c, V_{i_{j_1}}) \supseteq Up(c, V_{i_{j_2}})$.*

Proof. Let $D_k = \{i_1, \ldots, i_k\}$. Since $k \geqslant \lfloor n/2 \rfloor + \mathrm{DI}_r(n)$, this coalition of voters can guarantee that any given alternative be the winner under r, if they work together. Let $P_k^* = (V_{i_1}^*, \ldots, V_{i_k}^*)$ be a profile that can guarantee that c be the winner under r. That is, for any profile P' for the other voters ($\{1, \ldots, n\} \backslash D_k$), we have $r(P_k^*, P') = c$. For any $j \leqslant k$, we let $D'_{i_j} = \{1, \ldots, i_j\} \backslash D_k$—that is, the first i_j voters, except those in the coalition D_k. For any $j \leqslant k$, we let $P_j^* = (V_{i_1}^*, \ldots, V_{i_j}^*)$. That is, P_j^* consists of the first j votes in P_k^*. For any $i \leqslant n-1$ and any pair of profiles P_1 (consisting of i votes) and P_2 (consisting of $n-i$ votes), we let $SG_r(P_2 : P_1)$ denote the backward-induction winner of the subgame of the Stackelberg voting game in which voters 1 through i have already cast their votes P_1, and the true preferences of voters $i+1$ through n are as in P_2. We prove the following claim by induction.

[3] This may seem trivial because the coalition can guarantee that c wins if they work together. However, we have to keep in mind that the members of the coalition each pursue their own interest. For example, it may be the case that whenever the second-to-last voter in the coalition votes in a way that enables the last voter in the coalition to make c the winner, it also enables this last voter to make e the winner, which this last voter prefers—but the second-to-last voter actually prefers d to e, and therefore votes to make d win instead. We need the extra condition to rule out such examples.

Claim 7.2.1. *For any $j \leqslant k$ and any profile P'_{i_j} for the voters in D'_{i_j},*

$$SG_r((V_{i_j}, V_{i_j+1}, \ldots, V_n) : P'_{i_j}, P^*_{j-1}) \succeq_{V_{i_j}} c$$

Claim 7.2.1 states that for any $j \leqslant k$, if voters $i_1, \ldots, i_{j-1}$ have already voted as in P^*_j, and voter i_j will vote next, then the backward-induction outcome of the corresponding subgame must be (weakly) preferred to c by voter i_j.

Proof of Claim 7.2.1: The proof is by (reverse) induction on j. First, we consider the base case where $j = k$. If voter i_k casts $V^*_{i_k}$, then the winner is c, because the subprofile P^*_k will guarantee that c wins. Voter i_k will only vote differently if it results in at least as good an outcome for her as c. Therefore, the claim holds for $j = k$.

Now, suppose that for some j', the claim holds for $j' \leqslant j \leqslant k$. We will now show that it also holds for $j = j' - 1$. Let c' be the backward-induction outcome when voter $i_{j'-1}$ submits $V^*_{i_{j'-1}}$. By the induction hypothesis, we have that $c' \succeq_{V_{i_{j'}}} c$. That is, voter $i_{j'}$ (weakly) prefers c' to c. We recall that $\text{Up}(c, V_{i_{j'-1}}) \supseteq \text{Up}(c, V_{i_{j'}})$, which means that c' is also (weakly) preferred to c by voter $i_{j'-1}$. This means that voter $i_{j'-1}$ can guarantee that the outcome be at least as good as c for her. She will only vote differently from $V^*_{i_{j'-1}}$ if it results in at least as good an outcome for her as c' (which is at least as good as c already). Therefore, the claim also holds for $j' - 1$, and Claim 7.2.1 follows by induction. □

Letting $j = 1$ in Claim 7.2.1, we have that $SG_r(P) \succeq_{V_{i_1}} c$. Therefore, $d \neq SG_r(P)$ (because $c \succ_{V_{i_1}} d$). This completes the proof of Lemma 7.2.3. □

We are now ready to present our main theorem. We note that this theorem does not depend on the tie-breaking mechanism used in the rule.

Theorem 7.2.4. *For any voting rule r that satisfies non-imposition, and any $n \in \mathbb{N}$, there exists a profile P such that $SG_r(P)$ is ranked somewhere in the bottom two* *positions in $n - 2DI_r(n)$ of the votes, and, if $DI_r(n) < n/4$, then $SG_r(P)$ loses to all but one alternative in pairwise elections.*

Proof. The proof is constructive. Let $P = (V_1, \ldots, V_n)$ be the profile (the voters' true preferences) defined as follows.

$$V_1 = \cdots = V_{\lfloor n/2 \rfloor - \mathrm{DI}_r(n)} = [c_3 \succ \ldots \succ c_m \succ c_1 \succ c_2]$$
$$V_{\lfloor n/2 \rfloor - \mathrm{DI}_r(n)+1} = \cdots = V_{\lfloor n/2 \rfloor + \mathrm{DI}_r(n)} = [c_1 \succ c_2 \succ c_3 \succ \cdots \succ c_m]$$
$$V_{\lfloor n/2 \rfloor + \mathrm{DI}_r(n)+1} = \cdots = V_n = [c_2 \succ c_3 \succ \cdots \succ c_m \succ c_1]$$

We now use Lemma 7.2.3 to prove that $SG_r(P) = c_1$. First, we let $k = \lfloor n/2 \rfloor + \mathrm{DI}_r(n)$, and let P_k be the first k votes. It follows from Lemma 7.2.3 (letting $c = c_1$ and $d = c_2$) that $c_2 \neq SG_r(P)$. Next, for any $c' \in \mathcal{C} \backslash \{c_1, c_2\}$, we let $k = \lceil n/2 \rceil + \mathrm{DI}_r(n)$ and let P_k be the last k votes, that is, $P_k = (V_{\lfloor n/2 \rfloor - \mathrm{DI}_r(n)+1}, \ldots, V_n)$. By Lemma 7.2.3 (letting $c = c_2$ and $d = c'$), we have that $c' \neq SG_r(P)$. It follows that $SG_r(P) = c$.

In P, c_1 is ranked somewhere in the bottom two positions in $n - 2\mathrm{DI}_r(n)$ votes (the first $\lfloor n/2 \rfloor - \mathrm{DI}_r(n)$ votes and the last $\lceil n/2 \rceil - \mathrm{DI}_r(n)$ votes). If $\mathrm{DI}_r(n) < n/4$, then $2\mathrm{DI}_r(n) < n/2$, which means that c_1 will lose to any other alternative (except c_2) in pairwise elections. □

Combining Proposition 7.2.2 and Theorem 7.2.4, we obtain the following corollary for common voting rules.

Corollary 7.2.5. *Let r be any rule that satisfies non-imposition and majority criterion, and let $n \geqslant 5$. There exists a profile P such that $SG_r(P)$ is ranked somewhere in the bottom two positions in $n - 2$ votes; moreover, $SG_r(P)$ loses to all but one alternative in pairwise elections. (This holds regardless of how ties are broken.)*

While this is a strong paradox already, it is sometimes possible to obtain even stronger paradoxes if we restrict attention to individual rules. We have illustrated this on some voting rules including plurality, which can be found in Xia and Conitzer (2010b).

7.3 Computing the Backward-Induction Outcome

We have shown in the last section that the backward-induction solution to the Stackelberg voting game is socially undesirable for some profiles. We may ask ourselves whether such profiles are common, or just isolated instances that are not very likely to happen in practice. For this purpose, we would like to compare the backward-induction winner to the truthful winner by running simulations. For this purpose, we should be able to compute the backward-induction winners reasonably fast. However, even if the outcome of the rule r is easy to compute, it does not follow that the outcome of SG_r is easy to compute. The straightforward backward-induction process described above is very inefficient, because the game tree has $(m!)^n$ leaves.

In this section, we first propose a general dynamic-programming algorithm to compute $SG_r(P)$, for any anonymous voting rule r. Then, we show how to use *compilation functions* (Chevaleyre et al., 2009) (see also Section 1.6) to further reduce the time/space-complexity of the dynamic-programming algorithm. These techniques are crucial for obtaining our later experimental results.

The dynamic-programming algorithm still solves the game tree in a bottom-up fashion, but does not need to consider all the different profiles separately. Because r is anonymous, at any stage j of the game, the state (the profile of votes 1 through $j-1$) can be summarized by a vector composed of $m!$ natural numbers, one for each linear order: each number in the vector represents the number of times that the corresponding linear order appears in the $(j-1)$-profile. Formally, for any $j \leqslant n$, we let the set of these vectors (states) be $S_j = \{(s_1, \ldots, s_{m!}) \in {\mathbb{N}_{\geqslant 0}}^{m!} : \sum_{i=1}^{m!} s_i = j-1\}$. For any anonymous voting rule r and any $\vec{s} \in S_{n+1}$, let $r(\vec{s})$ be the winner for any profile that corresponds to $\vec{s}$ (because r is anonymous, the winner only depends on the vector $\vec{s}$). More generally, for arbitrary S_j, the algorithm computes a labeling function g that maps each state $\vec{s} \in S_j$ to the alternative representing the backward-induction outcome of the subgame whose root corresponds to $\vec{s}$.

```
Algorithm 7.3.1.
Input. $P = (V_1, \dots, V_n)$ and an anonymous voting rule $r$.
Output. $SG_r(P)$.
1. For $j$ from $n+1$ to 1, do Step 2.
2. For any state $\vec{s} \in S_j$, do
    2.1 If $j = n+1$, then let $g(\vec{s}) = r(\vec{s})$.
    2.2 If $j < n+1$, then let $\vec{e}^* \in \arg\min_{\vec{e} \in E} \text{rank}(V_j, g(\vec{s}+\vec{e}))$, where $E$ consists of all vectors that are composed of $m!-1$ zeroes and only one 1, and $\text{rank}(V_j, g(\vec{s}+\vec{e}))$ is the position of $g(\vec{s}+\vec{e})$ in $V_j$. (Thus, $e^*$ corresponds to an optimal vote for $j$.) Then, let $g(\vec{s}) = g(\vec{s}+\vec{e}^*)$.
3. Output $g((0, \dots, 0))$.
```

Analysis. For any $j \leqslant n$, $|S_j| = \binom{j+m!-2}{m!-1}$ (this is a basic combinatorial result, see e.g. Bender and Williamson (2006)). To analyze the runtime of the algorithm, we note that the total number of states considered is $\sum_{j=1}^{n+1} \binom{j+m!-2}{m!-1}$, which is $O((n+1)^{m!+1})$; in each state, we need to consider $m!$ vectors $\vec{e}$, resulting in a total bound of $O(m!(n+1)^{m!+1})$. To analyze the space requirements of the algorithm, we note that we only need to keep the last stage $j+1$ and the current stage j in memory, so that the maximum number of states in memory is $\binom{n+m!-1}{m!-1} + \binom{n+m!-2}{m!-1}$, which is $O((n+1)^{m!})$. Therefore, when m is bounded above by a constant, Algorithm 7.3.1 runs in polynomial time (using polynomial space).

However, when there is no upper bound on m, Algorithm 7.3.1 runs in exponential time and uses exponential space. We conjecture that for many common voting rules (e.g., plurality), computing SG_r is PSPACE-hard, but we have not managed to obtain any such result yet.[4]

Compilation. In the step corresponding to stage j in Algorithm 7.3.1, a very large set S_j is used to keep track of all possible $m!$-dimensional vectors whose entries sum to exactly $j-1$, representing the possible states. While it may be necessary to have this many states for anonymous rules in general, it turns out that for specific rules like plurality or veto, we need far fewer states to represent the profiles, because many of the states in Algorithm 7.3.1 will be equivalent *for the specific rule.* For example, if we have so far received only a single vote $a \succ b \succ c$, this in general is not equivalent to having received only a single vote $a \succ c \succ b$. However, if the rule is plurality, these states are equivalent.

Pursuing this idea, for any anonymous voting rule r, we can ask the following questions. (1) *What is the smallest set of states needed for stage j?* (2) *How can we incorporate smaller sets of states into Algorithm 7.3.1?*

The answer to question (1) corresponds to the *compilation complexity* of r, a concept introduced by Chevaleyre et al. (2009). For any $k, u \in \mathbb{N}$ with $k + u = n$, the compilation complexity $C_{m,k,u}(r)$ is defined to be the smallest number of bits needed to represent all "effectively different" k-profiles, when there are u remaining votes and the winner is chosen by using r. (Two k-profiles are "effectively the same" if, for any profile of u votes that we may add to them, they result in the same outcome.) It follows that, if we tailor Algorithm 7.3.1 to a specific rule r, the size of the smallest possible set of states for stage j is between $2^{C_{m,j-1,n-j+1}(r)-1}$ and $2^{C_{m,j-1,n-j+1}(r)}$. Chevaleyre et al. (2009) also studied the compilation complexity for some common voting rules.

Now we turn to address question (2). Suppose that we have already determined

[4] We have obtained a PSPACE-hardness result for a not-so-common rule with a different type of voter preferences, which thus falls somewhat outside of the setting described so far. We omit it due to the space constraint.

that we can use a smaller set of states. In order to modify the dynamic-programming algorithm to use this smaller set of states, for step (2.2) we must have a function that takes a state in S_j and a vote V as inputs, and outputs a state in S_{j+1}; moreover, this function must be easy to compute. Fortunately, the *compilation functions* designed for some common voting rules in Chevaleyre et al. (2009) and Xia and Conitzer (2010a), which map each profile to a string (state), can serve as such functions. For example, the compilation function for plurality simply counts how often each alternative has been ranked first, and this is easy to update. More generally, we can modify Algorithm 7.3.1 for any specific rule r as follows. Let $f^r_{m,k,u}$ be a compilation function for r. For any $j \leqslant n$, we let $S_j = f^r_{m,k,u}(\mathcal{F}_{j-1})$, that is, the set of all "compressed" $(j-1)$-profiles. Then, in step (2.2), for each given state $\vec{s} \in S_j$ and each[5] given vote $V \in L(\mathcal{C})$, the next state (which lies in S_{j+1}) is computed by applying the compilation function $f^r_{m,k,u}$ to the combination of $\vec{s}$ and V. Among these resulting states, we again find voter j's most-preferred outcome.

Illustration. Let us illustrate how the use of compilation functions helps reduce the time and space requirements of Algorithm 7.3.1 for the *nomination* rule, which selects the alternative that is ranked in the first position in at least one vote, where ties are broken in the order $c_1 \succ \cdots \succ c_m$. In this case, for any $j \leqslant n$, let $S_j = \mathcal{C}$, and let f^{Nom} be the following compilation function. For any profile P, let $f^{\text{Nom}}(P)$ be the first alternative (according to the order $c_1 > \cdots > c_m$) that has been nominated (is ranked first in some vote in P). For any profile P and any vote V, $f^{\text{Nom}}(P \cup \{V\})$ can be easily computed from $f^{\text{Nom}}(P)$ and V, by determining which of $f^{\text{Nom}}(P)$ and the alternative ranked in the top position in V is earlier in the order. (As in the case of plurality, we do not need to consider every vote V: we only need to consider which alternative is ranked first.) Because $|S_j| = m$ for all j in this case, it follows

[5] For some rules, we do not need to consider every vote: for example, under plurality, we do not need to consider both $a \succ b \succ c$ and $a \succ c \succ b$.

that Algorithm 7.3.1 (using f^{Nom}) runs in polynomial time for the nomination rule.

Proposition 7.3.1. *SG_{Nom} can be computed in polynomial time (and space) by Algorithm 7.3.1 (using f^{Nom}).*

For other, more common voting rules, the runtime of the dynamic-programming algorithm is also significantly reduced by using compilation functions, though it remains exponential. For example, for plurality and veto, the time/space complexity of our approach is $O(n^m)$, which allows us to conduct the simulation experiments in the next section much more efficiently.

7.4 Experimental Results

Using the algorithmic techniques developed in the last section, we are able to run simulations to compare the backward-induction winner $SG_r(P)$ to a benchmark outcome—namely, the alternative $r(P)$ that would win under r if all voters vote truthfully. This may seem like a difficult benchmark to achieve, because often strategic behavior comes at a cost (cf. price of anarchy, first-best vs. second-best in mechanism design, etc.) Nevertheless, in the experiments that we describe in this section, it turns out that in randomly chosen profiles, in fact, slightly more voters prefer the backward-induction outcome $SG_r(P)$ to the truthful outcome $r(P)$ than vice versa!

The setup of our experiment is as follows. We study the plurality and veto rules (these are the easiest to scale to large numbers of voters, because they have low compilation complexity).[6]

For any m, n, and $r \in \{\text{Plurality}, \text{Veto}\}$, our experiment has 25,000 iterations. In each iteration, we perform the following three steps.

[6] We also investigated other rules. It appears that they may lead to similar results, though it is difficult to say this with high confidence because we can only solve for the backward-induction outcome for small numbers of voters.

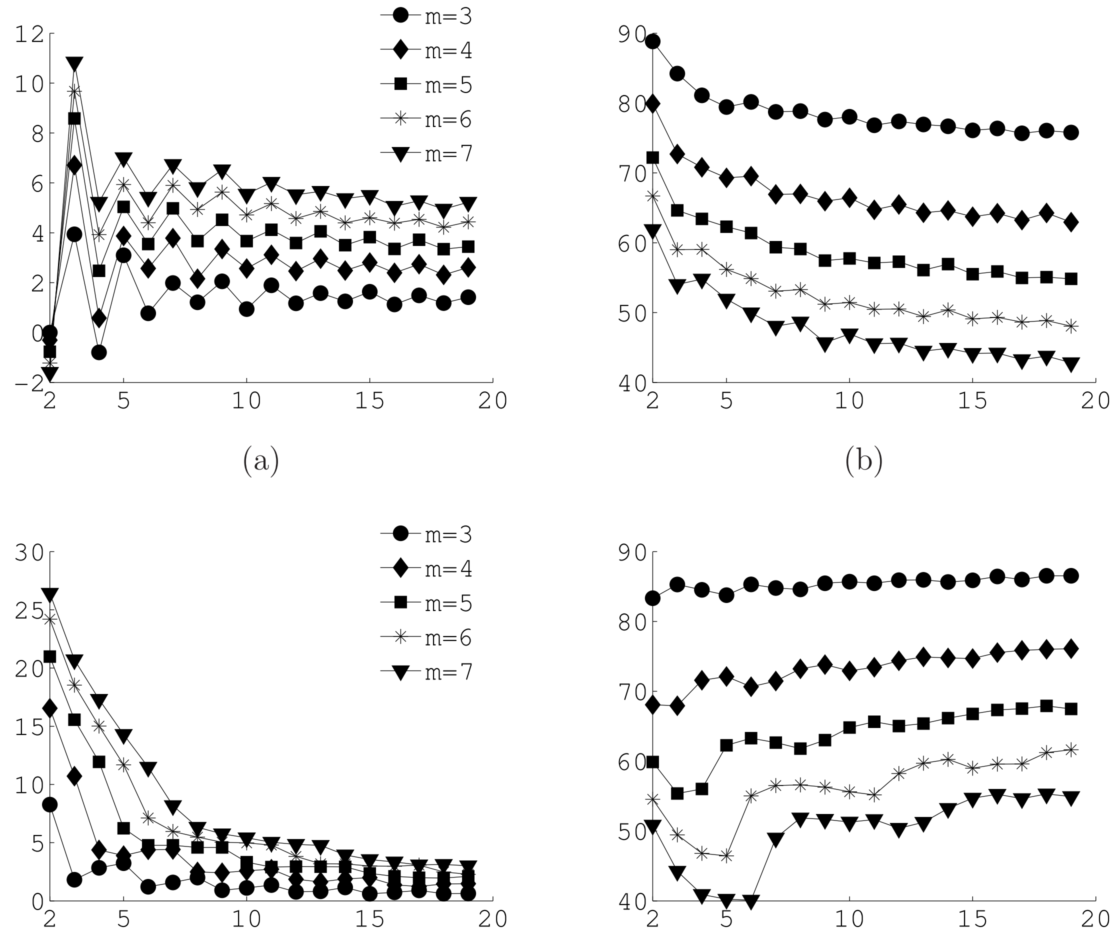
1. In iteration j, an n-profile P_j is chosen uniformly at random from $\mathcal{F}_n$.

2. We calculate $SG_r(P_j)$ using Algorithm 7.3.1 (with a compilation function to reduce time/space-complexity), and we calculate $r(P_j)$.

3. We then count the number of voters in this profile P that prefer $SG_r(P)$ to $r(P)$ (according to their true preferences in P), denoted by n_1, and vice versa, denoted by n_2. If $SG_r(P) = r(P)$, then $n_1 = n_2 = 0$.

For each m, n, r, we calculate the total percentage (across all 25,000 iterations) of voters that prefer the backward-induction winner for their profile to the winner under truthful voting for their profile, that is, $p_1 = \sum_{j=1}^{25000} n_1^j/(25000n)$. We also compute $p_2 = \sum_{j=1}^{25000} n_2^j/(25000n)$. We note that it is not necessarily the case that $p_1 + p_2 = 1$, because if $SG_r(P) = r(P)$, then $n_1 = n_2 = 0$. Let $p_3 = 1 - p_1 - p_2$ be the percentage of profiles for which the backward-induction (SG_r) winner coincides with the truthful (r) winner. We are primarily interested in $p_1 - p_2$.

The results are summarized in Figure 7.1.



(a)

(b)

(c)

(d)

Figure 7.1: Simulation results for plurality and veto

In Figure 7.1, the x-axis gives the number of voters (n); the y-axis gives the percentage of voters. In each case we consider various numbers of alternatives (m). (a) The percentage of voters who prefer the SG_r winner to the r winner minus the other way around, under plurality. (b) The percentage of profiles for which the SG_r winner and the r winner are the same, under plurality. (c) The percentage of voters who prefer the SG_r winner to the r winner minus the other way around, under veto. (d) The percentage of profiles for which the SG_r winner and the truthful r winner are the same, under veto. Please note the different scales on the y-axis for (a) and (c).

First, from (a) and (c) it can be observed that for plurality and veto, perhaps surprisingly, on average, more voters prefer the backward-induction winner to the winner under truthful voting than vice versa. Generally, the difference becomes smaller when n increases; the difference is larger when m is larger; and the percentage seems to converge to some limit as $n \rightarrow \infty$. Second, from (b) and (d) it can be observed that the percentage of profiles for which the two winners coincide is smaller for larger values of m; the percentage is decreasing in the number of voters n for plurality, but increasing for veto.

7.5 Summary

In this chapter we studied the voting game where voters cast votes one after another and the later voters can observe all previous voters' votes. We proved some paradoxes, which state that sometimes the strategic behavior of the voters can be harmful in Stackelberg voting games. To some extent, these paradoxes justify the line of research in which people seek to use computational complexity to prevent strategic behavior. We also developed algorithmic techniques to run simulations. Our simulation results show that, surprisingly, the strategic behavior of the voters does not seem as harmful as we might have expected.

8

Introduction to Combinatorial Voting

We recall from Section 2.3 that one major direction in Computational Social Choice is to investigate the computational complexity of winner determination for some common voting rules, and then design heuristic, fixed-parameter tractable, or approximation algorithms for voting rules for which the winner is hard to compute. In those situations, the computational complexity mainly comes from the choice of voting rule.

However, in many real life group decision making problems, the computational complexity comes from the extremely large number of alternatives. In such cases it may take an unbearably long time to compute the winner even for simple voting methods such as Borda. Perhaps the most natural and prominent voting setting in real-life with an extremely large number of alternatives is *combinatorial voting*, a.k.a. *voting in multi-issue domains*. In combinatorial voting, the set of alternatives has a combinatorial (namely, multi-issue) structure. That is, there are multiple *issues* (or *attributes*, or *characteristics*) and each alternative can be uniquely characterized by a vector of the values these issues take. For example, consider a situation where the inhabitants of a county vote to determine a government plan. The plan is composed of multiple sub-plans for several interrelated issues, such as transportation, environment, and health (Brams et al., 1998). Another example is *voting by committees*, in which the voters select a subset of objects (Barbera et al., 1991), where each object can be seen as a binary issue. In such situations, a voters' preferences over one issue may well depend on the values of other issues. For example, a voter may prefer creating a natural reserve if a highway is built, but if the highway is not built, she may prefer not creating a reserve.

In the remainder of this dissertation (Chapter 8–12), we will focus on the design and analysis of voting rules when the set of alternatives has a multi-issue structure. In this chapter, we give the formal definitions and notation that will be used throughout these chapters.

Definition 8.0.1 (Combinatorial voting). *Let $\mathcal{I} = \{X_1, \ldots, X_p\}$ denote a set of $p \geqslant 2$ issues, where for each $i \leqslant p$, X_i takes a value in a* local domain D_i*, where $|D_i| \geqslant 2$.*[1] Combinatorial voting *refers to the voting setting where the set of alternatives is $\mathcal{X} = D_1 \times \cdots \times D_p$. $\mathcal{X}$ is called a* multi-issue domain *or* combinatorial domain*.*

Example 8.0.2. *A group of people must make a joint decision on the menu for dinner (the caterer can only serve a single menu to everyone). The menu is composed of two issues: the main course (**M**) and the wine (**W**). There are three choices for the main course: beef (b), fish (f), or salad (s). The wine can be either red wine (r), white wine (w), or pink wine (p). The set of alternatives is a multi-issue domain: $\mathcal{X} = \{b, f, s\} \times \{r, w, p\}$.*

We call that the set of alternatives $\mathcal{C}$ studied in previous chapters constitutes an *unstructured domain*, because it does not need to have a multi-issue structure. In the above definition, we use $\mathcal{X}$ (instead of $\mathcal{C}$) to emphasize that the set of alternatives

[1] This is the standard assumption for studying voting in multi-issue domains, because otherwise either the domain can be simplified (by removing issues that can only take one value), or it has no multi-issue structure (when there is only one issue).

has a multi-issue structure. Following convention, for any $i \leqslant p$ we let $D_{-i} = D_1 \times \cdots \times D_{i-1} \times D_{i+1} \times \cdots \times D_p$.

A special case of multi-issue domains consists of the domains where all variables are binary, that is, for all $i \leqslant p$, $D_i = \{0_i, 1_i\}$. Such multi-issue domains are called *multi-binary-issue domains*. Even in multi-binary-issue domains, the number of alternatives is 2^p, which is already exponentially large. Moreover, we recall that the voting setting we defined in Chapter 2 requires a voter to submit a linear order over the set of alternatives. This requirement causes the major problem in combinatorial voting, which is that it is infeasible for a voter to give a full ranking over an exponentially large number of alternatives. Therefore, in combinatorial voting, the voters need to use another *voting language* to represent their preferences, and then we can design novel voting rules to aggregate voters' preferences represented by such a voting language.

An obvious solution is the following: we can simply ask voters to report only a (small) part of their preference relation and apply a voting rule that needs this information only. For example, we can ask the voters to report their most-preferred alternatives, and then apply the plurality rule. The voting language used in this case is the set of all alternatives instead of the set of all linear orders over the alternatives. One problem with this type of solution is the following: as soon as the number of alternative is large ($2^p \gg n$), the voters are likely to be unhappy about only expressing a small portion of their preferences. Moreover, the result of voting is likely to be completely insignificant or even catastrophic. For instance, with 5 voters and 6 binary issues, it is very likely that all 5 voters vote for different alternatives (since there are $2^6 = 64$ alternatives), and the winner under the plurality rule might be disliked by all but one voter. In fact, this phenomenon is a type of *multiple-election paradox*, which we will discuss in more detail in the next section.

Even though the above (plurality) solution itself does not work very well, it reveals the following two important high-level criteria for voting rules in multi-issue domains.

The first criterion: The quality of the voting language, which includes *compactness* and *expressiveness*.

The second criterion: The quality of the voting rule after the votes have been collected. Here the quality is measured by computational efficiency, satisfiability of axiomatic properties (see Section 2.2), resistance to multiple-election paradoxes, etc.

The compactness of a voting language can be measured by the number of bits that is used to represent a voter's preferences. For example, $\Theta(p \cdot 2^p)$ bits are necessary and sufficient to represent the voting language that consists of all linear orders over $\mathcal{X}$, because $\log((2^p)!)$ is $\Theta(p \cdot 2^p)$. Measuring the expressiveness of a voting language is more complicated. We consider the following two dimensions.

The first dimension of expressiveness: the *general usability* of the language. That is, the percentage of voters who are comfortable using this language to express their preferences. For example, if we only ask the voters to report their top-ranked alternative, no voter will feel ill at ease to do so. However, as we will see in the next section, voters are not always comfortable expressing their preferences in issue-by-issue voting and sequential voting.

The second dimension of expressiveness: the *informativeness* of the language. That is, how much of the voters' preferences are expressed by the language. For example, the top-ranked alternative only represents a tiny portion of the voter's preferences. The languages used in issue-by-issue voting rules and sequential voting rules both allow voters to express much more of their preferences.

8.1 Multiple-Election Paradoxes

Combinatorial voting has been extensively studied by economists. Most of previous work has focused on letting voters vote on the issues separately, in the following way. For each issue (simultaneously, not sequentially), each voter reports her preferences for that issue, and then, a *local rule* is used to select the winning value that the issue will take. This voting process is called *issue-by-issue* or *seat-by-seat* voting.[2] Recently, Ahn and Oliveros (2011) studied a Bayesian game of combinatorial voting, and showed the existence equilibrium under some conditions. We will not discuss the Bayesian setting in this dissertation.

Issue-by-issue voting has some drawbacks. First, a voter may feel uncomfortable expressing her preferences over one issue independently of the values that the other issues take. This means that, even though the voting language used in issue-by-issue voting can express more of a voter's preferences than the voting language that is used in plurality, it lack usability. That is, only voters whose preferences are *separable* (that is, for any issue i, regardless of the values for the other issues, the voter's preferences over issue i are always the same) are comfortable expressing their preferences in issue-by-issue voting (Kadane, 1972; Schwartz, 1977). Second, *multiple-election paradoxes* arise in issue-by-issue voting (Brams et al., 1998; Scarsini, 1998; Lacy and Niou, 2000), which we will discuss below in more detail.

Brams et al. (1998) showed that for multi-binary-issue domains, there exists a profile where the winner under issue-by-issue voting (where all local voting rules are majority rules) receives zero votes (that is, it is never ranked in the top position by any voter). Scarsini (1998) showed an even stronger paradox: there exists a profile

[2] The names "issue-by-issue" and "seat-by-seat" are a little bit misleading. It may sound like there is an ordering over issues, according to which the voters vote over issues sequentially. Even though the election can be organized in this sequential way, effectively these issue-wise elections are conducted in parallel in issue-by-issue voting, because the voters do not learn the outcomes of other issues before deciding on an issue.

where any alternative that is “close” to the winner in terms of Hamming distance under issue-by-issue voting receives zero votes. These paradoxes exist even when the voters’ preferences are separable.

We are more interested in the paradoxes demonstrated by Lacy and Niou (2000) for issue-by-issue voting when the voters’ preferences are non-separable. Of course in such cases the voters may feel ill at ease reporting their preferences over a single issue without knowing the values of the other issues. In Lacy and Niou (2000), it is assumed that voters vote according to their top-ranked alternative. That is, when a voter is asked to report her preferences over issue X_i, she will report the value of the X_i component in her top-ranked alternative. This behavior in some sense corresponds to very optimistic voters, and Lacy and Niou argued that when a voter does not know the votes of the other voters, she is likely to vote in this way. They illustrated the paradoxes in the following example.

Example 8.1.1. *Suppose there are three voters and the multi-issue domain is composed of three binary issues. The top-ranked alternatives of the three voters are* 110*,* 101*, and* 011*, respectively; and all voters rank* 111 *in their bottom positions. Now, by voting over each issue separately in parallel using the majority rule, the winner is* 111*, which is the least-preferred alternative for all voters.*

The above example illustrates the following three types of multiple-election paradoxes for issue-by-issue voting:

First type of paradox: the winner is a Condorcet loser (who loses to all the other alternatives in their pairwise elections).

Second type of paradox: the winner is Pareto-dominated by another alternative (that is, that alternative is preferred to the winner by all voters).

Third type of paradox: the winner is ranked in a very low position in all voters' true preferences.

8.2 CP-nets

We have seen so far that none of the approaches mentioned above works well. One common deficiency of them is that the voting languages are not expressive enough. We have seen that the voting language used by plurality has a high usability (meaning that all voters are comfortable using it), but it lacks informativeness (meaning that it only represents a tiny portion of the voters' preferences). The language used by issue-by-issue voting is much less usable, because only voters whose preferences are separable are comfortable with reporting their preferences in issue-by-issue voting, and only a tiny fraction of the linear orders are separable (Hodge, 2006). But in general it is much more informative when the voters' preferences are separable. However, none of these languages model the preferential dependence among the issues.

Fortunately, a new language for preference representation in multi-issue domains, called *conditional preference networks, or CP-nets*, that captures the dependence of voters' preferences among individual issues, was recently proposed in Artificial Intelligence (Boutilier et al., 2004). Next, we first give the formal definition of CP-nets, then discuss how to use them as the voting language for sequential voting.

The definition of a CP-net is similar to that of a Bayesian network (Pearl, 1988). We first give the formal definitions, and the present an example.

Definition 8.2.1. *A CP-net $\mathcal{N}$ over $\mathcal{X}$ consists of two parts:*

(a) A directed graph $G = (\mathcal{I}, E)$.

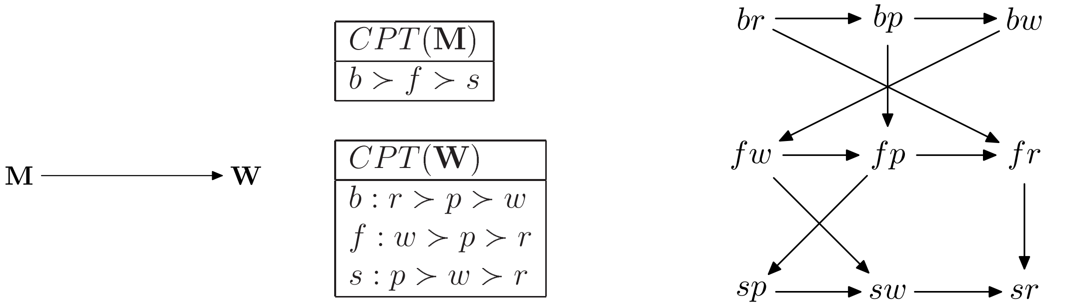
(b) A set of conditional linear preferences $\succ^i_{\vec{d}}$ over D_i, for each setting $\vec{d}$ of the parents of X_i in G. Let $CPT(X_i)$ be the set of the conditional preferences of a voter on D_i; this is called a conditional preference table (CPT).

A CP-net $\mathcal{N}$ captures dependencies across issues in the following sense. $\mathcal{N}$ induces a partial preorder $\succ_{\mathcal{N}}$ over the alternatives $\mathcal{X}$, representing the voter's preferences, as follows: for any $a_i, b_i \in D_i$, any setting $\vec{d}$ of the set of parents of X_i (denoted by $Par_G(X_i)$), and any setting $\vec{z}$ of $\mathcal{I} \backslash (Par_G(X_i) \cup \{X_i\})$, $(a_i, \vec{d}, \vec{z}) \succ_{\mathcal{N}} (b_i, \vec{d}, \vec{z})$ if and only if $a_i \succ^i_{\vec{d}} b_i$. In words, the preferences over issue X_i only depend on the setting of the parents of X_i (but not on any other issues). For any $1 \leqslant i \leqslant p$, CPT$(X_i)$ specifies conditional preferences over X_i. Now, if we obtain an alternative $\vec{d}'$ from $\vec{d}$ by only changing the value of the ith issue of $\vec{d}$, we can look up CPT(X_i) to conclude whether the voter prefers $\vec{d}'$ to $\vec{d}$, or vice versa. In general, however, from the CP-net, we will not always be able to conclude which of two alternatives a voter prefers, if the alternatives differ on two or more issues. This is why $\mathcal{N}$ usually induces a partial preorder rather than a linear order.

When the graph of $\mathcal{N}$ is acyclic, $\succ_{\mathcal{N}}$ is transitive and asymmetric, that is, a strict partial order (Boutilier et al., 2004). Let $\mathcal{O} = X_1 > \cdots > X_p$. We say that a CP-net $\mathcal{N}$ is *compatible* with (or, *follows*) $\mathcal{O}$, if the following is true: if X_i is a parent of X_j in the graph, then this implies that $i < j$. That is, preferences over any issue only depend on the values of earlier issues in $\mathcal{O}$. A CP-net is *separable* if there are no edges in its graph, which means that there are no preferential dependencies among issues.

Example 8.2.2. *Let $\mathcal{X}$ be the multi-issue domain defined in Example 8.0.2. We define a CP-net $\mathcal{N}$ as follows:* $\mathbf{M}$ *(the main course) is the parent of* $\mathbf{W}$ *(the wine), and the CPTs consist of the following conditional preferences: $CPT(\mathbf{M}) = \{b \succ f \succ s\}, CPT(\mathbf{W}) = \{b : r \succ p \succ w, f : w \succ p \succ r, s : p \succ w \succ r\}$, where $b : r \succ p \succ w$ is interpreted as follows: "when* $\mathbf{M}$ *is b, then, r is the most preferred value for* $\mathbf{W}$*, p is the second most preferred value, and w is the least preferred value." $\mathcal{N}$ and its induced partial order $\succ_{\mathcal{N}}$ are illustrated in Figure 8.1. $\mathcal{N}$ is compatible with* $\mathbf{M} > \mathbf{W}$.

$\mathcal{N}$ is not separable.



(a) A CP-net $\mathcal{N}$.

(b) The partial order induced by $\mathcal{N}$.

FIGURE 8.1: A CP-net $\mathcal{N}$ and its induced partial order.

When all issues are binary, a CP-net $\mathcal{N}$ can be visualized as a hypercube with directed edges in p-dimensional space (Domshlak and Brafman, 2002), in the following way. Each vertex is an alternative, each dimension corresponds to an issue, and any two adjacent vertices differ in only one component (issue). That is, for any $i \leqslant p$ and any $\vec{d}_{-i} \in D_{-i}$, there is a directed edge connecting $(0_i, \vec{d}_{-i})$ and $(1_i, \vec{d}_{-i})$, and the direction of the edge is from $(0_i, \vec{d}_{-i})$ to $(1_i, \vec{d}_{-i})$ if and only if $(0_i, \vec{d}_{-i}) \succ_{\mathcal{N}} (1_i, \vec{d}_{-i})$.

Example 8.2.3. *Let $p = 3$ and let $\mathcal{N}$ be a CP-net defined as follows: the directed graph of $\mathcal{N}$ has an edge from X_1 to X_2 and an edge from X_2 to X_3; the CPTs are $CPT(X_1) = \{0_1 \succ 1_1\}$, $CPT(X_2) = \{0_1 : 0_2 \succ 1_2, 1_1 : 1_2 \succ 0_2\}$, $CPT(X_3) = \{0_2 : 0_3 \succ 1_3, 1_2 : 1_3 \succ 0_3\}$. $\mathcal{N}$ is illustrated as a hypercube in Figure 8.2 (for simplicity, in the figure, a vertex abc represents the alternative $a_1 b_2 c_3$, for example, the vertex* 000 *represents the alternative* $0_1 0_2 0_3$*).*

A linear order V over $\mathcal{X}$ *extends* a CP-net $\mathcal{N}$, denoted by $V \sim \mathcal{N}$, if it extends the partial order that $\mathcal{N}$ induces. (This is merely saying that V is consistent with the preferences implied by the CP-net $\mathcal{N}$.) V is *separable* if it extends a separable

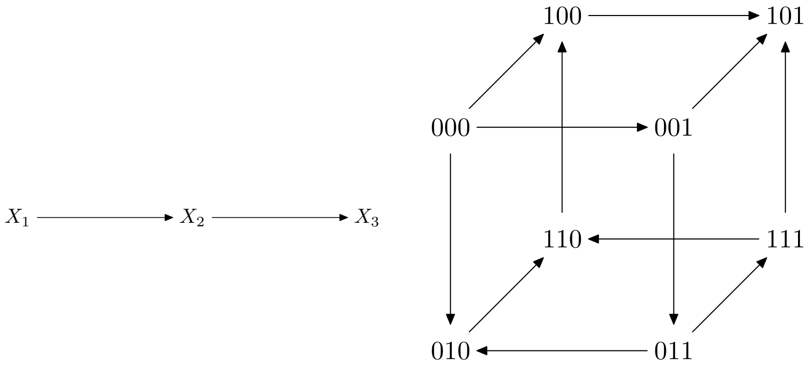


Figure 8.2: The hypercube representation of the CP-net in Example 8.2.3.

CP-net. Given an ordering $\mathcal{O}$ over issues, V is *$\mathcal{O}$-legal* if it extends a CP-net that is compatible with $\mathcal{O}$. The set of all $\mathcal{O}$-legal linear orders is denoted by $Legal(\mathcal{O})$.

To present our results, we will frequently use notations that represent the projection of a vote/CP-net/profile to an issue X_i (that is, the voter's local preferences over X_i), given the setting of all parents of X_i. These notations are defined as follows. For any issue X_i, any setting $\vec{d}$ of $Par_G(X_i)$, and any linear order V that extends $\mathcal{N}$, we let $V|_{X_i:\vec{d}}$ and $\mathcal{N}|_{X_i:\vec{d}}$ denote the the projection of V (or, equivalently $\mathcal{N}$) to X_i, given $\vec{d}$. That is, each of these notations evaluates to the linear order $\succ^i_{\vec{d}}$ in the CPT associated with X_i. For example, let $\mathcal{N}$ be the CP-net defined in Example 8.2.2. $\mathcal{N}|_{\mathbf{W}:b} = r \succ p \succ w$. For any $\mathcal{O}$-legal profile P, $P|_{X_i:\vec{d}}$ is the profile over D_i that is composed of the projections of each vote in P on X_i, given $\vec{d}$. That is, $P|_{X_i:\vec{d}} = (V_1|_{X_i:\vec{d}}, \ldots, V_n|_{X_i:\vec{d}}) = (\mathcal{N}_1|_{X_i:\vec{d}}, \ldots, \mathcal{N}_n|_{X_i:\vec{d}})$, where $P = (V_1, \ldots, V_n)$, and for any $1 \leqslant i \leqslant p$, V_i extends $\mathcal{N}_i$.

Let $\mathcal{O} = X_1 > \cdots > X_p$. The *lexicographic extension* of an $\mathcal{O}$-compatible CP-net $\mathcal{N}$ w.r.t. $\mathcal{O}$, denoted by $Lex_{\mathcal{O}}(\mathcal{N})$, is an $\mathcal{O}$-legal linear order V over $\mathcal{X}$ such that for any $1 \leqslant i \leqslant p$, any $\vec{d_i} \in D_1 \times \cdots \times D_{i-1}$, any $a_i, b_i \in D_i$, and any $\vec{y}, \vec{z} \in D_{i+1} \times \cdots \times D_p$, if $a_i \succ_{\mathcal{N}|_{X_i:\vec{d_i}}} b_i$, then $(\vec{d_i}, a_i, \vec{y}) \succ_V (\vec{d_i}, b_i, \vec{z})$. Intuitively, in the lexicographic extension of $\mathcal{N}$, X_1 is the most important issue, X_2 is the next-most important issue, and so on; a desirable change to an earlier issue always outweighs any changes to later issues. We note that the lexicographic extension of any CP-net is unique w.r.t. the order $\mathcal{O}$. Again, the subscript "$\mathcal{O}$" is sometimes omitted when there is no risk of confusion. We say that $V \in L(\mathcal{X})$ is *lexicographic* if it is the lexicographic extension of a CP-net $\mathcal{N}$. For example, let $\mathcal{N}$ be the CP-net defined in Example 8.2.2. We have $Lex(\mathcal{N}) = br \succ bp \succ bw \succ fw \succ fp \succ fr \succ sp \succ sw \succ sr$. A profile P is $\mathcal{O}$-legal/separable/lexicographic, if each of its votes is in $Legal(\mathcal{O})$/ is separable/ is lexicographic.

8.3 Sequential Voting

One natural approach to combinatorial voting is sequential voting. Let $\mathcal{O}$ denote be an ordering over the issues. W.l.o.g. $\mathcal{O} = X_1 > X_2 > \cdots > X_p$. Sequential voting selects the winner in p rounds. In round i, the voters report their preferences over the ith issue in $\mathcal{O}$, based on which the winning value is selected by applying a local voting rule, and this winning value is then announced to all the voters. The idea of sequential voting is not new. For example, Lacy and Niou (2000) suggested to use sequential voting to circumvent multiple-election paradoxes. But, again, in the sequential voting process they proposed, voters are sometimes ill at ease reporting their preferences over issues, and the voters are still assumed to behave optimistically.[3] Moreover, Lacy and Niou argued that the sequential voting process "*takes too long*," because the voters must wait for the results of previous issues to be announced before moving to the subsequent issues. They argued that "*the cost to voters of going to the polls and the cost to governments of keeping polls open for several days will likely prevent the use of sequential voting schemes*" (Lacy and Niou, 2000).

In fact, the voters do not need to go to voting booths multiple times. It suffices

[3] Lacy and Niou (2000) also suggested to let voters vote strategically and sequentially, and showed that the outcome will always be the Condorcet winner whenever one exists. This is the strategic sequential voting procedure that will be discussed in Chapter 11.

for them to report in one shot all their local preferences over single issues, given all (relevant) valuations of the previous issues. That is, to apply sequential voting w.r.t. the order $\mathcal{O}$ over issues, it suffices for the voters to use an $\mathcal{O}$-compatible CP-net to represent their preferences. Of course the voters need to report more of their preferences, and some of them are not used for the voting rule to decide the winner. This is not a big problem as long as the language is compact (as we will see later in this section). Similarly to the situation in issue-by-issue voting (where only the voters with separable preferences are comfortable with reporting their preferences), in sequential voting we have a similar criterion: if a voter's preferences are $\mathcal{O}$-legal, then she is comfortable with reporting their preferences; otherwise she is not comfortable with reporting her preferences.

The ground-breaking systematic method for analyzing sequential voting was proposed by Lang (2007), who focused on the profiles where voters are comfortable reporting their preferences (that is, $\mathcal{O}$-legal profiles), and defined sequential voting rules on top of these profiles.

Definition 8.3.1. *(Lang, 2007) Given a vector of* local rules $(r_1, \ldots, r_p)$*, where for each* $1 \leqslant i \leqslant p$*,* r_i *is a voting rule on* D_i*, the* sequential composition *of* $r_1, \ldots, r_p$ *w.r.t.* $\mathcal{O}$*, denoted by* $Seq_{\mathcal{O}}(r_1, \ldots, r_p)$*, is defined for all* $\mathcal{O}$*-legal profiles as follows:* $Seq_{\mathcal{O}}(r_1, \ldots, r_p)(P) = (d_1, \ldots, d_p) \in \mathcal{X}$*, so that for any* $1 \leqslant i \leqslant p$*,* $d_i = r_i(P|_{X_i:d_1\cdots d_{i-1}})$*.*

The *sequential composition* of *local correspondences* $r^c_1, \ldots, r^c_p$, denoted by $Seq_{\mathcal{O}}(r^c_1, \ldots, r^c_p)$, is defined in a similar way: for any $\mathcal{O}$-legal profile P,
$\vec{d} \in Seq_{\mathcal{O}}(r^c_1, \ldots, r^c_p)(P)$ if and only if for each $i \leqslant p$, we have that $d_i \in r^c_i(P|_{X_i:d_1\ldots d_{i-1}})$.

The subscript "$\mathcal{O}$" in $Seq_{\mathcal{O}}$ is sometimes omitted when there is no risk of confusion. We note that when a voter's preferences are $\mathcal{O}$-legal, she only needs to submit an $\mathcal{O}$-compatible CP-net instead of reporting the entire $\mathcal{O}$-legal linear order. Hence, the voting language used by a sequential rule is essentially the set of all $\mathcal{O}$-compatible CP-nets. Similarly, the voting language used in issue-by-issue voting is essentially the set of all separable CP-nets (where there are no edges in the graph). We note that if the voters' profile is separable, then sequential voting rules become issue-by-issue voting rules. In that sense, sequential voting rules are extensions of issue-by-issue voting rules.

To examine the compactness of the set of all $\mathcal{O}$-compatible CP-nets as a voting language, let us calculate the size of an $\mathcal{O}$-compatible CP-net (which is the sum of the sizes of all CPTs). It is easy to see that the size of a CP-net largely depends on how many parents each issue has in the graph. In fact, the size of a CP-net is

$$\sum_{i=1}^{p} \prod_{X_j \in Par_G(X_i)} |D_j| \log(|D_i|!)$$

Therefore, if both the number of members in each local domain and the number of parents for each issue are small, then the size of the CP-net is polynomial in the number of issues (for comparison, we recall that in multi-issue domains we need $\Theta(p \cdot 2^p)$ bits to represent a linear order, which is exponential in the number of issues); on the other hand, in the worst case the size of an $\mathcal{O}$-compatible CP-net is exponentially large in the number of issues. However, in practice it is reasonable to expect that all local domains are small, and the voters' preferences over each issue only depends on a few other issues. Hence, we can expect in practice that $\mathcal{O}$-compatible CP-nets are a compact language. Obviously $\mathcal{O}$-compatible CP-nets, as a voting language, are more expressive than the language used by issue-by-issue voting (that is, separable CP-nets), simply because separable CP-nets are special cases of $\mathcal{O}$-compatible CP-nets. In fact, it has been shown that the ratio between the number of $\mathcal{O}$-legal linear orders and the number of separable linear orders is

$\Omega(\frac{2^p}{\sqrt{\pi p}})$ (Lang and Xia, 2009), which in some sense shows quantitatively how much more usable $\mathcal{O}$-compatible CP-nets are, compared to separable CP-nets. Table 8.1 provides a comparison of plurality, common voting rules that require voters to report linear orders (e.g., Borda), issue-by-issue voting rules, sequential voting rules, the framework introduced in Chapter 9, and the MLE approach taken in Chapter 10, in terms of the following three aspects: (1) computational efficiency of computing the winner, (2) compactness of the voting language, and (3) expressiveness of the voting language, which includes general usability and informativeness.

Table 8.1: Comparing voting rules and languages for combinatorial voting.

Voting method	Computational efficiency	Compactness	Expressiveness	
			General usability	Informativeness
Plurality	High	High	High	Low
Borda, etc.	Low	Low	High	High
Issue-by-issue	High	High	Low	Medium
Sequential voting	High	Usually high	Medium	Medium
H-composition in Chapter 9	Low–High (depends on the voters' common preference structure)	Usually high	High	Medium
MLE approach in Chapter 10	Low–High (depends on the probabilistic model)	Usually high	High	Medium

For (truthful) sequential voting, multiple-election paradoxes are alleviated (Lacy and Niou, 2000; Lang and Xia, 2009), though they return when voters vote strategically, as we will see in Chapter 11. One natural question to ask is whether sequential voting rules satisfy some other desired axiomatic properties for voting rules (see Section 2.2). Not surprisingly, the answer depends on whether the local voting rules satisfy these axiomatic properties. Lang and Xia (2009) asked the following two questions for any axiomatic property Y.

1. If the sequential voting rule satisfies Y, is it true that all its local voting rules also satisfy Y? This corresponds to the "Global to local" column in Table 8.2.
2. If all local voting rules satisfy Y, is it true that their sequential composition also satisfies Y? This corresponds to the "Local to global" column in Table 8.2.

The answers for some of the axiomatic properties described in Section 2.2 are summarized in Table 8.2.[4]

Table 8.2: Local vs. global for sequential rules (Lang and Xia, 2009).

Criteria	Global to local	Local to global
Anonymity	Y	Y
Neutrality	Y	N
Consistency	Y	Y
Participation	Y	N
Pareto efficiency	Y	N
(Strong) monotonicity	Y	Y

For neutrality and Pareto efficiency, Xia and Lang (2009) showed that the existence of voting correspondences that satisfy neutrality (respectively, Pareto efficiency) can be characterized by the structure of the multi-issue domain: if the multi-issue domain is composed of two binary variables, then there exists a voting correspondence that satisfies neutrality (respectively, Pareto efficiency); otherwise no voting correspondence satisfies neutrality (respectively, Pareto efficiency).[5]

Nevertheless, we may still argue that in order for voters to feel comfortable expressing their preferences, sequential voting is quite restrictive at two levels: first, at the individual voters' level, sequential voting requires that a voter's preferences must be represented by an acyclic CP-net. Second, at the profile level, it requires

[4] Since sequential voting rules are defined for $\mathcal{O}$-legal profiles, the definitions of neutrality, Pareto efficiency, and monotonicity are slightly different. See Lang and Xia (2009).

[5] Again, here the definitions of neutrality and efficiency are slightly different from the definitions in Section 2.2.

that all voters' preferences are compatible with the same ordering $\mathcal{O}$. To overcome these restrictions, we need to consider even more expressive voting languages. One option is the set of all (possibly cyclic) CP-nets. Obviously it is more expressive, because it is a superset of the set of all acyclic CP-nets. Chapters 9 and 10 aim at designing new voting rules for combinatorial voting where the voters use (possibly cyclic) CP-nets to represent their preferences. In Chapter 9, we will further show how much more general (possibly cyclic) CP-nets are, by showing the ratio between the number of $\mathcal{O}$-legal votes and the number of all linear orders over $\mathcal{X}$ (note that any voter should be comfortable with using a possibly cyclic CP-net to represent her preferences, in the sense that for any linear order, a possibly cyclic CP-net can be constructed such that the linear order extends this CP-net). Then, we propose an extension of sequential voting rules to aggregate (possibly cyclic) CP-nets, which we call *hypercubewise composition (H-composition)*. We will analyze its normative and computational aspects. This framework was further studied by Li et al. (2011) and Conitzer et al. (2011b). In Chapter 10, we extend Condorcet's MLE model to combinatorial voting.

Chapters 11 and 12 investigate game-theoretic aspects of combinatorial voting. In Chapter 11 we study the sequential voting game mentioned by Lacy and Niou (2000), that is, the game where voters cast votes strategically on one issue after another, following some ordering over the issues. We call this type of voting games the *strategic sequential voting procedure (SSP)*. Lacy and Niou (2000) proved that strategic sequential voting[6] always selects the Condorcet winner whenever one exists, but they did not examine whether there are any multiple-election paradoxes for SSP. In Chapter 11 we show that all three types of multiple-election paradoxes still arise in strategic sequential voting. Moreover, changing the ordering of the issues according to which the voters vote on them cannot avoid at least the first and the

[6] They called it *sophisticated sequential voting*, following the convention of Farquharson (1969).

third paradoxes. Then, in Chapter 12, we will see how to restrict voters' preferences over multi-issue domains to obtain strategy-proof voting rules.

At the end of this chapter, let me briefly mention some other work in preference aggregation over multi-issue domains. Rossi et al. (2004) studied aggregating voters' preferences represented by *partial CP-nets*, which allows voters to be "indifferent" with between the values of some issues. Gonzales et al. (2008) studied aggregating preferences represented by another compact language called *GAI-networks*. Xia et al. (2007a) slightly extended sequential voting rules by removing the constraint that the order $\mathcal{O}$ is fixed before the voting process. However, the above two levels of restrictions for sequential voting rules still exist. Recently, Conitzer et al. (2009a) studied the agenda control problem in sequential voting—that is, the chair gets to choose the over in which the issues are voted on, and investigated its computational complexity.

8.4 Summary

In this chapter, we introduced the notation used in this dissertation for combinatorial voting, multiple-election paradoxes, CP-nets, sequential voting rules, and important criteria for designing new voting rules in combinatorial domains. We also evaluated voting rules proposed in previous work by these criteria, and the result is summarized in Table 8.1. We observed that all previous approach either used voting languages that lack expressivity, or is computationally intractable.

9

A Framework for Aggregating CP-nets

In this chapter, we define a new family of voting rules for combinatorial voting that allows voters to use any CP-nets (even cyclic ones) to represent their preferences. The set of all CP-nets, as a language, is compact and is much more expressive than acyclic CP-nets or separable CP-nets, which are used in sequential voting and issue-by-issue voting, respectively.[1] The voting rules we define are parameterized by: (1) the local voting rules that are used on individual attributes—we will use these to define a particular graph on the set of alternatives; and (2) a *choice set function* T that selects the winners based on this induced graph.[2] We show that if T satisfies a very natural assumption, then the voting rules induced by T extend the sequential voting rules (and therefore, also issue-by-issue voting rules) and the *order-independent sequential composition* of local rules from Xia et al. (2007b). We study whether properties of the local rules transfer to the global rule, and vice versa. Then, we focus on a particular choice set function, namely the *Schwartz set* (Schwartz, 1970), which has been argued

[1] Earlier work has also considered social choice for potentially cyclic CP-nets (Purrington and Durfee, 2007). However, that approach does not apply to all possible (cyclic) CP-nets.

[2] In principle, T could select multiple winners from the graph. We can use any tie-breaking mechanism to select the unique winner.

to be the largest reasonable choice set for tournament graphs (Laslier, 1997). For the Schwartz set, we study how to compute the winners under this methodology.

9.1 Acyclic CP-nets Are Restrictive

In this section, we show quantitatively that the set of all acyclic CP-nets lacks general usability as a voting language. We will show that even when each local domain is binary, the number of legal linear orders—the set of all linear orders $\succ$ for which there is some acyclic CP-net that $\succ$ extends—is exponentially smaller than the number of all linear orders. Let $CP(\mathcal{X}) = \{V \in L(\mathcal{X}) :$ There exists a CP-net $\mathcal{N}$ such that $V \sim \mathcal{N}\}$. That is, $CP(\mathcal{X}) = \bigcup_{\mathcal{O}} Legal(\mathcal{O})$.

Theorem 9.1.1. *If* $\mathcal{X} = \{0,1\}^p$*, then* $\dfrac{|CP(\mathcal{X})|}{|L(\mathcal{X})|} \leqslant \dfrac{p!}{2^{2^{p-2}}}$.

Proof. We prove the theorem by constructing a set of exponentially many permutations on the set of alternatives, and we prove that for any two different linear orders compatible with the same order over attributes, for any two (not necessarily different) permutations in the set, if we apply the first permutation to the first linear order and the second permutation to the second linear order, the results are different. That is, for any linear order compatible with a given order $\mathcal{O}$, we can find a large set of corresponding linear orders by applying the set of permutations to it; and the sets of linear orders corresponding to different $G_{\mathcal{O}}$-legal linear orders are disjoint.

More precisely, we define a set of permutations on $\mathcal{X}$, denoted by $K(\mathcal{X})$, and show that it satisfies the following two properties:

1. $|K(\mathcal{X})| = 2^{2^{p-2}}$.

2. For any $V_1, V_2 \in Legal(X_1 > \cdots > X_p)$ and any $M_1, M_2 \in K(\mathcal{X})$, if $M_1 \neq M_2$, then $M_1(V_1) \neq M_2(V_2)$.

Now we show how to construct $K(\mathcal{X})$. Given any setting $\overrightarrow{x_{p-2}}$ of $(X_1, \ldots, X_{p-2})$, let $M_{\overrightarrow{x_{p-2}}}$ be the permutation that only exchanges $(\overrightarrow{x_{p-2}}, 0, 0)$ and $(x_{p\vec{-}2}, 0, 1)$. Then, for any $E_{p-2} \subseteq \{0,1\}^{p-2}$, let $M_{E_{p-2}} = \circ_{\overrightarrow{x_{p-2}} \in E_{p-2}} M_{\overrightarrow{x_{p-2}}}$, where $\circ$ is the composition of two permutations. This notion is well-defined because for any $\overrightarrow{x_{p-2}}, \overrightarrow{x_{p-2}}' \in \{0,1\}^{p-2}$, $M_{\overrightarrow{x_{p-2}}}$ and $M_{\overrightarrow{x_{p-2}}'}$ are exchangeable, that is, $M_{\overrightarrow{x_{p-2}}} \circ M_{\overrightarrow{x_{p-2}}'} = M_{\overrightarrow{x_{p-2}}'} \circ M_{\overrightarrow{x_{p-2}}}$.

Let $K(\mathcal{X}) = \{M_{E_{p-2}} : E_{p-2} \subseteq \{0,1\}^{p-2}\}$. Then, $|K(\mathcal{X})| = |\{0,1\}^{\{0,1\}^{p-2}}| = 2^{2^{p-2}}$. For any $V_1, V_2 \in \mathit{Legal}(X_1 > \cdots > X_p)$ any $M_{E^1_{p-2}}, M_{E^2_{p-2}} \in K(\mathcal{X})$ such that $M_{E^1_{p-2}} \neq M_{E^2_{p-2}}$, since $E^1_{p-2} \neq E^2_{p-2}$, w.l.o.g. there exists $\overrightarrow{x_{p-2}}$ such that $\overrightarrow{x_{p-2}} \in E^1_{p-2}$ but $\overrightarrow{x_{p-2}} \notin E^2_{p-2}$. Then, $V_1|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}}$ extends a CP-net on $\{0,1\}^2$, and the CP-net is compatible with $X_{p-1} > X_p$. Here, $V_1|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}}$ is the restriction of V_1 to $\{X_{p-1}, X_p\}$, given that $(X_1, \ldots, X_{p-2}) = \overrightarrow{x_{p-2}}$. However, $M_{\overrightarrow{x_{p-2}}}(V_2)|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}}$ is not compatible with $X_{p-1} > X_p$—it either does not extend a CP-net, or extends a CP-net that is not compatible with $X_{p-1} > X_p$. We note that

$$M_{E^1_{p-2}}(V_1)|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}} = V_1|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}},$$

$$M_{E^2_{p-2}}(V_2)|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}} = M_{\overrightarrow{x_{p-2}}}(V_2)|_{\{X_{p-1},X_p\}:\overrightarrow{x_{p-2}}}$$

Hence, $M_{E^1_{p-2}}(V_1) \neq M_{E^2_{p-2}}(V_2)$, which means that $K(\mathcal{X})$ satisfies the two properties mentioned above.

Therefore, from the two properties of $K(\mathcal{X})$, we know that $|K(\mathcal{X})(\mathit{Legal}(X_1 > \ldots > X_p))| = 2^{2^{p-2}}|\mathit{Legal}(X_1 > \ldots > X_p)|$. Since $K(\mathcal{X})(\mathit{Legal}(X_1 > \ldots > X_p)) \subseteq L(\mathcal{X})$, and $|CP(\mathcal{X})| < p!|L(X_1 > \ldots > X_p)|$ (because there are $p!$ linear orders over $\{x_1, \ldots, x_p\}$, and a CP-net must be compatible with some order), we have $\dfrac{|CP(\mathcal{X})|}{|L(\mathcal{X})|} \leqslant \dfrac{p!}{2^{2^{p-2}}}$.

□

We note that $|\mathcal{X}| = 2^p$. Theorem 9.1.1 implies that the expressivity ratio of legal linear orders ($\dfrac{|CP(\mathcal{X})|}{|L(\mathcal{X})|}$) is $O((2^{0.2})^{-|\mathcal{X}|})$, which is exponentially small even in the number of alternatives.

9.2 H-Composition of Local Voting Rules

In this section, we introduce a new framework for composing local voting rules. We call this *hypercubewise composition (H-composition)* of local voting rules. The reason is that the outcome only depends on preferences between alternatives that differ on only one attribute. We can visualize the set of all alternatives as a hypercube, and alternatives that differ on only one attribute are neighbors on this hypercube, as discussed in Section 8.2. An H-composition of local rules is defined for all profiles in which for each vote, there exists a (possibly cyclic) CP-net that it extends. In fact, for any linear order V on $\mathcal{X}$, there exists a CP-net $\mathcal{N}$ such that V extends $\mathcal{N}$, so we can apply this to *any* linear orders (but also some partial orders). By Theorem 9.1.1, this means that the voting language used by these H-compositions (i.e., possibly cyclic CP-nets) is much more general than the voting language used by sequential voting rules (i.e., $\mathcal{O}$-compatible CP-nets for some ordering $\mathcal{O}$ over $\mathcal{I}$, in the sense we have discussed in Section 8.3.

An H-composition of local rules is defined in two steps. In the first step, an *induced graph* is generated by applying local rules to the input profile. Then, in the second step, a *choice set function* is selected based on the induced graph as the set of winners (the definition and examples of some major choice set functions are deferred to Definition 9.2.4 and the text below it). We first define the induced graph of P w.r.t. local rules (or correspondences) $r_1, \ldots, r_p$.

Definition 9.2.1. *Given a profile $P = (V_1, \ldots, V_n)$ and local rules (or correspondences) $r_1, \ldots, r_p$, the* induced graph *of P w.r.t. $r_1, \ldots, r_p$, denoted by $IG(r_1, \ldots, r_p)(P) = (\mathcal{X}, E)$, is defined by the following edges between alternatives. For any $i \leqslant p$, any setting $\overrightarrow{x_{-i}}$, let $C_i = r_i(P|_{X_i:\overrightarrow{x_{-i}}})$; for any $c_i \in C_i$, any $d_i \in D_i$, let there be an edge*

$(c_i, \overrightarrow{x_{-i}}) \to (d_i, \overrightarrow{x_{-i}})$.

Example 9.2.2. *Suppose the multi-issue domain consists of two binary attributes:* $\mathbf{S}$ *ranging over* $\{S, \bar{S}\}$ *and* $\mathbf{T}$ *ranging over* $\{T, \bar{T}\}$. *The local rules are both the majority rule. Two votes* V_1, V_2 *and their induced graph* $IG(Maj, Maj)(V_1, V_2)$ *are illustrated in Figure 9.2.2, where* Maj *denotes the majority correspondence. We note that* V_1 *is compatible with* $\mathbf{S} > \mathbf{T}$, V_2 *is compatible with* $\mathbf{T} > \mathbf{S}$.

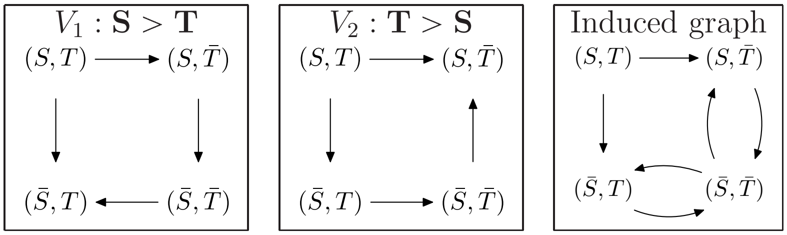


FIGURE 9.1: Two votes and their induced graph.

Next, we define the dominance relation in a directed graph.

Definition 9.2.3. *Given a directed graph* $G = (\mathcal{V}, E)$, *for any* $v_1, v_2 \in \mathcal{V}$, v_1 *is said to* dominate v_2, *denoted by* $v_1 \succ_G v_2$, *if and only if:*

1. there is a directed path from v_1 *to* v_2, *and*

2. there is no directed path from v_2 *to* v_1.

Let $\succeq_G$ be the *transitive closure* of E, that is, $\succeq_G$ is the minimum preorder such that if $(v_1, v_2) \in E$, then $v_1 \succeq_G v_2$. Then, another equivalent way to define the dominance relation is: $\succ_G$ is the strict order induced by $\succeq_G$, that is, $v_1 \succ_G v_2$ if and only if $v_1 \succeq_G v_2$ and $v_2 \not\succeq_G v_1$.

We further define two kinds of special vertices in a directed graph G as follows. The first is a vertex that dominates all the other vertices, and the second is a vertex that dominates all its neighbors. We call the former the *global Condorcet winner* (which must be unique), and the latter a *local Condorcet winner*.

Now, we are ready to define the *choice set function*, which specifies a *choice set* for each graph.

Definition 9.2.4. *A* choice set function T *is a mapping from any graph to a subset of its vertices.*

We now recall the definitions of some major choice sets in a graph $G = (V, E)$.

- The *Schwartz set* is the union of all maximal mutually connected subsets. A maximal mutually connected subset is a subset of vertices such that there is a path between any two vertices in the set, but there is no path from a vertex outside the set to a vertex inside the set.

- The *Smith set* is the smallest set of vertices such that every vertex in the set dominates all the vertices outside the set.

- The *Copeland set*: A vertex c's Copeland score is the number of vertices that are dominated by c minus the number of vertices that dominate c. The vertices with the highest Copeland score are the winners.

Choice sets were originally introduced to make group decisions for tournament graphs. However, the definitions are easily extended to general graphs, as we did above. See Laffond et al. (1995) and Brandt et al. (2007) for more discussion.

We say a choice set function T *always chooses the global Condorcet winner*, if for any graph $G = (V, E)$ in which c is the global Condorcet winner, we have $T(G) = \{c\}$. We say that T *always chooses local Condorcet winners*, if every local Condorcet winner is always in $T(G)$. We emphasize that here, the meaning of a Condorcet winner is different from traditional meaning of a Condorcet winner, which refers to an alternative that wins every pairwise election. We say that T is *monotonic*, if for any graph $(\mathcal{V}, E)$, any $c \in T(\mathcal{V}, E)$, and any $(\mathcal{V}, E')$ that is obtained from $(\mathcal{V}, E)$ by only flipping some of the incoming edges of c, we have $c \in T(\mathcal{V}, E')$.

Theorem 9.2.5 (known/easy)**.** *The Schwartz set, Smith set, and Copeland set are monotonic and always choose the global Condorcet winner; the Smith set and Schwartz set always choose local Condorcet winners.*

We are now ready to define the H-composition of local rules (correspondences).

Definition 9.2.6. *Let T be a choice set function. The Hypercubewise-T (H-T) composition of local rules $r_1, \ldots, r_p$, denoted by $H_T(r_1, \ldots, r_p)$, is defined as follows. For any profile P of linear orders on $\mathcal{X}$,*

$$H_T(r_1, \ldots, r_p)(P) = T(IG(r_1, \ldots, r_p)(P))$$

That is, for any profile P, $H_T(r_1, \ldots, r_p)$ computes the winner in the following two steps. First, the induced graph $IG(r_1, \ldots, r_p)(P)$ is generated by applying local rules $r_1, \ldots, r_p$ to the restrictions of P to all the local domains. Then, in the second step, the set of winners is selected by the choice set function T from the induced graph $IG(r_1, \ldots, r_p)(P)$.

From Theorem 9.1.1, the fact that all linear orders are consistent with some CP-net, and all CP-nets can be used under H-composition, we know that the domain of H-composition of local rules is exponentially larger than the domain of order-independent sequential composition. We note that to build the induced graph, only the preferences between adjacent alternatives are necessary. We note that the H-composition of local rules is a correspondence, and we can use any tie-breaking mechanism to select a unique winner.

One interesting question is how H-compositions are related to (order-independent) composition of local rules. Because the H-compositions are defined by both local rules and the choice set, the relationship should also depend on local rules and the properties of the choice set. The next theorem states that if a choice set function T always chooses the global Condorcet winner, then H-T composition of local rules is an extension of order-independent sequential composition of the same local rules (Xia et al., 2007b). The *order-independent sequential composition* of local rules, denoted by $Seq^{OI}(r_1, \ldots, r_p)$, extends the domain of sequential composition of local rules to the set of all legal profiles P, which means that the order $\mathcal{O}$ is not held fixed in the definition. For any permutation σ on $\{1, \ldots, p\}$, let $\mathcal{O} = X_{\sigma(1)} > \ldots > X_{\sigma(p)}$. Then, for any $\mathcal{O}$-legal profile P, $Seq^{OI}(r_1, \ldots, r_p)(P) = Seq_{\mathcal{O}}(r_{\sigma(1)}, \ldots, r_{\sigma(p)})(P)$. The order-independent sequential composition of local correspondences is defined similarly. This voting rule is well-defined because it has been shown in Lang (2007) that the winner does not depend on which ordering $\mathcal{O}$ that is used in the definition, as long as the profile is $\mathcal{O}$-legal.

Theorem 9.2.7. *Let T be a choice set function that always chooses the global Condorcet winner. Then, for all legal profiles P, $H_T(r_1, \ldots, r_p)(P) = Seq^{OI}(r_1, \ldots, r_p)(P)$.*

The proof is quite straightforward and is thus omitted.

Corollary 9.2.8. *If T is the Schwartz set, Smith set, or Copeland set, then $H_T(r_1, \ldots, r_p)$ is an extension of $Seq^{OI}(r_1, \ldots, r_p)$.*

9.3 Local vs. Global Properties

In this section we examine the "quality" of the H-compositions of local rules in terms of whether they satisfy some common voting axioms described in Section 2.2. We recall that in Section 8.3 we have asked a similar question for sequential voting rules, and whether sequential voting rule satisfies some desired axiomatic properties depends on whether the local voting rules satisfy these axiomatic properties. Lang and Xia (2009) asked the following two questions for any axiomatic property Y, and the answers are summarized in Table 8.2.

1. If the sequential voting rule satisfies Y, is it true that all its local voting rules satisfy Y?

2. If the sequential voting rule satisfies Y, is it true that all its local voting rules satisfy Y?

For H-composition of local rules, we can ask the same question. From Theorem 9.2.7 we know that if T always chooses the global Condorcet winner, then H_T is an extension of Seq^{OI}. We can use this observation to carry over some of the results in Lang (2007); Xia et al. (2007a,b) to H_T. Specifically, if T always chooses the global Condorcet winner, and if a criterion transfers from the order-independent sequential composition of local rules to each local rule, then it also transfers for H-T composition; if a criterion does not transfer from local rules to their order-independent sequential composition, then it also does not transfer for H-T composition. Given the results in Xia et al. (2007b), these observations allow us to resolve everything except how anonymity, homogeneity, monotonicity, and consistency transfer from local rules to their H-T composition. It is easy to see that anonymity and homogeneity always transfer. The next example shows that if T always chooses local Condorcet winners, then consistency does not transfer, even when the votes in the profile extend (possibly different) acyclic CP-nets.

Example 9.3.1. *Let $\mathcal{X} = \{0_1, 1_1\} \times \{0_2, 1_2\} \times \{0_3, 1_3\}$, and let all the local rules be the majority rule. Consider the following three CP-nets (the non-specified parts of the CPTs do not matter):*

$\mathcal{N}_1$: *compatible with $X_1 > X_2 > X_3$, and $1_1 \succ 0_1$, $1_1 : 1_2 \succ 0_2$, $1_1 1_2 : 1_3 \succ 0_3$, $0_1 : 0_2 \succ 1_2$, $0_1 0_2 : 0_3 \succ 1_3$.*

$\mathcal{N}_2$: *compatible with $X_2 > X_3 > X_1$, and $1_2 \succ 0_2$, $1_2 : 1_3 \succ 0_3$, $1_2 1_3 : 1_1 \succ 0_1$, $0_2 : 0_3 \succ 1_3$, $0_2 0_3 : 0_1 \succ 1_1$.*

$\mathcal{N}_1$: *compatible with $X_3 > X_1 > X_2$, and $1_3 \succ 0_3$, $1_3 : 1_1 \succ 0_1$, $1_3 1_1 : 1_2 \succ 0_2$, $0_3 : 0_1 \succ 1_1$, $0_3 0_1 : 0_2 \succ 1_2$.*

For any V_1, V_2, V_3 extending $\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3$, respectively, let $P = (V_1, V_2, V_3)$. Let $H_T(M) = H_T(Maj, Maj, Maj)$. Suppose ties are broken in favor of $0_1 0_2 0_3$. Because $0_1 0_2 0_3$ is a local Condorcet winner, so $H_T(M)(P) = 0_1 0_2 0_3$. However, $H_T(M)(V_1) = H_T(M)(V_2) = H_T(M)(V_3) = 1_1 1_2 1_3$, so $H_T(M)$ does not satisfy consistency (because otherwise, we must have $H_T(M) = 1_1 1_2 1_3$, which we know is not the case).

The next proposition states that for any monotonic choice set function T, the monotonicity is transferred from local rules to their H-T composition. The proof is quite straightforward and is omitted.

Proposition 9.3.2. *Let T be a monotonic choice set function. If all local rules $\{r_1, \ldots, r_p\}$ satisfy monotonicity, then $H_T(r_1, \ldots, r_p)$ also satisfies monotonicity.*

For choice sets T that always choose the global Condorcet winner, whether properties of local rules transfer to their H-T composition and vice versa is summarized in Table 9.1.

Table 9.1: Local vs. global for H-compositions.

Criteria	Global to local	Local to global
Anonymity	Y	Y
Homogeneity	Y	Y
Neutrality	Y	N
Monotonicity	Y	Y for monotonic T
Consistency	Y	N if T always chooses local Condorcet winner
Participation	Y	N
Pareto efficiency	Y	N

9.4 Computing H-Schwartz Winners

Among all choice sets, we are most interested in the Schwartz set, because first, it has been argued that the Schwartz set is the "largest" reasonable choice set for tournaments (Laslier, 1997), and second, it corresponds to the *nondominated set* previously considered in the context of CP-nets (Boutilier et al., 2004). In this section, we investigate the computational complexity of computing H-Schwartz winners. We note that in this section H-Schwartz is a voting correspondence. Recent work on the complexity of computing dominance relations in CP-nets shows that the *dominance* problem in a CP-net is hard (Goldsmith et al., 2008). More precisely, given a CP-net $\mathcal{N}$ and two alternatives a and b, it is PSPACE-complete to compute whether or not $a \succ_{\mathcal{N}} b$. This can be used to show that checking membership in the Schwartz set is PSPACE-complete (Goldsmith et al., 2008).

Although computing the Schwartz set is hard in general, if the preferences are more structured it can be easy. As an extreme example, if the voters' preferences extend an acyclic CP-net $\mathcal{N}$, then H-Schwartz is equivalent to order-independent sequential composition of local rules, under which computing the winner is easy. In this section, we introduce a technique to exploit more limited independence information in the submitted votes for the purpose of computing the set of H-Schwartz winners.

Definition 9.4.1. *Let $\{\mathcal{I}_1, \ldots, \mathcal{I}_q\}$ ($q \leqslant p$) be a partition of the set of issues $\mathcal{I}$. We say a CP-net $\mathcal{N}$ whose graph is G is* compatible *with the ordering $\mathcal{I}_1 > \cdots > \mathcal{I}_q$ if for any $l \leqslant q$ and any $X \in \mathcal{I}_l$, $Par_G(X) \subseteq \mathcal{I}_1 \cup \ldots \cup \mathcal{I}_l$. A linear order V is compatible with $\mathcal{I}_1 > \ldots > \mathcal{I}_q$ if there exists a CP-net $\mathcal{N}$ such that V extends $\mathcal{N}$ and $\mathcal{N}$ is compatible with $\mathcal{I}_1 > \ldots > \mathcal{I}_q$.*

Let $\mathcal{O} = X_1 > \ldots > X_p$. One special case is the following: if the input profile is $\mathcal{O}$-legal, then we can use the partition $\mathcal{I}_1 = \{X_1\}, \ldots, \mathcal{I}_p = \{X_p\}$. We can use the following algorithm to find a partition with which the input profile P is compatible. Suppose that we already know the graphs of the CP-nets that the votes in P extend.

Algorithm 1

1. Let G_P be the union of all the graphs of the CP-nets that the votes in P extend.

2. Let $q = 0$; repeat step 3 until $G_P = \emptyset$.

3. Let $q \leftarrow q+1$. Find a maximal mutually connected subset of G_P, and call it $\mathcal{I}_q$. Remove $\mathcal{I}_q$ and all edges connecting it to G_P.

4. Output the partition $\mathcal{I}_1 \cup \ldots \cup \mathcal{I}_q$.

This algorithm runs in time $O(p^3)$. Now we are ready to present the technique for computing the set of H-Schwartz winners more efficiently. Suppose the set of attributes can be partitioned into $\mathcal{I}_1 \cup \mathcal{I}_2$ so that P is compatible with $\mathcal{I}_1 > \mathcal{I}_2$. Let $r_{\mathcal{I}_1}$ denote the sub-vector of $(r_1, \ldots, r_p)$ that contains the local rules r_i if and only if $X_i \in \mathcal{I}_1$.

Process 1

1. Compute the Schwartz set $H_{Schwartz}(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1}) = W_1^1 \cup \ldots \cup W_1^k$, where the W_1^i are the maximal mutually connected subsets in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$.

2. For each $i \leqslant k$, let $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i}) = \bigcup_{\vec{w} \in W_1^i} IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:\vec{w}})$; then, compute the Schwartz set W_2^i for $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$.

3. Output $W_p = \bigcup_{i=1}^k W_1^i \times W_2^i$.

The next theorem states that we can compute the winners of $H_{Schwartz}(r_1, \ldots, r_p)(P)$ by Process 1.

Theorem 9.4.2. $W_P = H_{Schwartz}(r_1, \ldots, r_p)(P)$.

Proof. Let $\vec{w}_2$ be a setting of $\mathcal{I}_2$ and $\vec{w}_1, \vec{w}_1'$ be settings of $\mathcal{I}_1$ such that $\vec{w}_1$ and $\vec{w}_1'$ differ only on one attribute. Since P is compatible with $\mathcal{I}_1 > \mathcal{I}_2$, we have that there is an edge from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2)$ in $IG(r_{\mathcal{I}})(P)$ if and only if there is an edge from $\vec{w}_1$ to $\vec{w}_1'$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$. This implies the following claim.

Claim 9.4.1. *If there is a path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2')$ in $IG(r_{\mathcal{I}})(P)$, then its projection on $\mathcal{I}_1$ is a path from $\vec{w}_1$ to $\vec{w}_1'$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$.*

Proof of Claim 9.4.1: W.l.o.g. we only need to prove the case where there is an edge from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2')$ in $IG(r_{\mathcal{I}})(P)$. Because only neighboring alternatives are connected in $IG(r_{\mathcal{I}})(P)$, either $\vec{w}_1 = \vec{w}_1'$ or $\vec{w}_2 = \vec{w}_2'$. If $\vec{w}_1 = \vec{w}_1'$ then the claim is automatically proved, because the projections of the two alternatives in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$ are the same (that is, $\vec{w}_1$). If $\vec{w}_2 = \vec{w}_2'$, then by definition there is an from $\vec{w}_1$ to $\vec{w}_1'$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$. This proves the claim. □

We note that for any $i \leqslant k$, any $\vec{w}_1, \vec{w}_1' \in W_1^i$ such that there is a path from $\vec{w}_1$ to $\vec{w}_1'$, and any $\vec{w}_2 \in D_{\mathcal{I}_2}$, there is a path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2)$. Therefore, we have the following claim.

Claim 9.4.2. *For any $i \leqslant k$, any $(\vec{w}_1, \vec{w}_2), (\vec{w}_1', \vec{w}_2') \in W_1^i \times D_{\mathcal{I}_2}$, there is a path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2')$ if and only if there is a path from $\vec{w}_2$ to $\vec{w}_2'$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$.*

Proof of Claim 9.4.2: We first prove the "only if" part. W.l.o.g. we only need to prove the case where there is an edge from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2')$ in $IG(r_{\mathcal{I}})(P)$. In this case either $\vec{w}_1 = \vec{w}_1'$ or $\vec{w}_2 = \vec{w}_2'$. If $\vec{w}_1 = \vec{w}_1'$, then there is an edge from $\vec{w}_2$ to $\vec{w}_2'$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:\vec{w}_1})$, which means that there is an edge from $\vec{w}_2$ to $\vec{w}_2'$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$. If $\vec{w}_2 = \vec{w}_2'$, then the claim is automatically proved.

Now we prove the "if" part. W.l.o.g. we only need to prove the case where there is an edge from $\vec{w}_2$ to $\vec{w}_2'$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$. By definition of $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$, there exists $\vec{w}_1^* \in W_1^i$ such that there is an edge from $\vec{w}_2$ to $\vec{w}_2'$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:\vec{w}_1^*})$, which means that there is an edge from $(\vec{w}_1^*, \vec{w}_2)$ to $(\vec{w}_1^*, \vec{w}_2')$ in $IG(r_{\mathcal{I}})(P)$. Because W_1^i is a maximum mutually connected set, there exist a path from $\vec{w}_1$ to $\vec{w}_1^*$ and another path from $\vec{w}_1^*$ to $\vec{w}_1'$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$. Because P is compatible with $\mathcal{I}_1 > \mathcal{I}_2$, there exist two paths in $IG(r_{\mathcal{I}})(P)$, one from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1^*, \vec{w}_2)$ and the other from $(\vec{w}_1^*, \vec{w}_2')$ to $(\vec{w}_1', \vec{w}_2')$. These two paths can be connected by the edge from $(\vec{w}_1^*, \vec{w}_2)$ to $(\vec{w}_1^*, \vec{w}_2')$ to form a path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}_1', \vec{w}_2')$. □

Based on Claim 9.4.1 and Claim 9.4.1 we are now ready to prove that $W_P \subseteq H_{Schwartz}(r_{\mathcal{I}})(P)$ and $H_{Schwartz}(r_{\mathcal{I}})(P) \subseteq W_P$, which mean that $W_P = H_{Schwartz}(r_{\mathcal{I}})(P)$.

We first prove that $W_P \subseteq H_{Schwartz}(r_{\mathcal{I}})(P)$. Equivalently, we need to prove that for any $(\vec{w}_1, \vec{w}_2) \in W_P$, there is no alternative $(\vec{w}'_1, \vec{w}'_2) \in \mathcal{X}$ such that (1) there is a path from $(\vec{w}'_1, \vec{w}'_2)$ to $(\vec{w}_1, \vec{w}_2)$, (2) there is no path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}'_1, \vec{w}'_2)$. We prove this by contradiction. Suppose there exists $(\vec{w}'_1, \vec{w}'_2)$ that satisfies the above two conditions. Suppose $\vec{w}_1 \in W_1^i$. By Claim 9.4.1, there is a path from $\vec{w}'_1$ to $\vec{w}_1$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$. Because W_1^i is a maximum mutually connected set, $\vec{w}'_1 \in W_1^i$. By Claim 9.4.2 there exists a path from $\vec{w}'_2$ to $\vec{w}_2$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$. Because $\vec{w}_2 \in W_2^i$, there must exist a path from $\vec{w}_2$ to $\vec{w}'_2$ in $IG(r_{\mathcal{I}_2})(P|_{\mathcal{I}_2:W_1^i})$. Now, by Claim 9.4.2 there exists a path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}'_1, \vec{w}'_2)$, which contradicts the condition (2) above.

Next, we prove that $H_{Schwartz}(r_{\mathcal{I}})(P) \subseteq W_P$. Let $(\vec{w}_1, \vec{w}_2) \in H_{Schwartz}(r_{\mathcal{I}})(P)$. We first show that $\vec{w}_1 \in H_{Schwartz}(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$. Suppose for the sake of contradiction $\vec{w}_1 \notin H_{Schwartz}(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$, then there exists $\vec{w}'_1 \in D_{\mathcal{I}_1}$ such that (1) there is a path from $\vec{w}'_1$ to $\vec{w}_1$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$, and (2) there is no path from $\vec{w}_1$ to $\vec{w}'_1$ in $IG(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$. From (1) we know that there is a path from $(\vec{w}'_1, \vec{w}_2)$ to $(\vec{w}_1, \vec{w}_2)$. From (2) we know that there is no path from $(\vec{w}_1, \vec{w}_2)$ to $(\vec{w}'_1, \vec{w}_2)$, because otherwise by Claim 9.4.1 there is a path from $\vec{w}_1$ to $\vec{w}'_1$, which is a contradiction. It follows that $(\vec{w}_1, \vec{w}_2)$ is dominated by $(\vec{w}'_1, \vec{w}_2)$, which contradicts the assumption that $(\vec{w}_1, \vec{w}_2) \in H_{Schwartz}(r_{\mathcal{I}})(P)$. Therefore, $\vec{w}_1 \in H_{Schwartz}(r_{\mathcal{I}_1})(P|_{\mathcal{I}_1})$.

Now, suppose $\vec{w}_1 \in W_1^i$. If $\vec{w}_2 \notin W_2^i$, then there exists $\vec{w}'_2 \in W_2^i$ that dominates $\vec{w}_2$. However, it follows from Claim 9.4.2 that $(\vec{w}_1, \vec{w}'_2)$ dominates $(\vec{w}_1, \vec{w}_2)$ in $IG(r_{\mathcal{I}})(P)$, which contradicts the assumption that $(\vec{w}_1, \vec{w}_2) \in H_{Schwartz}(r_{\mathcal{I}})(P)$. It follows that $\vec{w}_2 \in W_2^i$, which means that $(\vec{w}_1, \vec{w}_2) \in W_P$.

Therefore, $W_P = H_{Schwartz}(r_{\mathcal{I}})(P)$, which completes the proof. □

If the decomposition is $\mathcal{I}_1 > \ldots > \mathcal{I}_q$ with $q > 2$, then Process 1 can be applied recursively to find the Schwartz set, as follows. First, compute the Schwartz set over $\mathcal{I}_1 \times \mathcal{I}_2$ by Process 1, then use this result to compute the Schwartz set over $(\mathcal{I}_1 \times \mathcal{I}_2) \times \mathcal{I}_3$, etc. up to $(\mathcal{I}_1 \times \ldots \times \mathcal{I}_{q-1}) \times \mathcal{I}_q$.

The next example shows how Process 1 works.

Example 9.4.3. *Let $\mathcal{X} = \{0,1\}^3$, and let three votes V_1, V_2, V_3 extend three CP-nets such that V_1 is $(X_1 > X_2 > X_3)$-legal, V_2 is $(X_2 > X_1 > X_3)$-legal, and V_3 is separable. Let the partition be $\mathcal{I}_1 = \{X_1, X_2\}$, $\mathcal{I}_2 = \{X_3\}$. Then, for all $i = 1,2,3$, V_i is compatible with $\mathcal{I}_1 > \mathcal{I}_2$. Suppose that $\{\vec{w}_1, \vec{w}_1'\} = H_{Schwartz}(r_1, r_2)(P|_{\{X_1,X_2\}})$, so that there is no path from $\vec{w}_1$ to $\vec{w}_1'$, and vise versa. Also suppose that $\{\vec{w}_2\} = H_{Schwartz}(r_3)(P|_{X_3:X_{-3}=\vec{w}_1})$ and $\{\vec{w}_2'\} = H_{Schwartz}(r_3)(P|_{X_3:X_{-3}=\vec{w}_1'})$. Then, the winners are $(\vec{w}_1, \vec{w}_2)$ and $(\vec{w}_1', \vec{w}_2')$.*

The next theorem states that if P is compatible with $\mathcal{I}_1 > \ldots > \mathcal{I}_q$, then the time required to compute the set of Schwartz winners by applying Process 1 is a polynomial function of the number of winners, the longest time it takes to apply local rules, p, n, and $\max|D_{\mathcal{I}_i}|$.

Theorem 9.4.4. *Suppose an n-vote profile P is compatible with $\mathcal{I}_1 > \ldots > \mathcal{I}_q$. Let $d_{max} = \max_{i \leqslant q} |D_{\mathcal{I}_q}|$. Let $t_{max}(n)$ be the longest time it takes to apply local rules on n inputs. Then, the running time of Process 1 is $O(apd_{max}(np + t_{max}(n)p + d_{max}))$, where a is the number of H-Schwartz winners.*

Usually $t_{max}(n)$ is polynomial. Therefore, the computational complexity of Process 1 mainly comes from the number of H-Schwartz winners, and the size of the largest partition d_{max}.

9.5 Summary

Sequential voting rules require the voters' preferences to extend acyclic CP-nets compatible with a common order on the attributes. We showed that this requirement is very restrictive, by proving that the number of linear orders extending an acyclic CP-net is exponentially smaller than the number of all linear orders. This means that the voting language used in sequential voting rules lacks general usability. To overcome this, in this chapter we introduced a very general methodology that allows us to aggregate preferences when voters express CP-nets that can be cyclic. There does not need to be any common structure among the submitted CP-nets. We studied whether properties of the local rules transfer to the global rule, and vice versa. We also addressed how to compute the winning alternatives.

10

A Maximum-Likelihood Approach

In voting, the joint decision is made based on the agents' preferences. Therefore, in some sense, this means that the agents' preferences are the "causes" of the joint decision. However, there is a different (and almost reversed) point of view: there is a "correct" joint decision, but the agents may have different perceptions (estimates) of what this correct decision is. Thus, the agents' preferences can be viewed as noisy reports on the correct joint decision. Even in this framework, the agents still need to make a joint decision based on their preferences, and it makes sense to choose their best estimate of the correct decision. Given a noise model, one natural approach is to choose the maximum likelihood estimate of the correct decision. The maximum likelihood estimator is a function from profiles to alternatives (more accurately, subsets of alternatives, since there may be ties), and as such is a voting rule (more accurately, a correspondence).

This maximum likelihood approach was first studied by Condorcet (1785) for the cases of two and three alternatives. Much later, Young (1995) and Young (1988) showed that for arbitrary numbers of alternatives, the MLE rule derived from Condorcet's noise model coincides with Kemeny's rule (Kemeny, 1959). The approach was further pursued by Drissi-Bakhkhat and Truchon (2004). More recently, Conitzer and Sandholm (2005a) studied whether and how common voting correspondences can be represented as maximum likelihood estimators. Truchon (2008) studied a different way of viewing Borda as an MLE. We studied the relationship between MLEs and *ranking scoring rules* (Conitzer et al., 2009b). Conitzer (2011) took an MLE approach towards voting in social networks. We studied an MLE approach towards voting with partial orders Xia and Conitzer (2011b). The related notion of *distance rationalizability* has also received attention in the computational social choice community recently (Elkind et al., 2009a).

All of the above work does not assume any structure on the set of alternatives. In this chapter, we take an MLE approach to preference aggregation in multi-issue domains, when the voters' preferences are represented by (not necessarily acyclic) CP-nets. Considering the structure of CP-nets, we focus on probabilistic models that are *very weakly decomposable.* That is, given the "correct" winner, a voter's local preferences over an issue are independent from her local preferences over other issues, and as well as from her local preferences over the same issue given a different setting of (at least some of) the other issues.

After reviewing some background, we start with the general case in which the issues are not necessarily binary. The goal here is to investigate when issue-by-issue or sequential voting rules can be modeled as maximum likelihood estimators. When the input profile is separable, we completely characterize the set of all voting correspondences that can be modeled as an MLE for a noise model satisfying a weak decomposability (respectively, strong decomposability) property. Then, when the input profile of CP-nets is compatible with a common order over issues, we prove that no sequential voting rule satisfying unanimity can be represented by an MLE, provided the noise model satisfies very weak decomposability. We show that this impossibility result no longer holds if the number of voters is bounded above by a constant.

Then, we move to the special case in which each issue has only two possible values. For such domains, we introduce *distance-based noise models*, in which the local distribution over any issue i under some setting of the other issues depends only on the Hamming distance from this setting to the restriction of the "correct" winner to the issues other than i. We characterize distance-based noise models axiomatically. Then we focus on *distance-based threshold noise models* in which there is a threshold such that if the distance is smaller than the threshold, then a fixed nonuniform local distribution is used, whereas if the distance is at least as large as the threshold, then a uniform local distribution is used. We show that when the threshold is one, it is NP-hard to compute the winner, but that when it is equal to the number of issues, the winner can be computed in polynomial time.

10.1 Maximum-Likelihood Approach to Voting in Unstructured Domains

In the maximum likelihood approach to voting, it is assumed that there is a correct winner $d \in \mathcal{C}$, and each vote V is drawn conditionally independently given d, according to a conditional probability distribution $\pi(V|d)$. The independence structure of the noise model is illustrated in Figure 10.1. The use of this independence structure is standard. Moreover, if conditional independence among votes is not required, then any voting rule can be represented by an MLE for some noise model (Conitzer and Sandholm, 2005a), which trivializes the question.

Under this independence assumption, the probability of a profile $P = (V_1, \ldots, V_n)$ given the correct winner d is $\pi(P|d) = \prod_{i=1}^{n} \pi(V_i|d)$. Then, the maximum likelihood estimate of the correct winner is $MLE_\pi(P) = \arg\max_{d \in \mathcal{C}} \pi(P|d)$.

MLE_π is a voting correspondence, as there may be several alternatives d that maximize $\pi(P|d)$. Of course we can turn it into a voting rule by using a tie-breaking

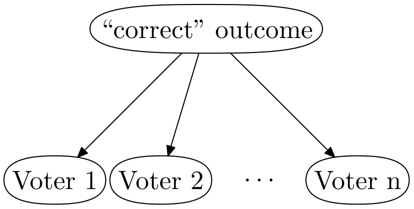


FIGURE 10.1: The noise model.

mechanism, but for most part of this chapter, we will study the properties of MLE correspondences. Another model that has been studied assumes that there is a correct *ranking* of the alternatives. Here, the model is defined similarly: given the correct linear order V^*, each vote V is drawn conditionally independently according to $\pi(V|V^*)$. The maximum likelihood estimate is defined as follows.

$$MLE_\pi(P) = \arg\max_{V^* \in L(\mathcal{C})} \prod_{V \in P} \pi(V|V^*)$$

In this chapter, we require that all such conditional probabilities to be positive for technical reasons.

Definition 10.1.1. *(Conitzer and Sandholm, 2005a). A voting rule (correspondence) r is a* maximum likelihood estimator for winners under i.i.d. votes (MLEWIV) *if there exists a noise model π such that for any profile P, we have that $MLE_\pi(P) = r(P)$.*

Conitzer and Sandholm (2005a) studied which common voting rules/correspondences are MLEWIVs.

10.2 Multi-Issue Domain Noise Models

In this section, we extend the maximum-likelihood estimation approach to multi-issue domains (where $\mathcal{X} = D_1 \times \ldots \times D_p$). For now, we consider the case where there is a correct winner, $\vec{d} \in \mathcal{X}$. We let the voting language to be the set of all (possibly cyclic) CP-nets, that is, votes are given by CP-nets and are conditionally independent, given $\vec{d}$. Let CPnet($\mathcal{X}$) denote the set of all (possibly cyclic) CP-nets over $\mathcal{X}$. The probability of drawing CP-net $\mathcal{N}$ given that the correct winner is $\vec{d}$ is $\pi(\mathcal{N}|\vec{d})$, where π is some noise model. We note that π is a conditional probability distribution over all CP-nets (in contrast to all linear orders in previous studies). Given this noise model, for any profile of CP-nets $P_{CP} = (\mathcal{N}_1, \ldots, \mathcal{N}_n)$, the maximum likelihood estimate of the correct winner is

$$MLE_\pi(P) = \arg\max_{\vec{d} \in \mathcal{X}} \prod_{j=1}^{n} \pi(\mathcal{N}_j|\vec{d})$$

Again, MLE_π is a voting correspondence.

Even if for all i, $|D_i| = 2$, the number of CP-nets (including cyclic ones) is $2^{p \cdot 2^{p-1}}$ (2 options for each entry of each CPT, and the CPT of any issue i has 2^{p-1} entries, one for each setting of the issues other than i). Hence, to specify a probability distribution over CP-nets, we will assume some structure in this distribution so that it can be compactly represented. Throughout the chapter, we will assume that the local preferences for individual issues (given the setting of the other issues) are drawn conditionally independently, both across issues and across settings of the other issues, given the correct winner. More precisely:

Definition 10.2.1. *A noise model is* very weakly decomposable *if for every $\vec{d} \in \mathcal{X}$, every $i \leqslant p$, and every $\vec{a}_{-i} \in D_{-i}$, there is a probability distribution $\pi_{\vec{d}}^{\vec{a}_{-i}}$ over $L(D_i)$, so that for every $\vec{d} \in \mathcal{X}$ and every $\mathcal{N} \in CPnet(\mathcal{X})$, $\pi(\mathcal{N}|\vec{d}) = \prod_{i \leqslant p, \vec{a}_{-i} \in D_{-i}} \pi_{\vec{d}}^{\vec{a}_{-i}}(\mathcal{N}|_{X_i:\vec{a}_{-i}})$*

For instance, if $D_i = \{0_i, 1_i, 2_i\}$, $\pi_{\vec{d}}^{\vec{a}_{-i}}(0_i \succ 2_i \succ 1_i)$ is the probability that the CP-net of a given voter contains $\vec{a}_{-i} : 0_i \succ 2_i \succ 1_i$, given that the correct winner is $\vec{d}$. Then, the probability of CP-net $\mathcal{N}$ is the product of the probabilities of all its local preferences $\mathcal{N}|_{X_i:\vec{a}_{-i}}$ over specific X_i given specific $\vec{a}_{-i}$ (which contains the setting for X_i's parents as a sub-vector), when the correct winner is $\vec{d}$. (We will introduce stronger decomposability notions soon.)

Assuming very weak decomposability is reasonable in the sense that a voter's preferences for one issue are not directly linked to her *preferences* for another issue. We note that this is completely different from saying that the voter's preferences for an issue do not depend on the *values* of the other issues. Indeed, the voter's preferences for an issue can, at least in principle, change drastically depending on the values of the other issues. For instance, in Example 8.2.2, the event "the voter prefers white to pink to red wine when the main course is fish" is probabilistically independent (conditional on the correct outcome) of the event "the voter prefers beef to salad to fish when the wine is red."

However, we do not want to argue that such a distribution always generates realistic preferences. In fact, with some probability, such a distribution generates cyclic preferences. This is not a problem, in the sense that the purpose of the maximum likelihood approach is to find a natural voting rule that maps profiles to outcomes. The fact that this rule is also defined for cyclic preferences does not hinder its application to acyclic preferences. Similarly, Condorcet's original noise model for the single-issue setting also generates cyclic preferences with some probability, but this does not prevent us from applying the corresponding (Kemeny) rule (Kemeny, 1959) to acyclic preferences.

Even assuming very weak decomposability, we still need to define exponentially many probabilities. We will now introduce some successive strengthenings of the decomposability notion. First, we introduce *weak decomposability*, which removes the dependence of an issue's local distribution on the settings of the other issues *in the correct winner.*

Definition 10.2.2. *A very weakly decomposable noise model π is* weakly decomposable *if for any $i \leqslant p$, any $\vec{d}_1, \vec{d}_2 \in \mathcal{X}$ such that $\vec{d}_1|_{X_i} = \vec{d}_2|_{X_i}$, we must have that for any $\vec{a}_{-i} \in D_{-i}$, $\pi_{\vec{d}_1}^{\vec{a}_{-i}} = \pi_{\vec{d}_2}^{\vec{a}_{-i}}$. Here $\vec{d}_1|_{X_i}$ is the X_i-component of $\vec{d}_1$.*

Next, we introduce an even stronger notion, namely *strong decomposability*, which removes all dependence of an issue's distribution on the settings of the other issues. That is, the local distribution only depends on the value of that issue in the correct winner.

Definition 10.2.3. *A very weakly decomposable noise model π is* strongly decomposable *if it is weakly decomposable, and for any $i \leqslant p$, any $\vec{a}_{-i}, \vec{b}_{-i} \in D_{-i}$, any $\vec{d} \in \mathcal{X}$, we must have that $\pi_{\vec{d}}^{\vec{a}_{-i}} = \pi_{\vec{d}}^{\vec{b}_{-i}}$.*

10.3 Characterizations of MLE correspondences

It seems that the MLE approaches are quite different from the voting rules that have previously been studied in the context of multi-issue domains, such as issue-by-issue voting and sequential voting. This may imply that the maximum likelihood approach can generate sensible new rules for multi-issue domains. Nevertheless, we may wonder whether previously studied rules also fit under the MLE framework.

In this section, we study whether or not issue-by-issue and sequential voting correspondences can be modeled as the MLEs for very weakly decomposable noise models. We note that even though MLEs for very weakly decomposable noise models are defined over profiles of CP-nets, they can be easily extended to deal with profiles of linear orders in the following way. For each linear order V_j in the input profile P, let $\mathcal{N}_j$ denote the CP-net (possibly cyclic) that V_j extends. Then, we apply the MLE rule to select winner(s) from $(\mathcal{N}_1, \ldots, \mathcal{N}_n)$. We recall that voting rules (which always output a unique winner) are a special case of voting correspondences. Therefore, our results easily extend to the case of voting rules. First, we restrict the domain to separable profiles, and characterize the set of all correspondences that can be modeled as the MLEs for strongly/weakly decomposable noise models.

Theorem 10.3.1. *Over the domain of separable profiles, a voting correspondence* r^c *can be modeled as the MLE for a strongly decomposable noise model if and only if* r^c *is an issue-by-issue voting correspondence composed of MLEWIVs.*

Proof of Theorem 10.3.1: First we prove the "if" part. Let r^c be an issue-by-issue voting correspondence that is composed of $r_1^c, \ldots, r_p^c$, in which for any $i \leqslant p$, r_i^c is an MLEWIV over D_i of the noise model $Pr(V^i|d_i)$, where $V^i \in L(D_i)$ and $d_i \in D_i$. Let π be a noise model over $\mathcal{X}$ defined as follows: for any $i \leqslant p$, any $\vec{d} \in \mathcal{X}$, any $\vec{a}_{-i} \in D_{-i}$ and any $V^i \in L(D_i)$, we have that $\pi_{\vec{d}}^{\vec{a}}(V_i) = Pr(V^i|d_i)$. We next prove that for any separable profile P, we must have that $MLE_\pi(P) = r^c(P)$.

$$\begin{aligned} MLE_\pi(P) &= \arg\max_{\vec{d}} \prod_{i \leqslant p, \vec{a}_{-i} \in D_{-i}} \prod_{j=1}^{n} \pi_{\vec{d}}^{\vec{a}_{-i}}(V_j) \\ &= \arg\max_{\vec{d}} \prod_{i \leqslant p} \prod_{j=1}^{n} Pr((V_i|_{X_i})|d_i)^{|D_{-i}|} \end{aligned}$$

Therefore, $\vec{b} \in MLE_\pi(P)$ if and only if for any $i \leqslant p$, we have

$$b_i \in \arg\max_{d_i} \prod_{j=1}^{n} Pr((V_i|_{X_i})|b_i)$$

We note that for any $\vec{d'} \in r(P)$, we must have that $d'_i = \arg\max_{d_i} \prod_{j=1}^{n} Pr((V_i|_{X_i})|d_i)$. Therefore, $\vec{d'} \in MLE_\pi(P)$.

Next, we prove the "only if" part. For any MLE_π where π is strongly decomposable, we define an issue-by-issue voting rule as follows: for any $i \leqslant p$, let r_i^c be the MLEWIV that corresponds to the noise model in which for any $d_i \in D_i$, we have that $Pr(V^i|d_i) = \pi_{\vec{d}}^{\vec{a}_{-i}}(V^i)$. Similar to the proof for the "if" part, we have that r^c and MLE_π are equivalent over the domain of separable profiles. □

A *candidate scoring correspondence* c is a correspondence defined by a scoring function $s : L(\mathcal{X}) \times \mathcal{X} \to \mathbb{R}$ in the following way: for any profile P, $c(P) = \arg\max_{d \in \mathcal{X}} \sum_{V \in P} s(V, d)$.

Theorem 10.3.2. *Over the domain of separable profiles, a voting correspondence r^c can be modeled as the MLE for a weakly decomposable noise model if and only if r^c is an issue-by-issue voting correspondence composed of candidate scoring correspondences.*

Proof of Theorem 10.3.2: First we prove the "if" part. Let r^c be an issue-by-issue voting correspondence in which the issue-wise correspondence over D_i is $r^c_{s_i}$, which has scoring function s_i. Let $\pi^{\vec{a}_{-i}}_{d_i}$ denote $\pi^{\vec{a}_{-i}}_{\vec{d}}$, where the ith component of $\vec{d}$ is d_i. Because r is strongly decomposable, $\pi^{\vec{a}_{-i}}_{d_i}$ is well-defined. For any $i \leqslant p$, we claim that there exists a set of probability distributions $\pi^{\vec{a}_{-i}}_{\vec{d}}$, $\vec{d} \in \mathcal{X}, \vec{a}_{-i} \in D_{-i}$ over $L(D_i)$ such that for any $d_i \in D_i$, $d_i \in \arg\max_{b_i \in D_i} \prod_{j=1}^{n} \prod_{\vec{a}_{-i} \in D_{-i}} \pi^{\vec{a}_{-i}}_{b_i}(V_j|_{X_i})$ if and only if $d_i \in r^c_{s_i}(P|_{X_i})$.

We note that for any scoring function s and any constant t, the ranking scoring rule that corresponds to s is equivalent to the ranking scoring rule that corresponds to $s + t$. Therefore, without loss of generality we let $s_i(V^i, d_i) < 0$ for any $i \leqslant p$, any $V^i \in L(D_i)$, and any $d_i \in D_i$. Let $K_i = |D_i|$, $L(D_i) = \{l_1, \ldots, l_{K_i!}\}$.

Claim 10.3.1. *There exist $k_i, t_i \in \mathbb{R}$ with $k_i > 0$, such that for any $V^i \in L(D_i)$ and any $d_i \in D_i$, we have that $\ln(\prod_{\vec{d}_{-i} \in D_{-i}} \pi^{\vec{d}_{-i}}_{d_i}(V^i)) = k_i s_i(V^i, d_i) + t_i$.*

Proof of Claim 10.3.1: We let k_i be a real number such that for any $d_i \in D_i$, we have that $\sum_{j=1}^{K_i!}(\exp(s_i(l_j, d_i)))^{k_i} < 1$; let $\hat{p}^j_{d_i} = \exp(s_i(l_j, d_i))$. For any $d_i \in D_i$, any

$1 \leqslant \alpha < \dfrac{K_i!}{K_i! - 1}$, we let

$$f_{d_i}(\alpha) = \ln((1 - \sum_{j=1}^{K_i!-1} \frac{\hat{p}^j_{d_i}}{\alpha})(1 - (K_i - 1)\frac{\alpha}{K_i!}))$$

Because $\sum_{j=1}^{K_i!} \hat{p}^j_{d_i} < 1$, we have that $\ln(1 - \sum_{j=1}^{K_i!-1} \hat{p}^j_{d_i}) > \ln \hat{p}^{K_i!}_{d_i} = k_i s_i(l_{K_i!}, d_i)$. Therefore, $f_{d_i}(1) \geqslant k_i s_i(l_{K_i!}, d_i) - \ln(K_i!)$. We note that $\lim_{\alpha \to \frac{K_i!}{K_i!-1}} f_{d_i}(\alpha) = -\infty$. It follows that there exists $1 \leqslant \alpha_{d_i} \leqslant \dfrac{K_i!}{K_i! - 1}$ such that $f_{d_i}(\alpha_{d_i}) = k_i s_i(l_{K_i!}, d_i) - \ln(K_i!)$.

For any $i \leqslant p$, any $d_i \in D_i$, we let $\vec{a}'_{-i}, \vec{a}^*_{-i} \in D_{-i}$ such that $\vec{a}'_{-i} \neq \vec{\vec{a}}^*_{-i}$. We define $\pi^{\vec{d}_{-i}}_{d_i}$ as follows.

- for any $j \leqslant K_i! - 1$, $\pi^{\vec{a}'_{-i}}_{d_i}(l_j) = \dfrac{1}{\alpha_{d_i}}(\exp(s_i(l_j, d_i)))^{k_i}$, $\pi^{\vec{a}^*_{-i}}_{d_i}(l_j) = \dfrac{\alpha_{d_i}}{K_i!}$.
- for any $j \leqslant K_i!$, any $\vec{d}_{-i} \in D_{-i}$ such that $\vec{d}_{-i} \neq \vec{a}'_{-i}$ and $\vec{d}_{-i} \neq \vec{a}^*_{-i}$, we have that $\pi^{\vec{d}_{-i}}_{d_i}(l_j) = \frac{1}{K_i!}$.

For any $\vec{d_i} \in D_i$ and any $j \leqslant K_i! - 1$, we have that

$$\begin{aligned}
&\ln(\prod_{\vec{d}_{-i} \in D_{-i}} \pi^{\vec{d}_{-i}}_{d_i}(l_j)) \\
=&\ln(\pi^{\vec{a}'_{-i}}_{d_i}(l_j) \cdot \pi^{\vec{a}^*_{-i}}_{d_i}(l_j)) + (|D_{-i}| - 2)\ln(\frac{1}{K_i!}) \\
=&\ln(\frac{1}{\alpha_{d_i}}(\exp(s_i(l_j, d_i)))^{k_i} \cdot \frac{\alpha_{d_i}}{K_i!}) - (|D_{-i}| - 2)\ln(K_i!) \\
=&k_i s_i(l_j, d_i) - (|D_{-i}| - 1)\ln(K_i!)
\end{aligned}$$

For $j = K_i!$, we have the following calculation.

$$\begin{aligned}
&\ln(\prod_{\vec{d}_{-i}\in D_{-i}} \pi_{d_i}^{\vec{d}_{-i}}(l_{K_i!}))\\
=&\ln(\pi_{d_i}^{\vec{a}'_{-i}}(l_{K_i!}) \cdot \pi_{d_i}^{\vec{a}^*_{-i}}(l_{K_i!})) + (|D_{-i}| - 2)\ln(\frac{1}{K_i!})\\
=&f_{d_i}(\alpha_i) - (|D_{-i}| - 2)\ln(K_i!)\\
=&k_i s_i(l_{K_i!}, d_i) - (|D_{-i}| - 1)\ln(K_i!)
\end{aligned}$$

Therefore, let $t_i = -(|D_{-i}| - 1)\ln(K_i!)$. It follows that for any $V^i \in L(D_i)$, and any $d_i \in D_i$, we must have that $\ln(\prod_{\vec{d}_{-i}\in D_{-i}} \pi_{d_i}^{\vec{d}_{-i}}(V^i)) = k_i s_i(V^i, d_i) + t_i$. □

Next, we show that for any separable profile P, $r^c(P) = MLE_\pi(P)$. Similar to in the proof of Theorem 10.3.1, it suffices to prove that for any $i \leqslant p$, $\arg\max_{d_i\in D_i} \prod_{j\leqslant n} \prod_{\vec{d}_{-i}\in D_{-i}} \pi_{d_i}^{\vec{d}_{-i}}(V_j|_{X_i}) = r^c_{s_i}(P|_{X_i})$.

$$\begin{aligned}
&\arg\max_{d_i\in D_i} \prod_{j\leqslant n} \prod_{\vec{d}_{-i}\in D_{-i}} \pi_{d_i}^{\vec{d}_{-i}}(V_j|_{X_i})\\
=&\arg\max_{d_i\in D_i} \ln(\prod_{j\leqslant n} \prod_{\vec{d}_{-i}\in D_{-i}} \pi_{d_i}^{\vec{d}_{-i}}(V_j|_{X_i}))\\
=&\arg\max_{d_i\in D_i} \sum_{j\leqslant n} \sum_{\vec{d}_{-i}\in D_{-i}} \ln(\pi_{d_i}^{\vec{d}_{-i}}(V_j|_{X_i}))\\
=&\arg\max_{d_i\in D_i} \sum_{j\leqslant n} \sum_{\vec{d}_{-i}\in D_{-i}} (k_i s_i(V_j|_{X_i}, d_i) + t_i)\\
=&\arg\max_{d_i\in D_i} \sum_{j\leqslant n} \sum_{\vec{d}_{-i}\in D_{-i}} s_i(V_j|_{X_i}, d_i)\\
=&r^c_{s_i}(P|_{X_i})
\end{aligned}$$

Next, we prove the “only if” part. Let π be a weakly decomposable noise model. For any $i \leqslant p$, any $d_i \in D_i$, and any $V^i \in L(D_i)$, we let $s_i(V^i, d_i) = \ln \prod_{\vec{a}_{-i}\in D_{-i}} \pi_{d_i}^{\vec{a}_{-i}}(V^i)$. Then, we have that d_i maximizes $s_i(P|_{X_i}, d_i)$ if and only if d_i

maximizes $\prod_{\mathcal{N}\in P}\prod_{\vec{a}_{-i}\in D_{-i}}\pi_{d_i}^{\vec{a}_{-i}}(\mathcal{N}|_{X_i})$, which means that $r^c(P) = MLE_\pi(P)$.

(End of the proof of Theorem 10.3.2). □

However, for sequential voting correspondences, we have the following negative result. A voting correspondence r^c satisfies *unanimity* if for any profile P in which each vote ranks an alternative $\vec{d}$ first, we have $r(P) = \{\vec{d}\}$. In the remainder of this section, w.l.o.g. we let $\mathcal{O} = X_1 > \cdots > X_p$.

Theorem 10.3.3. *Let $Seq(r_1^c,\ldots,r_p^c)$ be a sequential voting correspondence that satisfies unanimity. Over the domain of $\mathcal{O}$-legal profiles, there is no very weakly decomposable noise model such that $Seq(r_1^c,\ldots,r_p^c)$ is the MLE.*

This theorem tells us that even assuming the weakest conditional independence of the noise model, the voting correspondence defined by the MLE of that noise model is different from any sequential voting correspondence satisfying unanimity. This suggests that the MLE approach gives us new voting rules/correspondences.

Proof of Theorem 10.3.3: For the sake of contradiction, we let $Seq(r_1^c,\ldots,r_p^c)$ be a sequential voting correspondence and MLE_π be an MLE model equivalent to it. A voting correspondence c satisfies *consistency*, if for any profiles P_1, P_2, if $r^c(P_1) = r^c(P_2)$, then $r^c(P_1 \cup P_2) = r^c(P_1)$; c satisfies anonymity, if it is indifferent with the name of the voters. Because MLE_π satisfies consistency and anonymity, we have the following claim.

Claim 10.3.2. *For any $i \leqslant p$, r_i^c satisfies consistency, anonymity (see* Lang and Xia (2009)*) and unanimity.*

For any $\vec{d}\in\mathcal{X}$, any $\mathcal{O}$-legal CP-net $\mathcal{N}$, we let

$$\pi_{\vec{d}}^{X_1}(\mathcal{N}) = \prod_{\vec{a}_{-1}\in D_{-1}}\pi_{\vec{d}}^{\vec{a}_{-1}}(\mathcal{N}|_{X_1})$$

$$\pi_{\vec{d}}^{X_{-1}}(\mathcal{N}) = \prod_{2\leqslant i\leqslant p,\vec{a}_{-i}\in D_{-i}}\pi_{\vec{d}}^{\vec{a}_{-i}}(\mathcal{N}|_{X_i:a_1\ldots a_{i-1}})$$

Let $\mathcal{N}_1, \mathcal{N}_2$ be CP-nets. We note that if $\mathcal{N}_1|_{X_1} = \mathcal{N}_2|_{X_1}$, then $\pi_{\vec{d}}^{X_1}(\mathcal{N}_1) = \pi_{\vec{d}}^{X_1}(\mathcal{N}_2)$; if for any $d_1 \in D_1$, $\mathcal{N}_1|_{X_{-1}:d_1} = \mathcal{N}_2|_{X_{-1}:d_1}$, then we must have that $\pi_{\vec{d}}^{X_{-1}}(\mathcal{N}_1) = \pi_{\vec{d}}^{X_{-1}}(\mathcal{N}_2)$, where $\mathcal{N}_1|_{X_{-1}:d_1}$ is the sub-CP-net of $\mathcal{N}_1$ given $X_1 = d_1$. For any $\mathcal{O}$-legal vote V that extends a CP-net $\mathcal{N}$, we write $\pi_{\vec{d}}^{X_1}(V) = \pi_{\vec{d}}^{X_1}(\mathcal{N})$ and $\pi_{\vec{d}}^{X_{-1}}(V) = \pi_{\vec{d}}^{X_{-1}}(\mathcal{N})$; for any $\mathcal{O}$-legal profile P, we write $\pi_{\vec{d}}^{X_1}(P) = \prod_{V \in P} \pi_{\vec{d}}^{X_1}(V)$ and $\pi_{\vec{d}}^{X_{-1}}(P) = \prod_{V \in P} \pi_{\vec{d}}^{X_{-1}}(V)$. It follows that for any $\mathcal{O}$-legal profile P, we have that

$$MLE_\pi(P) = \arg\max_{\vec{d} \in \mathcal{X}} [\pi_{\vec{d}}^{X_1}(P) \cdot \pi_{\vec{d}}^{X_{-1}}(P)]$$

For any linear order V, let $top(V) = \text{Alt}(V, 1)$. That is, $top(V)$ is the alternative that is ranked in the top position of V. For any $V_1^1, V_2^1 \in L(D_1)$ with $top(V_1^1) \neq top(V_2^1)$, and any $n \in \mathbb{N}$, we let $P_{1,n}^1$ be the profile that is composed of n copies of V_1^1; let $P_{2,n}^1$ be the profile that is composed of n copies of V_2^1. Because r_1^c satisfies unanimity, we must have that $r_1^c(P_{1,n}^1) = \{top(V_1^1)\}$ and $r_1^c(P_{2,n}^1) = \{top(V_2^1)\}$. For any $j \leqslant n$, we let $Q_{j,n}$ be the profile in which the preferences of the first j voters are V_1^1, and the preferences of the remaining $n - j$ voters are V_2^1. We have that $Q_{1,n} = P_{1,n}^1$ and $Q_{n,n} = P_{2,n}^1$. Therefore, there exists $j \leqslant n - 1$ and $b_1 \in D_1$ with $b_1 \neq top(V_1^1)$, such that $top(V_1^1) \in r_1^c(Q_{j,n})$ and $b_1 \in r_1^c(Q_{j+1,n})$. For any $n \in \mathbb{N}$, we let C_n denote the set of pairs (a_1, b_1) such that

- $a_1, b_1 \in D_1$, $a_1 \neq b_1$.
- There exists two profiles W_1^1, W_2^1 over D_1 such that $a_1 \in r_1^c(W_1^1)$, $b_1 \in r_1^c(W_2^1)$, and W_1^1 differs from W_2^1 only on one vote.

That is, C_n is composed of the pairs (a_1, b_1) such that there exists a profile Q over D_1 that consists of n votes, $a_1 \in r_1^c(Q)$, and by changing one vote of Q, there is another alternative b_1 who is one of the winners. We note that for any $n \in \mathbb{N}$, $(a_1, b_1) \in C_n$ if and only if $(b_1, a_1) \in C_n$. It follows that for any $n \in \mathbb{N}$, $C_n \neq \varnothing$. Because $|D_1| < \infty$, there exists $(a_1, b_1) \in (D_1)^2$ such that for any $k \in \mathbb{N}$, there exists $n \geqslant k$ such that $(a_1, b_1) \in C_n$.

Claim 10.3.3. *For any $\vec{a}_{-1}, \vec{b}_{-1} \in D_{-1}$, and any pair of CP-nets $\mathcal{N}', \mathcal{N}^*$, we must have that $\frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}')}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}')} = \frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}^*)}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}^*)}$, where $\vec{a} = (a_1, \vec{a}_{-1})$, $\vec{b} = (b_1, \vec{b}_{-1})$.*

Proof of Claim 10.3.3: Suppose for the sake of contradiction there exist $\vec{a}_{-1}, \vec{b}_{-1}$, and $\mathcal{N}', \mathcal{N}^*$ so that $\frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}')}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}')} \neq \frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}^*)}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}^*)}$. Without loss of generality we let $\frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}')}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}')} > \frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}^*)}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}^*)}$. We next claim that there exits a natural number k such that for any $i \leqslant p$ and any profile P^i composed of k votes, if at least $k-1$ votes in P^i rank the same alternative d_i in the top position, then $r_i^c(P^i) = \{d_i\}$.

Claim 10.3.4. *There exists $k \in \mathbb{N}$ such that for any $i \leqslant p$, any $d_i \in D_i$, and any profile $P^i = (V_1^i, \ldots, V_k^i)$ with $d_i = top(V_1^i) = \ldots = top(V_{k-1}^i)$, we have that $r_i^c(P) = \{d_i\}$.*

Proof of Claim 10.3.4: Let $U = \max_{\vec{d_1}, \vec{d_2}, \mathcal{N}} \frac{Pr(\mathcal{N}|\vec{d_1})}{Pr(\mathcal{N}|\vec{d_2})}$. Let $u = \min_{\vec{d_1} \neq \vec{d_2}, \mathcal{N}: top(\mathcal{N}) = \vec{d_1}} \frac{Pr(\mathcal{N}|\vec{d_1})}{Pr(\mathcal{N}|\vec{d_2})}$. Because $MLE_\pi(\mathcal{N})$ satisfies unanimity, for any $\vec{d_1}$ and $\mathcal{N}$ such that $top(\mathcal{N}) = \vec{d_1}$, we must have that $MLE_\pi(\mathcal{N}) = \{\vec{d_1}\}$, which means that $u > 1$. Let k be a natural number such that $u^{k-1} > U$. We arbitrarily choose $\vec{d}_{-i} \in D_{-i}$, and let $\vec{d} = (d_i, \vec{d}_{-i})$. We define k CP-nets $\mathcal{N}_1, \ldots, \mathcal{N}_k$ as follows.

- For any $j \leqslant k$, $top(\mathcal{N}_j) = (\vec{d}_{-i}, top(V^i))$.
- For any $j \leqslant k$, $\mathcal{N}_j|_{X_i:d_1,\ldots,d_{i-1}} = V^i$.
- Other conditional preferences are defined arbitrarily.

Because $Seq(r^c_1, \ldots, r^c_p)$ satisfies unanimity, we have that $Seq(r^c_1, \ldots, r^c_p)(\mathcal{N}_1, \ldots, \mathcal{N}_{k-1}) = \{\vec{d}\}$. Therefore, for any $\vec{d'} \in \mathcal{X}$ and any CP-net $\mathcal{N}$, we have the following calculation:

$$\begin{aligned}\frac{Pr((\mathcal{N}_1, \ldots, \mathcal{N}_k)|\vec{d})}{Pr((\mathcal{N}_1, \ldots, \mathcal{N}_k)|\vec{d'})} =& \frac{\prod_{j=1}^{k-1} Pr(\mathcal{N}_j|\vec{d})}{\prod_{j=1}^{k-1} Pr(\mathcal{N}_j|\vec{d'})} \cdot \frac{Pr(\mathcal{N}_k|\vec{d})}{Pr(\mathcal{N}_k|\vec{d'})} \\ \geqslant& u^{(k-1)} \frac{1}{U} > 1\end{aligned}$$

Therefore $r^c_i(V^1, \ldots, V^k) = \{d_i\}$.

(End of proof of Claim 10.3.4.) □

Let $\mathcal{N}_{\vec{a}}$ be a CP-net such that $top(\mathcal{N}_{\vec{a}}) = \vec{a}$ and $top(\mathcal{N}|_{X_{-1}:b_1}) = \vec{b}_{-1}$. That is, $\mathcal{N}_{\vec{a}}$ is a CP-net in which $\vec{a}$ is ranked in the top position, and given $X_1 = b_1$, $\vec{b}_{-1}$ is ranked in the top position. Next, we show that for any CP-net $\mathcal{N}$, $\dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N})} = \dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N}_{\vec{a}})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N}_{\vec{a}})}$. Suppose for the sake of contradiction, there exists $\mathcal{N}$ such that $\dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N})} \neq \dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N}_{\vec{a}})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N}_{\vec{a}})}$. We next show contradiction in the case $\dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N})} > \dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N}_{\vec{a}})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N}_{\vec{a}})}$. Let $U_{X_1} = \max\limits_{\vec{d_1}, \vec{d_2}, \mathcal{N}} \dfrac{\pi^{X_1}_{\vec{d_1}}(\mathcal{N})}{\pi^{X_1}_{\vec{d_2}}(\mathcal{N})}$. Let K be a natural number such that $(\dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N})} / \dfrac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N}_{\vec{a}})}{\pi^{X_{-1}}_{\vec{b}}(\mathcal{N}_{\vec{a}})})^K > U^2_{X_1}$. Let $n \in \mathbb{N}$ be such that $n > kK$ and $(a_1, b_1) \in C_n$. It follows that there exist $(V^1_1, \ldots, V^1_n)$ and W^1_1 such that $a_1 \in r^c_1(V^1_1, \ldots, V^1_n)$ and $b_1 \in r^c_1(W^1_1, V^1_2, \ldots, V^1_n)$. We define $2n+1$ CP-nets $\mathcal{N}'_1, \mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_n, \hat{\mathcal{N}}_1, \hat{\mathcal{N}}_2, \ldots, \hat{\mathcal{N}}_n$ as follows.

- For any $j \leqslant n$, $\mathcal{N}_j|_{X_1} = \hat{\mathcal{N}}_j|_{X_1} = V^1_j$; $\mathcal{N}'_1|_{X_1} = W^1_1$.

- For any $j_1 \leqslant K$, $1 \leqslant j_2 \leqslant k-1$, and any $d_1 \in D_1$, $\mathcal{N}_{(j_1-1)k+j_2}|_{X_{-1}:d_1} = \mathcal{N}_{\vec{a}}|_{X_{-1}:d_1}$ and $\mathcal{N}_{j_1k}|_{X_{-1}:d_1} = \mathcal{N}|_{X_{-1}:d_1}$; for any $j \leqslant n$ and any $d_1 \in D_1$, $\mathcal{N}_j|_{X_{-1}:d_1} = \mathcal{N}_{\vec{a}}|_{X_{-1}:d_1}$.

- For any $kK+1 \leqslant j \leqslant n$, $\mathcal{N}_j = \hat{\mathcal{N}}_j = \mathcal{N}_{\vec{a}}$.
- For any $d_1 \in D_1$, $\mathcal{N}'_1|_{X_{-1}:d_1} = \mathcal{N}_{\vec{a}}|_{X_{-1}:d_1}$.

For any $j \leqslant n$, we let V_j ($\hat{V}_j$) be an arbitrary linear order that extends $\mathcal{N}_j$ ($\hat{\mathcal{N}}_j$); let V'_1 be an arbitrary linear order that extends $\mathcal{N}'_1$; let $P = (V_1, \ldots, V_n)$, $P' = (V'_1, V_2, \ldots, V_n)$, $\hat{P} = (\hat{V}_1, \ldots, \hat{V}_n)$, $\hat{P}' = (\hat{V}'_1, \hat{V}_2, \ldots, \hat{V}_n)$. We make the following observations.

- $a_1 \in r^c_1(P|_{X_1}), a_1 \in r^c_1(\hat{P}|_{X_1})$, $b_1 \in r^c_1(P'|_{X_1}), b_1 \in r^c_1(\hat{P}'|_{X_1})$.
- For any $1 \leqslant i \leqslant p-1$, $P|_{X_i:a_1\ldots a_{i-1}} = K((k-1)\mathcal{N}_{\vec{a}}|_{X_i:a_1\ldots a_{i-1}} \cup \mathcal{N}'|_{X_i:a_1\ldots a_{i-1}}) \cup (n-kK)\mathcal{N}_{\vec{a}}|_{X_i:a_1\ldots a_{i-1}}$. From Claim 10.3.4 we have that $r^c_i((k-1)\mathcal{N}_{\vec{a}}|_{X_i:a_1\ldots a_{i-1}} \cup \mathcal{N}'|_{X_i:a_1\ldots a_{i-1}}) = \{a_i\}$. Because r^c_i satisfies unanimity and consistency, and for any $i \leqslant p$, $top(\mathcal{N}_{\vec{a}}|_{X_i:a_1\ldots a_{i-1}}) = a_i$, we have that for any $i \leqslant p$, $r^c_i(P|_{X_i:a_1\ldots a_{i-1}}) = \{a_i\}$. Similarly for any $i \leqslant p$, $r^c_i(\hat{P}|_{X_i:a_1\ldots a_{i-1}}) = \{a_i\}$.
- For any $1 \leqslant i \leqslant p-1$, $P|_{X_i:b_1\ldots b_{i-1}} = K((k-1)\mathcal{N}_{\vec{a}}|_{X_i:b_1\ldots b_{i-1}} \cup \mathcal{N}'|_{X_i:b_1\ldots b_{i-1}}) \cup (n-kK)\mathcal{N}_{\vec{a}}|_{X_i:b_1\ldots b_{i-1}}$. Similarly, we have that for any $1 \leqslant i \leqslant p$, $r^c_i(P'|_{X_i:b_1\ldots b_{i-1}}) = r^c_i(\hat{P}'|_{X_i:b_1\ldots b_{i-1}}) = \{b_i\}$.

Therefore, we have that $\vec{a} \in Seq(r^c_1, \ldots, r^c_p)(P), \vec{a} \in Seq(r^c_1, \ldots, r^c_p)(\hat{P})$, and $\vec{b} \in Seq(r^c_1, \ldots, r^c_p)(P'), \vec{b} \in Seq(r^c_1, \ldots, r^c_p)(\hat{P}')$. That is, $\dfrac{Pr(P'|\vec{b})}{Pr(P'|\vec{a})} \geqslant 1$, $\dfrac{Pr(\hat{P}'|\vec{b})}{Pr(\hat{P}'|\vec{a})} \geqslant 1$. We note that P and P' differ only on the first vote. Therefore, we have the following calculation.

$$\begin{aligned}1 \leqslant&\frac{Pr(P'|\vec{b})}{Pr(P'|\vec{a})}\\ =&\frac{\pi_{\vec{b}}^{X_1}(V_1')\cdot\pi_{\vec{b}}^{X_{-1}}(V_1')\prod_{2\leqslant j\leqslant n}(\pi_{\vec{b}}^{X_1}(V_j)\cdot\pi_{\vec{b}}^{X_{-1}}(V_j))}{\pi_{\vec{a}}^{X_1}(V_1')\cdot\pi_{\vec{a}}^{X_{-1}}(V_1')\prod_{2\leqslant j\leqslant n}(\pi_{\vec{a}}^{X_1}(V_j)\cdot\pi_{\vec{a}}^{X_{-1}}(V_j))}\\ =&\frac{\pi_{\vec{b}}^{X_1}(V_1')}{\pi_{\vec{a}}^{X_1}(V_1')}\cdot\frac{\pi_{\vec{a}}^{X_1}(V_1)}{\pi_{\vec{b}}^{X_1}(V_1)}\cdot\frac{Pr(P|\vec{b})}{Pr(P|\vec{a})}\\ \leqslant&U_{X_1}^2\frac{Pr(P|\vec{b})}{Pr(P|\vec{a})}\end{aligned}$$

Therefore, $\dfrac{Pr(P|\vec{a})}{Pr(P|\vec{b})}\leqslant U_{X_1}^2$. We note that P and P' differ on K votes.

$$\begin{aligned}&(\frac{Pr(P|\vec{a})}{Pr(P|\vec{b})})/(\frac{Pr(\hat{P}|\vec{a})}{Pr(\hat{P}|\vec{b})})\\ =&(\prod_{j=1}^{K}\frac{\pi_{\vec{a}}^{X_1}(V_{jk})\cdot\pi_{\vec{a}}^{X_{-1}}(V_{jk})}{\pi_{\vec{b}}^{X_1}(V_{jk})\cdot\pi_{\vec{b}}^{X_{-1}}(V_{jk})})/(\prod_{j=1}^{K}\frac{\pi_{\vec{a}}^{X_1}(\hat{V}_{jk})\cdot\pi_{\vec{a}}^{X_{-1}}(\hat{V}_{jk})}{\pi_{\vec{b}}^{X_1}(\hat{V}_{jk})\cdot\pi_{\vec{b}}^{X_{-1}}(\hat{V}_{jk})})\\ =&(\frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N})}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N})}/\frac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}_{\vec{a}})}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}_{\vec{a}})})^K\\ >&U_{X_1}^2\end{aligned}$$

We note that $(\dfrac{Pr(\hat{P}|\vec{a})}{Pr(\hat{P}|\vec{b})})\geqslant 1$. Therefore, $\dfrac{Pr(P|\vec{a})}{Pr(P|\vec{b})}>U_{X_1}^2$, which is a contradiction.

Similarly, for the case of $\dfrac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N})}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N})}<\dfrac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}_{\vec{a}})}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}_{\vec{a}})}$ we still have a contradiction.

Hence, $\dfrac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N})}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N})}=\dfrac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}_{\vec{a}})}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}_{\vec{a}})}$ for all $\mathcal{N}$, which means that for any $\mathcal{N}'$ and $\mathcal{N}^*$, we must have that $\dfrac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}')}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}')}=\dfrac{\pi_{\vec{a}}^{X_{-1}}(\mathcal{N}^*)}{\pi_{\vec{b}}^{X_{-1}}(\mathcal{N}^*)}$.

(End of proof of Claim 10.3.3.) □

By Claim 10.3.3, for any CP-net $\mathcal{N}$, any $\vec{b}_{-1}, \vec{b}'_{-1} \in D_{-1}$, we must have that $\frac{\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N})}{\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_{\vec{a}})} = \frac{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N})}{\pi^{X_{-1}}_{\vec{a}}(\mathcal{N}_{\vec{a}})} = \frac{\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N})}{\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_{\vec{a}})}$, which means that $\frac{\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N})}{\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N})} = \frac{\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_{\vec{a}})}{\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_{\vec{a}})}$.

Let $\mathcal{N}_1$ be a CP-net such that $top(\mathcal{N}_1) = (b_1, \vec{b}'_{-1})$, $\mathcal{N}_2$ be a CP-net such that $top(\mathcal{N}_2) = (b_1, \vec{b}_{-1})$ and $\mathcal{N}_1|_{X_1} = \mathcal{N}_2|_{X_1}$. Because $Seq(r^c_1, \ldots, r^c_p)$ satisfies unanimity, we have that $\frac{Pr(\mathcal{N}_1|(b_1,\vec{b}'_{-1}))}{Pr(\mathcal{N}_1|(b_1,\vec{b}_{-1}))} > 1$ and $\frac{Pr(\mathcal{N}_2|(b_1,\vec{b}'_{-1}))}{Pr(\mathcal{N}_2|(b_1,\vec{b}_{-1}))} < 1$. However, we have the following calculation.

$$\begin{aligned}
1 <& \frac{Pr(\mathcal{N}_1|(b_1,\vec{b}'_{-1}))}{Pr(\mathcal{N}_1|(b_1,\vec{b}_{-1}))}\\
=& \frac{\pi^{X_1}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_1)\cdot\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_1)}{\pi^{X_1}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_1)\cdot\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_1)}\\
=& \frac{\pi^{X_1}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_2)\cdot\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_{\vec{a}})}{\pi^{X_1}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_2)\cdot\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_{\vec{a}})} && \text{(Because } \mathcal{N}_1|_{X_1} = \mathcal{N}_2|_{X_1}\text{)}\\
=& \frac{\pi^{X_1}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_2)\cdot\pi^{X_{-1}}_{(b_1,\vec{b}'_{-1})}(\mathcal{N}_2)}{\pi^{X_1}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_2)\cdot\pi^{X_{-1}}_{(b_1,\vec{b}_{-1})}(\mathcal{N}_2)}\\
=& \frac{Pr(\mathcal{N}_2|(b_1,\vec{b}'_{-1}))}{Pr(\mathcal{N}_2|(b_1,\vec{b}_{-1}))}\\
<& 1
\end{aligned}$$

Therefore, we have a contradiction. **(End of proof of Theorem 10.3.3.)** □

However, a connection between MLEs for very weakly decomposable noise models and sequential voting correspondences can be obtained if there is an upper bound on the number of voters. The next theorem states that for any natural number n and any sequential composition of MLEWIVs, there exists a very weakly decomposable noise model such that for any profile of no more than n $\mathcal{O}$-legal votes, the set of winners under the MLE for that noise model is always a subset of the set of winners under the sequential correspondence. That is, if the local correspondences can be justified by a noise model, then, to some extent, so can the sequential voting correspondence that uses these local rules.

Theorem 10.3.4. *For any $n \in \mathbb{N}$ and any sequential voting correspondence $Seq(r_1^c, \ldots, r_p^c)$ where for each $i \leqslant p$, r_i^c is an MLEWIV, there exists a very weakly decomposable noise model π such that for any $\mathcal{O}$-legal profile P composed of no more than n votes, we have that $MLE_\pi(P) \subseteq Seq(r_1^c, \ldots, r_p^c)(P)$.*

Proof of Theorem 10.3.4: Let r_i be the MLEWIV with the conditional probability distribution $Pr_i(V^i|d_i)$, where $V^i \in L(D_i)$, $d_i \in D_i$. For any $i \leqslant p$, we let $R_{max}^{i,n} = \max_{P_i,P_i',d_i,d_i'} \left\{ \dfrac{Pr_i(P_i|d_i)}{Pr_i(P_i'|d_i')} \right\}$, where $d_i, d_i' \in D_i$, and P_i and P_i' are profiles with the same number (but no more than n) of linear orders over D_i. We let $R_{min}^{i,n} = 1$ if r_i is the trivial correspondence that always outputs the whole domain; and $R_{min}^{i,n} = \min_{P_i,\vec{d}_i,\vec{d}_i'} \left\{ \dfrac{Pr_i(P_i|d_i)}{Pr_i(P_i|d_i')} : \dfrac{Pr_i(P_i|d_i)}{Pr_i(P_i|d_i')} > 1 \right\}$, where $d_i, d_i' \in D_i$, and P_i is a profile of no more than n linear orders over D_i. We note that for any $i \leqslant p$, any $n \in \mathbb{N}$, we have that $R_{max}^{i,n} \geqslant R_{min}^{i,n} \geqslant 1$.

For any $V^i \in L(D_i)$, any $\vec{d} \in \mathcal{X}$, and any $\vec{a}_{-i} \in D_{-1}$, we let

$$\pi_{\vec{d}}^{\vec{a}_{-i}}(V^i) = \begin{cases} Pr_i(V^i|d_i)^{k_i}/Z_i & \text{if } \vec{a}_{-i} = \vec{d}_{-i} \\ \dfrac{1}{|D_i|!} & \text{otherwise} \end{cases},$$

where $Z_i = \sum_{V^i \in L(D_i)} Pr_i(V^i|d_i)^{k_i}$ is a normalizing factor, and $1 = k_1 > k_2 > \cdots > k_p > 0$ are chosen in the following way: for any $i' < i \leqslant p$, any $V^i, W^i \in L(D_i)$, and any $d_i, d_i' \in D_i$, if $R_{min}^{i,n} > 1$, then we must have that $(R_{max}^{i,n})^{k_i} < (R_{min}^{i',n})^{k_{i'}/2^{i-i'}}$.

We next prove that for any profile P_{CP} of no more than n CP-nets, we must have that $MLE_\pi(P_{CP}) \subseteq Seq(r_1^c, \ldots, r_p^c)(P_{CP})$. For the sake of contradiction, let P_{CP} be a profile of no more than n CP-nets with $MLE_\pi(P_{CP}) \nsubseteq Seq(r_1^c, \ldots, r_p^c)(P_{CP})$. Let $\vec{d} \in$

$MLE_\pi(P_{CP})$, and i^* be the number such that there exists $\vec{d}^* \in Seq(r^c_1, \ldots, r^c_p)(P_{CP})$ such that for all $i' < i^*$, $d_{i'} = d^*_{i'}$, and $d_{i*} \notin r^c_{i*}(P_{CP}|_{X_{i*}:d_1\ldots d_{i*-1}})$. Because $r^c_{i*}(P_{CP}|_{X_{i*}:d_1\ldots d_{i*-1}}) \neq D_{i*}$, we must have that $R^{i*,n}_{min} > 1$. Because $\vec{d} \in MLE_\pi(P_{CP})$, we must have that $\dfrac{\pi(P_{CP}|\vec{d})}{\pi(P_{CP}|\vec{d}^*)} \geqslant 1$. However, we have the following calculation that leads to a contradiction.

$$\begin{aligned}
1 \leqslant & \frac{\pi(P_{CP}|\vec{d})}{\pi(P_{CP}|\vec{d}^*)} = \frac{\prod_{i=1}^p Pr_i(P_{CP}|_{X_i:d_1\ldots d_{i-1}}|d_i)}{\prod_{i=1}^p Pr_i(P_{CP}|_{X_i:d^*_1\ldots d^*_{i-1}}|d^*_i)}\\
= & \frac{\prod_{i=i*}^p Pr_i(P_{CP}|_{X_i:d_1\ldots d_{i-1}}|d_i)}{\prod_{i=i*}^p Pr_i(P_{CP}|_{X_i:d^*_1\ldots d^*_{i-1}}|d^*_i)}\\
\leqslant & \frac{1}{(R^{i*,n}_{min})^{k_{i*}}} \cdot \prod_{i=i*+1}^{p} (R^{i,n}_{max})^{k_i}\\
< & \frac{1}{(R^{i*,n}_{min})^{k_{i*}}} \cdot \prod_{i=i*+1}^{p} (R^{i*,n}_{min})^{k_{i*}/2^{i-i*}} < 1
\end{aligned}$$

Therefore, we must have that $MLE_\pi(P) \subseteq Seq(r^c_1, \ldots, r^c_p)(P)$ for all profiles P that consist of no more than n CP-nets. □

10.4 Distance-Based Models

We have shown in the previous section that the MLE approach may give us new voting rules in multi-issue domains. However, assuming very weak decomposability, there are too many (exponentially many) parameters in the noise model, which makes it very hard to implement a rule based on the MLE approach. In this section, we focus on a family of maximum likelihood estimators that are based on noise models defined over multi-binary-issue domains (domains composed of binary issues), and that need only a few parameters to be specified. We recall that a CP-net on a multi-binary-issue domain corresponds to a directed hypercube in which each edge has a direction representing the local preference. A very weakly decomposable noise model π can be represented by a collection of weighted directed hypercubes, one for each correct winner, in which the weight of each directed edge is the probability of the local preference represented by the directed edge. For any outcome $\vec{d} \in \mathcal{X}$, any issue X_i, any $\vec{e}_{-i} \in D_{-i}$, and any $d_i \neq d'_i \in D_i$, the weight on the directed edge $((\vec{e}_{-i}, d_i), (\vec{e}_{-i}, d'_i))$ of the weighted hypercube corresponding to the correct winner $\vec{d}$ is denoted by $\pi_{\vec{d}}^{\vec{e}_{-i}}(d_i \succ d'_i)$, and represents the probability that a given voter reports the preference $\vec{e}_{-i} : d_i \succ d'_i$ in her CP-net, given that the correct winner is $\vec{d}$.[1] For example, when the correct winner is $0_1 0_2 0_3$, the weight on the directed edge $(0_1 1_2 0_3, 0_1 1_2 1_3)$ is the probability $\pi_{0_1 0_2 0_3}^{0_1 1_2}(0_3 \succ 1_3)$. We now propose and study very weakly decomposable noise models in which the weight of each edge depends only on the Hamming distance between the edge and the correct winner.

For any pair of alternatives $\vec{d}, \vec{d'} \in \mathcal{X}$, the *Hamming distance* between $\vec{d}$ and $\vec{d'}$, denoted by $|\vec{d} - \vec{d'}|$, is the number of components in which $\vec{d}$ is different from $\vec{d'}$, that is, $|\vec{d} - \vec{d'}| = \#\{i \leqslant p : d_i \neq d'_i\}$. Let $e = (\vec{d_1}, \vec{d_2})$ be a pair of alternatives such that $|\vec{d_1} - \vec{d_2}| = 1$ (equivalently, an edge in the hypercube). The distance between e and an alternative $\vec{d} \in \mathcal{X}$, denoted by $|e - \vec{d}|$, is the smaller Hamming distance between $\vec{d}$ and the two ends of e, that is, $|e - \vec{d}| = \min\{|\vec{d_1} - \vec{d}|, |\vec{d_2} - \vec{d}|\}$. For example, $|0_1 1_2 0_3 - 0_1 0_2 0_3| = 1, |0_1 1_2 1_3 - 0_1 0_2 0_3| = 2$, and $|(0_1 1_2 0_3, 0_1 1_2 1_3) - 0_1 0_2 0_3| = 1$.

We next introduce *distance-based noise models* in which the probability distribution $\pi_{\vec{d}}^{\vec{a}_{-i}}$ only depends on d_i and the Hamming distance between $\vec{a}_{-i}$ and $\vec{d}_{-i}$.

Definition 10.4.1. *Let $\mathcal{X}$ be a multi-binary-issue domain. For any $\vec{q} = (q_0, \ldots, q_{p-1})$ such that $1 > q_0, \ldots, q_{p-1} > 0$, a* distance-based (noise) model *$\pi_{\vec{q}}$ is a very weakly decomposable noise model such that for any $\vec{d} \in \mathcal{X}$, any $i \leqslant p$, and any $\vec{a}_{-i} \in D_{-i}$*

[1] For every pair of alternatives differing on exactly one issue, there is exactly one weighted edge between them; the direction of the edge only says that we are going further from the correct winner. This will be made more precise after Definition 10.4.1.

with $|\vec{a}_{-i} - \vec{d}_{-i}| = k \leqslant p-1$*, we have that* $\pi_{\vec{d}}^{\vec{a}_{-i}}(d_i \succ \bar{d}_i) = q_k$.

The intuition behind the notion of a distance-based model is as follows. First, it is plausible to assume that the "closer" two alternatives are to the correct alternative, the more likely a given voter will order them in the "correct" way, that is, will prefer the one which is closer to the correct alternative. The family of distance-based voting rules is actually more general than this, because we do not impose $q_1 \geqslant \ldots \geqslant q_{p-1}$, but we may of course add this restriction if we wish to. Moreover, the choice of the Hamming distance is not necessary, and other intuitive distance-based models can be defined, using other distances – for instance, domain-dependent distances. But, the Hamming distance is a natural starting point (most works in distance-based belief base merging and distance-based belief revision also focus on the Hamming distance).

Given the correct winner $\vec{d}$, a distance-based model $\pi_{\vec{q}}$ can be visualized by the following weighted directed graph built on the hypercube:

- For any undirected edge $e = (\vec{d_1}, \vec{d_2})$ in the hypercube, where $\vec{d_1}, \vec{d_2}$ differ only on the value assigned to X_i for some $i \leqslant p$, if $\vec{d_1}|_{X_i} = d_i$, then the direction of e is from $\vec{d_1}$ to $\vec{d_2}$; if $\vec{d_2}|_{X_i} = d_i$, then the direction of e is from $\vec{d_2}$ to $\vec{d_1}$. That is, the direction of the edge is always from the alternative whose X_i component is the same as the X_i component of the correct winner to the other end of the edge.

- For any edge e with $|e - \vec{d}| = l$, the weight of e is q_l.

For example, given that $0_1 0_2 0_3$ is the correct winner, the distance-based model is illustrated in Figure 10.2.

We are especially interested in a special type of distance-based models in which there exists a threshold $1 \leqslant k \leqslant p$ and $q > \frac{1}{2}$, such that for any $i < k$, we have that $q_i = q$, and for any $k \leqslant i \leqslant p-1$, we have that $q_i = \frac{1}{2}$. Such a model is denoted

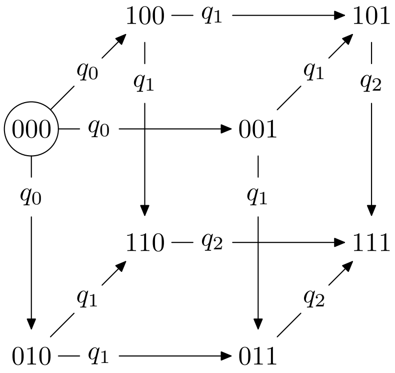
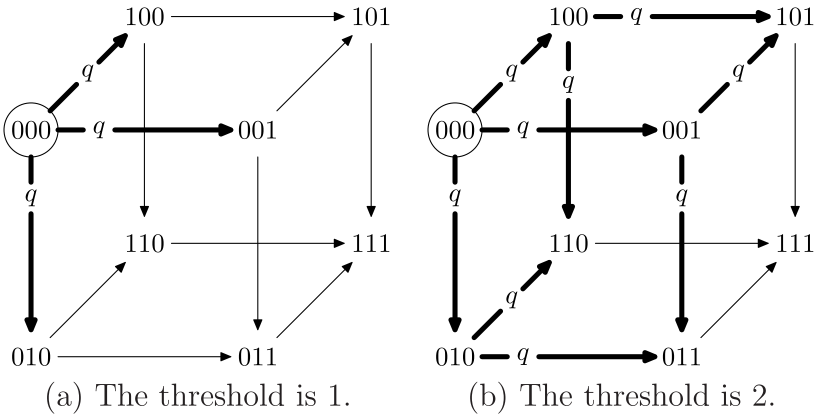


FIGURE 10.2: The distance-based model $\pi_{(q_0,q_1,q_2)}$ when the correct winner is 000.

by $\pi_{k,q}$. We call $\pi_{k,q}$ a *distance-based threshold noise model* with threshold k. We say that a noise model π has threshold $k \leqslant p$ if and only if there exists $q > \frac{1}{2}$ such that $\pi = \pi_{k,q}$. The MLE for a distance-based threshold model $\pi_{k,q}$ is denoted by $MLE_{\pi_{k,q}}$.

Example 10.4.2. *Let* $p = 3$. $\pi_{1,q}$ *and* $\pi_{2,q}$ *are illustrated in Figure 10.3 (when the correct winner is* 000*).*



(a) The threshold is 1.

(b) The threshold is 2.

FIGURE 10.3: Distance-based threshold models. The weight of the bold edges is $q > \frac{1}{2}$; the weight of all other edges is $\frac{1}{2}$.

We next present a direct method for computing winners under the MLE correspondences of distance-based threshold models. For any $1 \leqslant k \leqslant p$, any $\vec{d} \in \mathcal{X}$, and any CP-net $\mathcal{N}$, we define the *consistency of degree* k between $\vec{d}$ and $\mathcal{N}$, denoted by $N_k(\vec{d},\mathcal{N})$, as follows. $N_k(\vec{d},\mathcal{N})$ is the number of triples $(\vec{a},\vec{b},i)$ such that $\vec{a}_{-i}=\vec{b}_{-i}$, $a_i=d_i$, $b_i=\bar{d}_i$, $|(a_i,b_i)-\vec{d}|\leqslant k-1$, and $\mathcal{N}$ contains $a_{-i}:d_i\succ\bar{d}_i$. That is, $N_k(\vec{d},\mathcal{N})$ is the number of local preferences (over any issue X_i, given any $\vec{a}_{-i}\in D_{-i}$) in $\mathcal{N}$ that are $d_i\succ\bar{d}_i$, where the distance between $\vec{d}$ and the edge $((d_i,\vec{a}_{-i}),(\bar{d}_i,\vec{a}_{-i}))$ is at most $k-1$. For any profile P_{CP} of CP-nets, we let $N_k(\vec{d},P_{CP})=\sum_{\mathcal{N}\in P_{CP}}N_k(\vec{d},\mathcal{N})$.

Theorem 10.4.3. *For any $k\leqslant p$, any $q>\frac{1}{2}$, and any profile P_{CP} of CP-nets, we have that $MLE_{\pi_{k,q}}(P_{CP})=\arg\max_{\vec{d}}N_k(\vec{d},P_{CP})$.*

That is, the winner for any profile of CP-nets under any MLE for a distance-based threshold model $\pi_{k,q}$ maximizes the sum of the consistencies of degree k between the winning alternative and all CP-nets in the profile.

Proof of Theorem 10.4.3: For any $k\leqslant p$, any $\vec{d}\in\mathcal{X}$, we let $L_k=\#\{e:|e-\vec{d}|\leqslant k-1\}$. That is, L_k is the number of edges in the hypercube whose distance from a given alternative $\vec{d}$ is no more than $k-1$. For any $\vec{d}\in\mathcal{X}$ and any CP-net $\mathcal{N}$, we have that

$$\begin{aligned}&\ln\pi(P_{CP}|\vec{d})\\ =&\sum_{\mathcal{N}\in P_{CP}}\ln\prod_{i,\vec{a}_{-i}\in D_{-i}}\pi_{d_i}^{\vec{a}_{-i}}(\mathcal{N}|_{X_i:\vec{a}_{-i}})\\ =&\sum_{\mathcal{N}\in P_{CP}}(N_k(\vec{d},\mathcal{N})\ln q+(L_k-N_k(\vec{d},\mathcal{N}))\ln(1-q))\\ =&\sum_{\mathcal{N}\in P_{CP}}(N_k(\vec{d},\mathcal{N})\ln\frac{q}{1-q}+L_k\ln(1-q))\end{aligned}$$

Therefore, $MLE_{\pi_{k,q}}(P_{CP})=\arg\max_{\vec{d}}\pi(P_{CP}|\vec{d})$
$=\arg\max_{\vec{d}}\sum_{\mathcal{N}\in P_{CP}}(N_k(\vec{d},\mathcal{N})\ln\frac{q}{1-q}+L_k\ln(1-q))$
$=\arg\max_{\vec{d}}N_k(\vec{d},P_{CP})$. □

Therefore, we have the following corollary, which states that the winners for any profile under $MLE_{\pi_{k,q}}$ do not depend on q, provided that $q>\frac{1}{2}$.

Corollary 10.4.4. *For any $k \leqslant p$, any $q_1 > \frac{1}{2}, q_2 > \frac{1}{2}$, and any profile P_{CP} of CP-nets, we have $MLE_{\pi_{k,q_1}}(P_{CP}) = MLE_{\pi_{k,q_2}}(P_{CP})$.*

Example 10.4.5. *Consider two binary issues X_1, X_2, and three voters, who report the following CP-nets:*

- *$\mathcal{N}_1$ has an edge from X_1 to X_2, and the following local preferences: $\{0_1 \succ 1_1, 0_1 : 0_2 \succ 1_2, 1_1 : 1_2 \succ 0_2\}$.*
- *$\mathcal{N}_2$ has an edge from X_1 to X_2 and an edge from X_2 to X_1, and the following local preferences: $\{0_2 : 1_1 \succ 0_1, 1_2 : 0_1 \succ 1_1, 0_1 : 1_2 \succ 0_2, 1_1 : 0_2 \succ 1_2\}$.*
- *$\mathcal{N}_3$ has no edge, and the following local preferences: $\{1_1 \succ 0_1, 1_2 \succ 0_2\}$.*

Let $P_{CP} = (\mathcal{N}_1, \mathcal{N}_2, \mathcal{N}_3)$.

First, consider $k = 1$. Let us compute $N_1(1_1 1_2, \mathcal{N}_1)$. There are two edges whose distance to $1_1 1_2$ is 0: one from $1_1 1_2$ to $1_1 0_2$ and one from $1_1 1_2$ to $0_1 1_2$. The first one is in the preference relation induced from $\mathcal{N}_1$; the second one is not. Therefore, $N_1(1_1 1_2, \mathcal{N}_1) = 1$. Similarly, we get $N_1(1_1 1_2, \mathcal{N}_2) = 0$ and $N_1(1_1 1_2, \mathcal{N}_3) = 2$, henceforth, $N_1(1_1 1_2, P_{CP}) = 3$. Similar calculations lead to $N_1(1_1 0_2, P_{CP}) = 3$, $N_1(0_1 1_2, P_{CP}) = 4$ and $N_1(0_1 0_2, P_{CP}) = 2$, hence $MLE_{\pi_{1,q}}(P_{CP}) = \{0_1 1_2\}$ (for any value of $q > \frac{1}{2}$).

Now, consider $k = 2$. Let us compute $N_1(1_1 1_2, \mathcal{N}_1)$. Now, we have to consider all four edges, since all of them are at a distance 0 or 1 to $1_1 1_2$. The two edges not considered for the case $k = 1$ are the edge from $0_1 1_2$ to $0_1 0_2$ and one from $1_1 0_2$ to $0_1 1_2$. In both cases, voter 1 prefers the alternative which is further from $1_1 1_2$, therefore, $N_2(1_1 1_2, \mathcal{N}_1) = 1$. Similarly, we get $N_2(1_1 1_2, \mathcal{N}_2) = 2$ and $N_2(1_1 1_2, \mathcal{N}_3) = 4$, henceforth, $N_2(1_1 1_2, P_{CP}) = 7$. Similar calculations lead to $N_2(1_1 0_2, P_{CP}) = 5$, $N_2(0_1 1_2, P_{CP}) = 7$ and $N_1(0_1 0_2, P_{CP}) = 5$, hence $MLE_{\pi_{2,q}}(P_{CP}) = \{0_1 1_2, 1_1 1_2\}$.

We next investigate the computational complexity of applying MLE rules with distance-based threshold models. First, we present a polynomial-time algorithm that computes the winners and outputs the winners in a compact way, under $MLE_{\pi_{p,q}}$, where p is the number of issues. This algorithm computes the correct value(s) of each issue separately: for any issue X_i, the algorithm counts the number of tuples $(\vec{a}_{-i}, \mathcal{N})$, where $\vec{a}_{-i} \in D_{-i}$ and $\mathcal{N}$ is a CP-net in the input profile P_{CP}, such that $\mathcal{N}$ contains $a_{-i} : 0_i \succ 1_i$. If there are more tuples $(\vec{a}_{-i}, \mathcal{N})$ in which $\mathcal{N}$ contains $a_{-i} : 0_i \succ 1_i$ than there are tuples in which $\mathcal{N}$ contains $a_{-i} : 1_i \succ 0_i$, then we select 0_i to be the ith component of the winning alternative, and vice versa. We note that the time required to count tuples $(\vec{a}_{-i}, \mathcal{N})$ depends on the size of $\mathcal{N}$. Therefore, even though computing the value for X_i takes time that is exponential in $|Par_G(X_i)|$ (the number of parents of X_i in the directed graph of $\mathcal{N}$), the CPT of X_i in $\mathcal{N}$ itself is also exponential in $|Par_G(X_i)|$ (for each setting of $Par_G(X_i)$, there is an entry in $CPT(X_i)$). This explains why the algorithm runs in polynomial time.

Algorithm 10.4.1. INPUT: $p \in \mathbb{N}$, $\frac{1}{2} < q < 1$, and a profile of CP-nets P_{CP} over a binary domain consisting of p issues.

1. For each $i \leqslant p$:

1a. Let $S_i = 0$, $W_i = \varnothing$.

1b. For each CP-net $\mathcal{N} \in P_{CP}$: let $Par_G(X_i) = \{X_{i_1}, \ldots, X_{i_{p'}}\}$ be the parents of X_i in the directed graph of $\mathcal{N}$. Let l be the number of settings $\vec{y}$ of $Par_G(X_i)$ for which $\mathcal{N}|_{X_i:\vec{y}} = 0_i \succ 1_i$. Let $S_i \leftarrow S_i + l2^{p-p'} - 2^{p-1}$. Here, p' is the number of parents of X_i, and $l2^{p-p'} - 2^{p-1}$ is the number of edges in the CP-net where $0_i \succ 1_i$, minus the number of edges where $1_i \succ 0_i$.

1c. At this point, let $W_i = \begin{cases} \{0_i\} & \text{if } S_i > 0 \\ \{1_i\} & \text{if } S_i < 0 \\ \{0_i, 1_i\} & \text{if } S_i = 0 \end{cases}$

2.Output $W_1 \times \ldots \times W_p$.

Proposition 10.4.6. *The output of Algorithm 10.4.1 is $MLE_{\pi_{p,q}}(P_{CP})$, and the algorithm runs in polynomial time.*

Proof of Proposition 10.4.6: First we prove that the output of Algorithm 10.4.1 is $MLE_{\pi_{p,q}}(P_{CP})$. For any $\vec{d} \in \mathcal{X}$, $N_p(\vec{d}, P_{CP}) = \sum_{i \leqslant p,} \#\{\vec{a}_{-1} \in D_{-1} : (d_i, \vec{a}_{-i}) \succ_{\mathcal{N}} (\bar{d}_i, \vec{a}_{-i}), \mathcal{N} \in P_{CP}\}$. We note that $d_i \in W_i$ if and only if $\#\{\vec{a}_{-1} \in D_{-1} : (d_i, \vec{a}_{-i}) \succ_{\mathcal{N}} (\bar{d}_i, \vec{a}_{-i}), \mathcal{N} \in P_{CP}\} > \#\{\vec{a}_{-1} \in D_{-1} : (\bar{d}_i, \vec{a}_{-i}) \succ_{\mathcal{N}} (d_i, \vec{a}_{-i}), \mathcal{N} \in P_{CP}\}$. Therefore, $\vec{d} \in MLE_{\pi_{p,q}}(P_{CP})$ if and only if for all $i \leqslant p$, we have that $d_i \in W_i$.

Next we prove that the algorithm runs in polynomial time. We note that in step 1b, the complexity of computing l is $O(2^{|Par_G(X_i)|})$, and $CPT(X_i)$ of the CP-net $\mathcal{N}$ has exactly $2^{|Par_G(X_i)|}$ entries, which means that the complexity of computing l is in polynomial of the size of $CPT(X_i)$ of the input. Therefore, Algorithm 10.4.1 is a polynomial-time algorithm. □

The next example shows how to compute the winners under $MLE_{\pi_{p,q}}$ for the profile defined in Example 10.4.5.

Example 10.4.5, continued *Let us first compute S_1. In $\mathcal{N}_1$ (respectively, $\mathcal{N}_1$ and $\mathcal{N}_3$), the table for x_1 contributes to 2 edges (respectively, one edge and no edge) from 0_1 to 1_1, and to no edge (respectively, one edge and two edge) from 1_1 to 0_1, therefore $S_1 = (+2) + 0 + (-2) = 0$. Similarly, $S_2 = 0 + 0 + (-2) = -2$. Therefore, $W_1 = \{0_1, 1_1\}$ and $W_2 = \{1_2\}$, which gives us $MLE_{\pi_{2,q}}(P_{CP}) = \{0_1 1_2, 1_1 1_2\}$.*

However, when the threshold is one, computing the winners is NP-hard, and the associated decision problem, namely checking whether there exists an alternative $\vec{d}$ such that $N_1(\vec{d}, P_{CP}) \geqslant T$, is NP-complete.

Theorem 10.4.7. *It is NP-complete to find a winner under $MLE_{\pi_{1,q}}$. More precisely, it is NP-complete to decide whether there exists an alternative $\vec{d}$ such that $N_1(\vec{d}, P_{CP}) \geqslant T$.*

Proof of Theorem 10.4.7: By Theorem 10.4.3, the decision problem of finding a winner under $MLE_{\pi_{1,q}}$ is the following: for any profile P that consists of n CP-nets, and any $T \leqslant pn$, we are asked whether or not there exists $\vec{d} \in \mathcal{X}$ such that

$N_1(\vec{d}, P) \geqslant T$.

We prove the NP-hardness by reduction from the decision problem of MAX2SAT. The inputs of an instance of the decision problem of MAX2SAT consists of (1) a set of t atomic propositions $x_1, \ldots, x_t$; (2) a formula $F = C_1 \wedge \ldots \wedge C_m$ represented in *conjunctive normal form*, in which for any $i \leqslant m$, $C_i = l_{i_1} \vee l_{i_2}$, and there exists $j_1, j_2 \leqslant t$ such that l_{i_1} is x_{j_1} or $\neg x_{j_1}$, and l_{i_2} is x_{j_2} or $\neg x_{j_2}$; (3) $T \leqslant m$. We are asked whether or not there exists a valuation $\vec{x}$ for the atomic propositions $x_1, \ldots, x_t$ such that at least T clauses are satisfied under $\vec{x}$.

Given any instance of MAX2SAT, we construct a decision problem instance of computing a winner under $MLE_{\pi_{1,q}}$ as follows.

- Let $\mathcal{X}$ be composed of t issues $X_1, \ldots, X_t$.
- Let $T' = 16T - 12m$.
- For any $i \leqslant m$, we let v_{i_1} be the valuation of x_{i_1} under which l_{i_1} is true; let v_{i_2} be the valuation of x_{i_2} under which l_{i_2} is true. For any $j \leqslant t$, we let 0_j corresponds to X_j being false, and 1_j corresponds to X_j being true. Then, any valuation of the atomic propositions is uniquely identified by an alternative. We next define six CP-nets as follows:

– $\mathcal{N}_{i,1}$: the DAG of $\mathcal{N}_{i,1}$ has only one directed edge (X_{i_1}, X_{i_2}). In $\mathcal{N}_{i,1}$, $v_{i_1} \succ \bar{v}_{i_1}$, $v_{i_1} : v_{i_2} \succ \bar{v}_{i_2}$, $\bar{v}_{i_1} : v_{i_2} \succ \bar{v}_{i_2}$, and for any $j \neq i_1$ and $j \neq i_2$, we have that $0_j \succ 1_j$.

– $\mathcal{N}_{i,2}$: the DAG of $\mathcal{N}_{i,2}$ has only one directed edge (X_{i_1}, X_{i_2}). In $\mathcal{N}_{i,2}$, $v_{i_1} \succ \bar{v}_{i_1}$, $v_{i_1} : \bar{v}_{i_2} \succ v_{i_2}$, $\bar{v}_{i_1} : v_{i_2} \succ \bar{v}_{i_2}$, and for any $j \neq i_1$ and $j \neq i_2$, we have that $0_j \succ 1_j$.

– $\mathcal{N}_{i,3}$: the DAG of $\mathcal{N}_{i,3}$ has only one directed edge (X_{i_2}, X_{i_1}). In $\mathcal{N}_{i,1}$, $v_{i_2} \succ \bar{v}_{i_2}$, $v_{i_2} : \bar{v}_{i_1} \succ v_{i_1}$, $\bar{v}_{i_2} : v_{i_1} \succ \bar{v}_{i_1}$, and for any $j \neq i_1$ and $j \neq i_2$, we have that $0_j \succ 1_j$.

We next obtain $\mathcal{N}'_{i,1}$, $\mathcal{N}'_{i,2}$, and $\mathcal{N}'_{i,3}$ from $\mathcal{N}_{i,1}$, $\mathcal{N}_{i,2}$, and $\mathcal{N}_{i,3}$, respectively, by letting $1_j \succ 0_j$ for any j with $j \neq i_1$ and $j \neq i_2$. Let $\vec{\mathcal{N}_i} = (\mathcal{N}_{i,1}, \mathcal{N}'_{i,1}, \mathcal{N}_{i,2}, \mathcal{N}'_{i,2}, \mathcal{N}_{i,3}, \mathcal{N}'_{i,3})$. We let the profile of CP-nets be $P_{CP} = (\vec{\mathcal{N}}_1, \ldots, \vec{\mathcal{N}}_m)$.

We make the following claim about the number of consistent edges between an alternative $\vec{d}$ and $\vec{\mathcal{N}}_i$.

Claim 10.4.1. *For any $\vec{d} \in \mathcal{X}$ and any $i \leqslant m$,*

$$N_1(\vec{d}, \vec{\mathcal{N}}_i) = \begin{cases} 4 & \text{if } \vec{d}_{i_1} = v_{i_1} \text{ or } d_{i_2} = v_{i_2} \\ -12 & \text{if } \vec{d}_{i_1} = \bar{v}_{i_1} \text{ and } d_{i_2} = \bar{v}_{i_2} \end{cases}$$

Claim 10.4.1 states that the number of consistent edges between $\vec{d}$ and $\vec{\mathcal{N}}_i$ within distance 1 is 4 if the clause C_i is true under the valuation represented by $\vec{d}$; otherwise it is -12. For any $\vec{d} \in \mathcal{X}$, we let $T_{\vec{d}}$ denote the number of clauses in $C_1, \ldots, C_m$ that are true under $\vec{d}$. Then, we have that $N_1(\vec{d}, P_{CP}) = 4T_{\vec{d}} - 12(m - T_{\vec{d}}) = 16T_{\vec{d}} - 12m$. It follows from Theorem 10.4.3 that for any $q > \frac{1}{2}$, $MLE_{\pi_{1,q}}(P_{CP}) = \arg\max_{\vec{d}} N_1(\vec{d}, P_{CP}) = \arg\max_{\vec{d}} T_{\vec{d}}$. Therefore, a winner of P_{CP} under $MLE_{\pi_{1,q}}$ corresponds to a valuation under which the number of satisfied clauses is maximized; and any valuation that maximizes the number of satisfied clauses corresponds to a winner of P_{CP} under $MLE_{\pi_{1,q}}$. We note that the size of P_{CP} is $O(mt)$. It follows that computing a winner under $MLE_{\pi_{1,q}}$ is NP-hard.

Clearly the decision problem is in NP. Therefore, the decision problem is NP-complete to compute a winner under $MLE_{\pi_{1,q}}$. □

As we have seen (cf. Corollary 10.4.4), for a given multi-issue domain composed of p binary issues, there are *exactly* p voting correspondences defined by distance-based threshold models. As far as we know, these voting correspondences are entirely novel, and are tailored especially for multi-issue domains. Now, among these p voting correspondences, two are even more natural and interesting: $MLE_{\pi_{1,q}}$ and $MLE_{\pi_{p,q}}$. $MLE_{\pi_{1,q}}$ proceeds by electing the alternatives which maximize the sum, over all voters, of the number of neighboring alternatives in the voter's hypercube to which she prefers $\vec{x}$. Now, recall that the Borda correspondence can be characterized as the correspondence where candidate x is a winner if it maximizes the sum, over all voters, of the number of candidates the voter prefers to x. Therefore, $MLE_{\pi_{1,q}}$ is somewhat reminiscent of Borda—except, of course, that we do not count all alternatives defeated by $\vec{x}$ but only defeated alternatives that are one of its neighbors in the hypercube. $MLE_{\pi_{p,q}}$ is even more intuitive: for each issue X_i, the winning value maximizes the number of edges (summing over all voters) that are in favor of it, that is, it is somewhat reminiscent of Kemeny.

So, $MLE_{\pi_{1,q}}$ and $MLE_{\pi_{p,q}}$ are genuinely new voting correspondences for multi-issue binary domains, which can be characterized in terms of maximum likelihood estimators and are quite intuitive; lastly, $MLE_{\pi_{p,q}}$ can be computed in polynomial time. We conjecture that for any $2 \leqslant k \leqslant p-1$, winner determination for $MLE_{\pi_{k,q}}$ is NP-hard.

10.5 Summary

In this chapter, we considered the maximum likelihood estimation (MLE) approach to voting, and generalized it to multi-issue domains, assuming that the voters' preferences are expressed by CP-nets. We first studied whether issue-by-issue voting rules and sequential voting rules can be represented by the MLE of some noise model. For separable input profiles, we characterized MLEs of strongly/weakly decomposable models as issue-by-issue voting correspondences composed of local MLEWIVs/candidate scoring correspondences. Although we showed that no sequential voting correspondence can be represented as the MLE for a very weakly decomposable model, we did obtain a positive result here under the assumption that the number of voters is bounded above by a constant.

In the case where all issues are binary, we proposed a class of distance-based noise models; then, we focused on a specific subclass of such models, parameterized by a threshold. We identified the computational complexity of winner determination for the two most relevant values of the threshold.

We note that, whereas Section 10.3 has a non-constructive flavor because we studied existing voting mechanisms and Theorem 10.3.3 is an impossibility theorem, quite the opposite is the case for Section 10.4. Indeed, the MLE principle led us to define genuinely new families of voting rules and correspondences for multi-issue domains. These rules are radically different from the rules that had previously been proposed and studied for these domains. Unlike sequential or issue-by-issue rules, they do not require any domain restriction, and yet their computational complexity is not that bad (the decision problem is NP-complete at worst, and sometimes polynomial in the size of the CP-nets). We believe that these new rules are promising.

11

Strategic Sequential Voting

In previous two chapters we have been focusing on designing "good" voting rules for combinatorial voting. In most of the previous work on combinatorial voting, it was assumed that the voters report their true preferences using the voting language we provide to them, when the voting language is expressive enough to do so. Now, if the voters vote on issues sequentially, one issue after another according to some ordering over issues, and are assumed to know the preferences of other voters well enough, then we can expect them to vote strategically at each step, forecasting the outcome at later steps conditional on the outcomes at earlier steps. Let us consider the following motivating example (a similar example was shown in Lacy and Niou (2000)).

Example 11.0.1. Three residents want to vote to decide whether they should build a swimming pool and/or a tennis court. There are two issue **S** and **T**. **S** can take the value of s (meaning "to build the swimming pool") or $\bar{s}$ (meaning "not to build the swimming pool"). Similarly, **T** takes a value in $\{t, \bar{t}\}$. Suppose the preferences of the three voters are, respectively, $st \succ \bar{s}t \succ s\bar{t} \succ \bar{s}\bar{t}$, $s\bar{t} \succ st \succ \bar{s}t \succ \bar{s}\bar{t}$ and

$\bar{s}t \succ \bar{s}\bar{t} \succ s\bar{t} \succ st$. Voter 2 and 3 do not rank st as their first choices, because they thought that the money could be spent on something else. Suppose the voters first vote on issue **S** then on issue **T**. Since both issues are binary, the local rule used at each step is majority (there will be no ties, because the number of voters is odd). Voter 1 is likely to reason in the following way: *if the outcome of the first step is s, then voters 2 and 3 will vote for $\bar{t}$, since they both prefer $s\bar{t}$ to st, and the final outcome will be $s\bar{t}$; but if the outcome of the first step is $\bar{s}$, then voters 2 and 3 will vote for t, and the final outcome will be $\bar{s}t$; because I prefer $\bar{s}t$ to $s\bar{t}$, I am better off voting for $\bar{s}$, since either it will not make any difference, or it will lead to a final outcome of $\bar{s}t$ instead of $s\bar{t}$.* If voters 2 and 3 reason in the same way, then 2 will vote for s and 3 for $\bar{s}$; hence, the result of the first step is $\bar{s}$, and then, since two voters out of three prefer $\bar{s}t$ to $\bar{s}\bar{t}$, the final outcome will be $\bar{s}t$. Note that the result is fully determined, provided that (1) it is common knowledge that voters behave strategically according to the principle we have stated informally, (2) the order in which the issues are decided, as well as the local voting rules used in all steps, are also common knowledge, and (3) voters' preferences are common knowledge. Therefore, these three assumptions allow the voters and the modeler (provided he knows as much as the voters) to predict the final outcome.

Let us take a closer look at voter 1 in Example 11.0.1. Her preferences are *separable*: she prefers s to $\bar{s}$ whatever the value of **T** is, and t to $\bar{t}$ whatever the value of **S** is. *And yet she strategically votes for $\bar{s}$*, because the outcome for **S** affects the outcome for **T**. Moreover, while voters 2 and 3 have nonseparable preferences, still, all three voters' preferences enjoy the following property: their preferences over the value of **S** are independent of the value of **T**. That is, the profile is ($\mathbf{S} > \mathbf{T}$)-legal. Hence, we can apply the sequential voting rule w.r.t. the order $\mathbf{S} > \mathbf{T}$, using majority rules for **S** and **T**. For the profile given in Example 11.0.1, the outcome of the first step under the sequential voting rule will be s (since two voters out of three prefer s to $\bar{s}$, unconditionally), and the final outcome will be $s\bar{t}$. This outcome is different from the outcome we obtain if voters behave strategically. The reason for this discrepancy is that in Lang and Xia (2009), voters are not assumed to know the others' preferences and are assumed to vote truthfully.

We have seen that even if the voters' preferences are $\mathcal{O}$-legal, voters may in fact have no incentive to vote truthfully. Consequently, existing results on multiple-election paradoxes are not directly applicable to situations where voters vote strategically.

Overview of this chapter

In this chapter, we analyze the complete-information game-theoretic model of sequential voting that we illustrated in Example 11.0.1. This model applies to any preferences that the voters may have (not just $\mathcal{O}$-legal ones), though they must be strict orders on the set of all alternatives.

We focus on voting in multi-binary-issue domains, that is, for any $i \leqslant p$, X_i must take a value in $\{0_i, 1_i\}$. This has the advantage that for each issue, we can use the majority rule as the local rule for that issue. We use a game-theoretic model to analyze outcomes that result from sequential voting. Specifically, we model the sequential voting process as a p-stage complete-information game as follows. There is an order $\mathcal{O}$ over all issues (without loss of generality, let $\mathcal{O} = X_1 > X_2 > \cdots > X_p$), which indicates the order in which these issues will be voted on. For any $1 \leqslant i \leqslant p$, in stage i, the voters vote on issue X_i simultaneously, and the majority rule is used to choose the winning value for X_i. We make the following game-theoretic assumptions: it is common knowledge that all voters are perfectly rational; the order $\mathcal{O}$ and the fact that in each step, the majority rule is used to determine the winner are common knowledge; all voters' preferences are common knowledge.

We can solve this game by a type of backward induction already illustrated in Example 11.0.1: in the last (pth) stage, only two alternatives remain (corresponding to the two possible settings of the last issue), so at this point it is a weakly dominant strategy for each voter to vote for her more preferred alternative of the two. Then, in the second-to-last (($p-1$)th) stage, there are two possible local outcomes for the ($p-1$)th issue; for each of them, the voters can predict which alternative will finally be chosen, because they can predict what will happen in the pth stage. Thus, the ($p-1$)th stage is effectively a majority election between two alternatives, and each voter will vote for her more preferred alternative; etc. We call this procedure the *strategic sequential voting procedure (SSP)*.[1]

Given exogenously the order $\mathcal{O}$ over the issues, this game-theoretic analysis maps every profile of strict ordinal preferences to a unique outcome. Since any function from profiles of preferences to alternatives can be interpreted as a voting rule, the voting rule that corresponds to SSP is denoted by $SSP_{\mathcal{O}}$.

Lacy and Niou (2000) showed that whenever there exists a Condorcet winner, it must be the SSP winner. That is, SSP is Condorcet consistent. We will show that, unfortunately, all three major types of multiple-election paradoxes (see Section 8.1) also arise under SSP. To better present our results, we introduce a parameter which we call the *minimax satisfaction index (MSI)*. For an election with m alternatives and n voters, it is defined in the following way. For each profile, consider the highest position that the winner obtains across all input rankings of the alternatives (the ranking where this position is obtained corresponds to the most-satisfied voter); this is the *maximum satisfaction index* for this profile. Then, the minimax satisfaction index is obtained by taking the minimum over all profiles of the maximum satisfaction index. A low minimax satisfaction index means that there exists a profile in which

[1] Lacy and Niou (2000) called such a procedure *sophisticated voting* following the convention of Farquharson (1969).

the winner is ranked in low positions in all votes, thus indicating a multiple-election paradox. Our main theorem is the following.

Theorem 11.3.1 *For any $p \in \mathbb{N}$ and any $n \geqslant 2p^2 + 1$, the minimax satisfaction index of SSP when there are $m = 2^p$ alternatives and n voters is $\lfloor p/2 + 2 \rfloor$. Moreover, in the profile P that we use to prove the upper bound, the winner $SSP_{\mathcal{O}}(P)$ is Pareto-dominated by $2^p - (p+1)p/2$ alternatives.*

We note that an alternative c Pareto-dominates another alternative c' implies that c beats c' in their pairwise election. Therefore, Theorem 11.3.1 implies that the winner for SSP is an almost Condorcet loser. It follows from this theorem that SSP exhibits all three types of multiple-election paradoxes: the winner is ranked almost in the bottom in every vote, the winner is an almost Condorcet loser, and the winner is Pareto-dominated by almost every other alternative. We further show a paradox (Theorem 11.3.6) that states that there exists a profile such that for *any* order $\mathcal{O}$ over the issues, for every voter, the SSP winner w.r.t. $\mathcal{O}$ is ranked almost in the bottom position. We also show that even when the voters' preferences can be represented by CP-nets that are compatible with a common order, multiple-election paradoxes still arise.

Related work and discussion

The setting of SSP has been considered by Lacy and Niou (2000). But at a high level, our motivation, results, and conclusion are quite different from those of Lacy and Niou. We focus on the game-theoretic aspects of SSP, and we aim at examining the equilibrium outcomes in voting games. They viewed SSP as a voting rule (see Section 11.1.3 for more discussion on this point of view), and aimed at proposing solutions to aggregate non-separable profiles in combinatorial voting. They showed that SSP satisfies Condorcet consistency, but did not mention whether the other types of multiple-election paradoxes can be avoided. We, on the other hand, show that the other three types of multiple-election paradoxes still arise in SSP.[2] In terms of their conclusion, Lacy and Niou argued that SSP might not be a good solution as a voting rule, because it requires the voters to have complete information about the other voters' true preferences. The paradoxes that will be shown in this chapter, like the paradoxes we showed for Stackelberg voting games in Chapter 7, are an ordinal version of price-of-anarchy results. Consequently, these paradoxes provide more evidence that strategic behavior of the voters should be prevented, and therefore motivate the study in the next chapter, where the objective is to design strategy-proof voting rules that are computationally tractable for combinatorial domains.

More generally, SSP is closely related to *multi-stage sophisticated voting*, studied by McKelvey and Niemi (1978), Moulin (1979), and Gretlei (1983). They investigated the model where the backward induction outcomes correspond to the truthful outcomes of voting trees. Therefore, SSP is a special case of multi-stage sophisticated voting. However, their work focused on the characterization of the outcomes as the outcomes in sophisticated voting (Farquharson, 1969), and therefore did not shed much light on the quality of the equilibrium outcome. We, on the other hand, are primarily interested in the strategic outcome of the natural procedure of voting sequentially over multiple issues. Also, the relationship between sequential voting and voting trees takes a particularly natural form in the context of domains with multiple binary issues, as we will show. More importantly, we illustrate several multiple-election paradoxes for SSP, indicating that the equilibrium outcome could be extremely undesirable.

Another paper that is closely related to part of this work was written by
Dutta and Sen (1993). They showed that social choice rules corresponding to binary voting trees can be implemented via backward induction via a sequential vot-

[2] In fact, those paradoxes were also discovered by Lacy and Niou in the same paper (Lacy and Niou, 2000), but they did not discuss whether they arise in SSP. See Section 8.1.

ing mechanism. This is closely related to the relationship revealed for multi-stage sophisticated voting and will also be mentioned later in this chapter, that is, an equivalence between the outcome of strategic behavior in sequential voting over multiple binary issues, and a particular type of voting tree. It should be pointed out that the sequential mechanism that Dutta and Sen consider is somewhat different from sequential voting as we consider it—in particular, in the Dutta-Sen mechanism, one voter moves at a time, and a move consists not of a vote, but rather of choosing the next player to move (or in some states, choosing the winner).

Nevertheless, the approach by Dutta and Sen and our approach are related at a high level, though they are motivated quite differently: Dutta and Sen are interested in social choice rules corresponding to voting trees, and are trying to create sequential mechanisms that implement them via backward induction. We, on the other hand again, are primarily interested in the strategic outcome of the natural mechanism for voting sequentially over multiple issues, and use voting trees merely as a useful tool for analyzing the outcome of this process.

11.1 Strategic Sequential Voting

11.1.1 Formal Definition

In this chapter, we focus on multi-binary-issue domains. That is, the multi-issue domain is composed of multiple binary issues. Sequential voting on multi-binary-issue domains can be seen as a game where in each step, the voters decide whether to vote for or against the issue under consideration after reasoning about what will happen next. We make the following assumptions.

1. All voters act strategically (in an optimal manner that will be explained later), and this is common knowledge.

2. The order in which the issues will be voted upon, as well as the local voting rules

used at the different steps (namely, majority rules), are common knowledge.

3. All voters' preferences on the set of alternatives are common knowledge.

Assumption 1 is standard in game theory. Assumption 2 merely means that the rule has been announced. Assumption 3 (complete information) is the most significant assumption. It may be interesting to consider more general settings with incomplete information, resulting in a Bayesian game. Nevertheless, because the complete-information setting is a special case of the incomplete-information setting (where the prior distribution is degenerate), in that sense, *all the worst-case negative results obtained for the complete-information setting also apply to the incomplete-information setting.* That is, the restriction to complete information only strengthens negative results. Of course, for incomplete information setting in general, we need a more elaborate model to reason about voters' strategic behavior.

Given these assumptions, the voting process can be modeled as a game that is composed of p stages where in each stage, the voters vote simultaneously on one issue. Let $\mathcal{O}$ be the order over the set of issues, which without loss of generality we assume to be $X_1 > \cdots > X_p$. Let P be the profile of preferences over $\mathcal{X}$. The game is defined as follows: for each $i \leqslant p$, in stage i the voters vote simultaneously on issue i; then, the value of X_i is determined by the majority rule (plus, in the case of an even number of voters, some tie-breaking mechanism), and this local outcome is broadcast to all voters.

We now show how to solve the game. Because of assumptions 1 to 3, at step i the voters vote strategically, by recursively figuring out what the final outcome will be if the local outcome for X_i is 0_i, and what it will be if it is 1_i. More concretely, suppose that steps 1 to $i-1$ resulted in issues $X_1, \ldots, X_{i-1}$ taking the values $d_1, \ldots, d_{i-1}$, and let $\vec{d} = (d_1, \ldots, d_{i-1})$. Suppose also that if X_i takes the value 0_i (respectively, 1_i), then, recursively, the remaining issues will take the tuple of values $\vec{a}$ (respectively,

$\vec{b}$). Then, X_i is determined by a pairwise comparison between $(\vec{d}, 0_i, \vec{a})$ and $(\vec{d}, 1_i, \vec{b})$ in the following way: if the majority of voters prefer $(\vec{d}, 0_i, \vec{a})$ over $(\vec{d}, 1_i, \vec{b})$, then X_i takes the value 0_i; in the opposite case, X_i takes the value 1_i. This process, which corresponds to the strategic behavior in the sequential election, is what we call the *strategic sequential voting (SSP)* procedure, and for any profile P, the winner with respect to the order $\mathcal{O}$ is denoted by $SSP_{\mathcal{O}}(P)$.

As we shall see later, SSP can not only be thought of as the strategic outcome of sequential voting, but also as a voting rule in its own right. The following definition and two propositions merely serve to make the game-theoretic solution concept that we use precise; a reader who is not interested in this may safely skip them.

Definition 11.1.1. *Consider a finite extensive-form game which transitions among* states. *In each nonterminal state s, all players simultaneously take an action; this joint local action profile $(a_1^s, \ldots, a_n^s)$ determines the next state s'.*[3] *Terminal states t are associated with payoffs for the players (alternatively, players have ordinal preferences over the terminal states). The current state is always common knowledge among the players.*[4]

Suppose that in every final nonterminal state s (that is, every state that has only terminal states as successors), every player i has a (weakly) dominant action a_i^s. At each final nonterminal state, its local profile of dominant actions $(a_1^s, \ldots, a_n^s)$ results in a terminal state $t(s)$ and associated payoffs. We then replace each final nonterminal state s with the terminal state $t(s)$ that its dominant-strategy profile leads to. Furthermore suppose that in the resulting smaller tree, again, in every final nonterminal state, every player has a (weakly) dominant strategy. Then, we can repeat this procedure, etc. *If we can repeat this all the way to the root of the tree,*

[3] In the extensive-form representation of the game, each state is associated with multiple nodes, because in the extensive form only one player can move at a node.

[4] Hence, the only imperfect information in the extensive form of the game is due to simultaneous moves within states.

then we say that the game is solvable by within-state dominant-strategy backward induction (WSDSBI).

We note that the backward induction in perfect-information extensive-form games is just the special case of WSDSBI where in each state only one player acts.

Proposition 11.1.2. *If a game is solvable by WSDSBI, then the solution is unique.*

Proposition 11.1.3. *The complete-information sequential voting game with binary issues (with majority as the local rule everywhere) is solvable by WSDSBI when voters have strict preferences over the alternatives.*

Both propositions are straightforward to prove and have been mentioned implicitly in Lacy and Niou (2000). We note that SSP corresponds to a particular balanced voting tree, as illustrated in Figure 11.1 for the case $p = 3$. In this voting tree, in the first round, each alternative is paired up against the alternative that differs only on the pth issue; each alternative that wins the first round is then paired up with the unique other remaining alternative that differs only on the $(p-1)$th and possibly the pth issue; etc. This bottom-up procedure corresponds exactly to the backward induction (WSDSBI) process.

Of course, there are many voting trees that do *not* correspond to an SSP election; this is easily seen by observing that there are only $p!$ different SSP elections (corresponding to the different orders of the issues), but many more voting trees. The voting tree corresponding to the order $\mathcal{O} = X_1 > \cdots > X_p$ is defined by the property that for any node v whose depth is i (where the root has depth 1), the alternative associated with any leaf in the left (respectively, right) subtree of v gives the value 0_i (respectively, 1_i) to X_i.

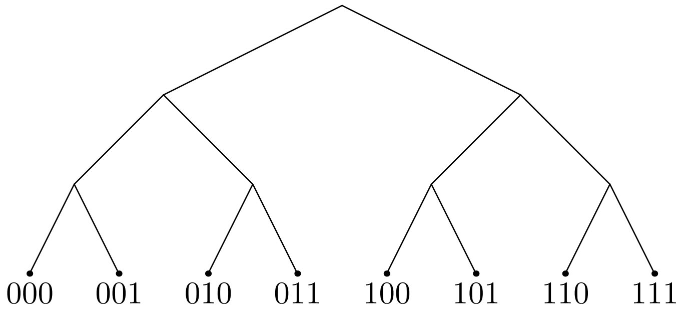


FIGURE 11.1: A voting tree that is equivalent to the strategic sequential voting procedure ($p = 3$). 000 is the abbreviation for $0_1 0_2 0_3$, etc.

11.1.2 Strategic Sequential Voting vs. Truthful Sequential Voting

We have seen on Example 11.0.1 that even when the profile P is $\mathcal{O}$-legal, $SSP_{\mathcal{O}}(P)$ can be different from $Seq_{\mathcal{O}}(Maj, \ldots, Maj)(P)$. This means that even if the profile is $\mathcal{O}$-legal, voters may be better off voting strategically than truthfully. However, $SSP_{\mathcal{O}}(P)$ and $Seq_{\mathcal{O}}(Maj, \ldots, Maj)(P)$ are guaranteed to coincide under the further restriction that P is $\mathcal{O}$-lexicographic.

Proposition 11.1.4. *For any $\mathcal{O}$-lexicographic profile P,*

$$SSP_{\mathcal{O}}(P) = Seq_{\mathcal{O}}(Maj, \ldots, Maj)(P)$$

The intuition for Proposition 11.1.4 is as follows: if P is $\mathcal{O}$-lexicographic, then, as is shown in the proof of the proposition, when voters vote strategically under sequential voting (the *Seq* process), they are best off voting according to their true preferences in each round (their preferences in each round are well-defined because voters have $\mathcal{O}$-legal preferences in this case). When voters with $\mathcal{O}$-legal preferences vote truthfully in each round under sequential voting, the outcome is $Seq_{\mathcal{O}}(Maj, \ldots, Maj)(P)$; when they vote strategically, the outcome is $SSP_{\mathcal{O}}(P)$; and so, these must be the same when preferences are $\mathcal{O}$-lexicographic.

Now, there is another interesting domain restriction under which $SSP_{\mathcal{O}}(P)$ and $Seq(Maj, \ldots, Maj)(P)$ coincide, namely when P is $inv(\mathcal{O})$- legal, where $inv(\mathcal{O}) =$

$(X_p > \ldots > X_1)$.

Proposition 11.1.5. *Let $inv(\mathcal{O}) = X_p > \ldots > X_1$. For any $inv(\mathcal{O})$-legal profile P, $SSP_{\mathcal{O}}(P) = Seq_{inv(\mathcal{O})}(Maj, \ldots, Maj)(P)$.*

As a consequence, when P is separable, it is *a fortiori* $inv(\mathcal{O})$-legal, and therefore, $SSP_{\mathcal{O}}(P) = Seq_{inv(\mathcal{O})}(Maj, \ldots, Maj)(P)$, which in turn is equal to $Seq_{\mathcal{O}}(Maj, \ldots, Maj)(P)$ and coincides with issue-by-issue voting.

Corollary 11.1.6. *If P is separable, then $SSP_{\mathcal{O}}(P) = Seq_{\mathcal{O}}(Maj, \ldots, Maj)(P)$.*

11.1.3 A Second Interpretation of SSP

The first interpretation of SSP (that we follow in this chapter) is the one we have discussed so far, namely, SSP consists in modeling sequential voting as a complete-information game, which allows us to analyze sequential voting on multi-issue domains from a game-theoretic point of view. For this, assumptions 1, 2, and 3 above are crucial. Under this interpretation, $SSP_{\mathcal{O}}(P)$ is a (specific kind of) equilibrium for sequential voting.

However, there is a second interpretation of SSP. It consists in seeing $SSP_{\mathcal{O}}$ as a new voting rule on multi-issue domains (which is implementable in complete-information contexts by using sequential voting).[5] This seems to be the point of view of Lacy and Niou (2000). This defines a family of voting rules (one for each order over issues), which can be applied to any profile. The family of voting rules thus defined is a distinguished subset of the family of voting trees. This interpretation does not say anything about how preferences are to be elicited; unlike in the game-theoretic interpretation, the p-step protocol does not apply here. The communication complexity of finding the outcome of $SSP_{\mathcal{O}}$ (without any complete-information assumption, of

[5] Of course, by Gibbard-Satterthwaite (Gibbard, 1973; Satterthwaite, 1975), SSP is not strategy-proof.

course)[6] is given as follows.

Proposition 11.1.7. *When the voters' preferences over alternatives are unrestricted, the communication complexity of $SSP_{\mathcal{O}}$ is $\Theta(2^p \cdot n)$.*

Proof of Proposition 11.1.7: This now follows immediately from a result in Conitzer and Sandholm (2005b), where it is established that the communication complexity for balanced voting trees is $\Theta(m \cdot n)$ for m alternatives and n voters. Since we do not place any restrictions on the preferences in the multi-issue domain in the statement of the proposition, the communication complexity is identical, and $m = 2^p$. □

The upper bound in this proposition is obtained simply by eliciting the voters' preferences for every pair of alternatives that face each other in the voting tree.

Now, Propositions 11.1.4 and 11.1.5 immediately give us conditions under which this communication complexity can be reduced. Indeed, these Propositions say that when P is $\mathcal{O}$-lexicographic or $inv(\mathcal{O})$-legal, then the SSP winner coincides with the sequential election winner in the sense of Lang and Xia (2009). Now, the sequential election winner in the sense of Lang and Xia (2009) can be found with $O(pn)$ communication, simply by having each agent vote for a value for the issue at each round. This leads immediately to the following two corollaries (to Propositions 11.1.4 and 11.1.5, respectively).

Corollary 11.1.8. *When the voters' preferences over alternatives are $\mathcal{O}$-lexicographic, the communication complexity of $SSP_{\mathcal{O}}$ is $O(pn)$.*

Corollary 11.1.9. *When the voters' preferences over alternatives are $inv(\mathcal{O})$-legal, the communication complexity of $SSP_{\mathcal{O}}$ is $O(pn)$.*

[6] The communication complexity of a voting rule is the smallest number of bits that must be transmitted to compute the winner of that rule (i.e., taking the minimum across all correct protocols). See Conitzer and Sandholm (2005b).

11.1.4 The Winner is Sensitive to The Order over The Issues

In the definition of SSP, we simply fixed the order $\mathcal{O}$ to be $X_1 > X_2 > \cdots > X_p$. A question worth addressing is, to what extent is the outcome of SSP sensitive to the variation of the order $\mathcal{O}$? More precisely, given a profile P, let $\mathrm{PW}(P) = |\{\vec{d} \in \mathcal{X} \mid \vec{d} = SSP_{\mathcal{O}'}(P) \text{ for some order } \mathcal{O}'\}|$. $\mathrm{PW}(P)$ is the number of different alternatives that can be made SSP winners by choosing a particular order $\mathcal{O}'$. Then, for a given number of binary issues p, we look for the maximal value of $\mathrm{PW}(P)$, for all profiles P on $\mathcal{X} = D_1 \times \ldots \times D_p$; we denote this number by $\mathrm{MW}(p)$.

A first observation is that there are $p!$ different choices for $\mathcal{O}'$. Therefore, a trivial upper bound on $\mathrm{MW}(p)$ is $p!$. Since there are 2^p alternatives, the $p!$ upper bound is only interesting when $p! < 2^p$, that is, $p \leqslant 3$. Example 11.1.10 shows that when $p = 2$ or $p = 3$, this trivial upper bound is actually tight, i.e. $\mathrm{MW}(2) = 2!$ and $\mathrm{MW}(3) = 3!$: there exists a profile such that by changing the order over the issues, all $p!$ different alternatives can be made winners. Due to McGarvey's Theorem (see Lemma 2.2.3), any complete and asymmetric directed graph G over the alternatives corresponds to the majority graph of some profile (we recall that the majority graph of a profile P is the directed graph whose vertices are the alternatives and containing an edge from c to c' if and only if a majority of voters in P prefer c to c'). Therefore, in the example, we only show the majority graph instead of explicitly constructing the whole profile.

Example 11.1.10. *The majority graphs for $p = 2$ and $p = 3$ are shown in Figure 11.2. Let P (respectively, P') denote an arbitrary profile whose majority graph is the same as Figure 11.2(a) (respectively, Figure 11.2(b)). It is not hard to verify that $SSP_{X_1 > X_2}(P) = 00$ and $SSP_{X_2 > X_1}(P) = 01$. For P', the value of $SSP_{\mathcal{O}'}(P')$ for the six possible orders is shown on Table 11.1. Note that $2! = 2$ and $3! = 6$. It follows that when $p = 2$ or $p = 3$, there exists a profile for which the SSP winners* *w.r.t. different orders over the issues are all different from each other.*

Table 11.1: The SSP winners for P' w.r.t. different orders over the issues.

The order	$X_1 > X_2 > X_3$	$X_1 > X_3 > X_2$	$X_2 > X_1 > X_3$
SSP winner	010	011	001
The order	$X_2 > X_3 > X_1$	$X_3 > X_1 > X_2$	$X_3 > X_2 > X_1$
SSP winner	100	110	101

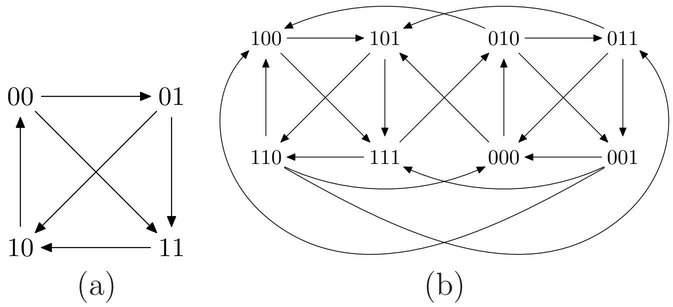


FIGURE 11.2: The majority graphs for $p = 2$ and $p = 3$.

In Figure 11.2, (a) is the majority graph for $p = 2$. (b) is the majority graph for $p = 3$, where four edges are not shown in the graph: 100 → 000, 101 → 001, 110 → 010, *and* 111 → 011. *The directions of the other edges are defined arbitrarily.* 000 *is the abbreviation for* $0_1 0_2 0_3$*, etc.*

When $p \geqslant 4$, $p! > 2^p$. However, it is not immediately clear whether $\text{MW}(p) = 2^p$ or not, i.e., whether each of the 2^p alternatives can be made a winner by changing the order over the issues. The next theorem shows that this can actually be done, that is, $\text{MW}(p) = 2^p$.

Theorem 11.1.11. *For any $p \geqslant 4$ and any $n \geqslant 142 + 4p$, there exists an n-profile P such that for every alternative $\vec{d}$, there exists an order $\mathcal{O}'$ over $\mathcal{I}$ such that $SSP_{\mathcal{O}'}(P) = \vec{d}$.*

Proof. We prove the theorem by induction on the number of issues p. Surprisingly, the hardest part in the inductive proof is the base case: when we first show how to construct a desirable majority graph $\mathcal{M}$ for $p = 4$, then we show how to construct a n-profile that corresponds to $\mathcal{M}$.

To define $\mathcal{M}$ when $p = 4$, we first define a majority graph $\mathcal{M}_3$ over $\mathcal{X}_3 = D_2 \times D_3 \times D_4$. Let $\mathcal{M}'$ denote the majority graph defined in Example 11.1.10 when $p = 3$. We note that $\mathcal{M}'$ is defined over $D_1 \times D_2 \times D_3$. The structure of $\mathcal{M}_3$ is exactly the same as $\mathcal{M}'$, except that $\mathcal{M}_3$ is defined over $D_2 \times D_3 \times D_4$. Formally, let $h_1 : D_1 \rightarrow D_2$ be a mapping such that $h_1(0_1) = 0_2$ and $h_1(1_1) = 1_2$; let $h_2 : D_2 \rightarrow D_3$ be a mapping such that $h_2(0_2) = 0_3$ and $h_2(1_2) = 1_3$; and let $h_3 : D_3 \rightarrow D_4$ be a mapping such that $h_3(0_3) = 0_4$ and $h_3(1_3) = 1_4$. Let $h : D_1 \times D_2 \times D_3 \rightarrow D_2 \times D_3 \times D_4$ be a mapping such that for any $(a_2, a_3, a_4) \in \{0, 1\}$, $h(a_1, a_2, a_3) = (h_1(a_1), h_2(a_2), h_3(a_3))$. For example, $h(0_1 1_2 0_3) = 0_2 1_3 0_4$. Then, we let $\mathcal{M}_3 = h(\mathcal{M}')$.

For any $\vec{a} = (a_2, a_3, a_4) \in \mathcal{X}_3$, let $f(\vec{a}) = (1_1, \vec{a})$ and let $g(\vec{a}) = (0_1, \overline{a_2}, a_3, a_4)$. That is, f concatenates 1_1 and $\vec{a}$, and g flips the first two components of $f(\vec{a})$. For example, $f(0_2 0_3 0_4) = 1_1 0_2 0_3 0_4$ and $g(0_2 0_3 0_4) = 1_1 1_2 0_3 0_4$. We define $\mathcal{M}$ as follows.

(1) The subgraph of $\mathcal{M}$ over $\{1_1\} \times \mathcal{X}_3$ is $f(\mathcal{M}_3)$. That is, for any $\vec{a}, \vec{b} \in \mathcal{X}_3$, if $\vec{a} \rightarrow \vec{b}$ in $\mathcal{M}'$, then $f(\vec{a}) \rightarrow f(\vec{b})$ in $\mathcal{M}$.

(2) The subgraph of $\mathcal{M}$ over $\{0_1\} \times \mathcal{X}_3$ is $g(\mathcal{M}_3)$.

(3) For any $\vec{a} \in \mathcal{X}_3$, we have $(1_1, \vec{a}) \rightarrow (0_1, \vec{a})$. For any $\vec{a} \in \mathcal{X}_3$ and $\vec{a} \neq 111$, we have $g(\vec{a}) \rightarrow f(\vec{a})$.

(4) We then add the following edges to $\mathcal{M}$. 0100 → 1110, 1000 → 0010, 1101 → 0111, 0001 → 1011, 1101 → 0100, 1000 → 0001, 0001 → 1101, 0100 → 1000, 1111 → 0110, 1100 → 0101, 0011 → 1010, 1001 → 0000, 1111 → 0011, 0011 → 1100, 0011 → 1001, 1111 → 0000.

(5) Any other edge that is not defined above is defined arbitrarily.

Let P be an arbitrary profile whose majority graph satisfies conditions (1) through (4) above. We make the following observations.

- If X_1 is the first issue in $\mathcal{O}'$, then the first component of $\text{SSP}_{\mathcal{O}'}(P)$ is 1_1. Moreover, every alternative whose first component is 1_1 (except 1111 and 1000) can be made to win by changing the order of X_2, X_3, X_4.

- If X_1 is the last issue in $\mathcal{O}'$, then the first component of $\text{SSP}_{\mathcal{O}'}(P)$ is 0_1. Moreover, every alternative whose first component is 0_1 (except 0011 and 0100) can be made to win by changing the order of X_2, X_3, X_4.

- Let $\mathcal{O}' = X_3 > X_1 > X_2 > X_4$, we have $\text{SSP}_{\mathcal{O}'}(P) = 0100$; let $\mathcal{O}' = X_3 > X_1 > X_4 > X_2$, we have $\text{SSP}_{\mathcal{O}'}(P) = 1000$; let $\mathcal{O}' = X_4 > X_1 > X_3 > X_2$, we have $\text{SSP}_{\mathcal{O}'}(P) = 0011$; let $\mathcal{O}' = X_2 > X_4 > X_1 > X_3$, we have $\text{SSP}_{\mathcal{O}'}(P) = 1111$.

In summary, every alternative is a winner of SSP w.r.t. at least one order over the issues. The reader can also check out the java program online at `http://www.cs.duke.edu/~lxia/Files/SSP.zip`, to verify the correctness of such a construction. We notice that conditions (1) through (4) impose 79 constraints on pairwise comparisons. Therefore, using McGarvey's trick (Lemma 2.2.3), for any $n \geqslant 2 \times 79 = 158$, we can construct an n-profile whose majority graph satisfies conditions (1) through (4). This means that the theorem holds for $p = 4$.

Now, suppose that the theorem holds for $p = p'$. Let $P = (V_1, \ldots, V_n)$ be an n-profile over $\mathcal{X}' = D_2 \times \ldots \times D_{p'+1}$ such that $n \geqslant 142 + 4p'$ and each alternative in $\mathcal{X}'$ can be made to win in SSP by changing the order over $X_2, \ldots, X_{p'+1}$. Let $\mathcal{X} = D_1 \times \cdots \times D_{p'+1}$. Let $f : \mathcal{X}' \to \mathcal{X}$ be the mapping defined as follows. For any $\vec{a} \in \mathcal{X}'$, $f(\vec{a}) = (1_1, \vec{a})$. That is, for any $\vec{a} \in \mathcal{X}'$, f concatenates 1_1 and $\vec{a}$. Let $g : \mathcal{X}' \to \mathcal{X}$ be the mapping defined as follows. For any $\vec{a} = (a_2, \ldots, a_{p'+1}) \in \mathcal{X}'$,

$g(\vec{a}) = (0_1, \overline{a_2}, a_3, \ldots, a_{p'+1})$. That is, for any $\vec{a} \in \mathcal{X}'$, g flips the first two components of $f(\vec{a})$. Next, we define an $(n+4)$-profile $P' = (V'_1, \ldots, V'_{n+4})$ as follows.

For any $i \leqslant 2\lfloor (n-1)/2 \rfloor$, we let $V'_i = \begin{cases} f(V_i) \succ g(V_i) & \text{if } i \text{ is odd} \\ g(V_i) \succ f(V_i) & \text{if } i \text{ is even} \end{cases}$. For any $2\lfloor (n-1)/2 \rfloor + 1 \leqslant i \leqslant n$, we let $V'_i = [f(V_i) \succ g(V_i)]$. For any $j \leqslant 4$, we let

$$V'_{n+j} = \begin{cases} \begin{array}{c} g(0_2 \ldots 0_{p+1}) \succ f(0_2 \ldots 0_{p+1}) \succ g(0_2 \ldots 0_p 1_{p+1}) \\ \succ f(0_2 \ldots 0_p 1_{p+1}) \succ g(1_2 \ldots 1_{p+1}) \succ f(1_2 \ldots 1_{p+1}) \end{array} & \text{if } j \text{ is odd} \\ \begin{array}{c} g(1_2 \ldots 1_{p+1}) \succ f(1_2 \ldots 1_{p+1}) \succ g(1_2 \ldots 1_p 0_{p+1}) \\ \succ f(1_2 \ldots 1_p 0_{p+1}) \succ g(0_2 \ldots 0_{p+1}) \succ f(0_2 \ldots 0_{p+1}) \end{array} & \text{if } j \text{ is even} \end{cases}$$

For any pair of alternatives c, c', and any profile P^*, we let $D_{P^*}(c, c')$ denote the number of times that c is preferred to c', minus the number of times c' is preferred to c, both in the profile P^*. That is, $D_{P^*}(c, c') > 0$ if and only if c beats c' in their pairwise election. We make the following observations on P'.

- For any $\vec{a} \in \mathcal{X}'$, $D_{P'}(f(\vec{a}), g(\vec{a})) > 0$ and $D_{P'}((1_1, \vec{a}), (0_1, \vec{a})) > 0$.
- For any $\vec{a}, \vec{b} \in \mathcal{X}'$ (with $\vec{a} \neq \vec{b}$), $D_{P'}(f(\vec{a}), f(\vec{b})) > 0$ if and only if $D_P(\vec{a}, \vec{b}) > 0$; $D_{P'}(g(\vec{a}), g(\vec{b})) > 0$ if and only if $D_P(\vec{a}, \vec{b}) > 0$.

It follows that for any order $\mathcal{O}'$ over $\{X_2, \ldots, X_{p+1}\}$, we have $\text{SSP}_{[X_1 > \mathcal{O}']}(P') = f(SSP_{\mathcal{O}'}(P))$ (because after voting on issue X_1, all alternatives whose first component is 0_1 are eliminated, then it reduces to SSP over $\mathcal{X}'$); we also have that $\text{SSP}_{[\mathcal{O}' > X_1]}(P') = g(SSP_{\mathcal{O}'}(P))$ (because in the last round, the two competing alternatives are considering are $f(SSP_{\mathcal{O}'}(P))$ and $g(SSP_{\mathcal{O}'}(P))$, and the majority of voters prefer the latter). We recall that each alternative in $\mathcal{X}'$ can be made to win w.r.t. an order O' over $\{X_2, \ldots, X_{p'+1}\}$. It follows that each alternative in $\mathcal{X}$ can also be made to win w.r.t. an order over $\{X_1, \ldots, X_{p'+1}\}$, which means that the theorem holds for $p = p' + 1$. Therefore, the theorem holds for any $p \geqslant 4$. □

11.2 Minimax Satisfaction Index

In the rest of this chapter, we will show that strategic sequential voting on multi-issue domains is prone to paradoxes that are almost as severe as previously studied multiple-election paradoxes under models that are not game-theoretic (Brams et al., 1998; Lacy and Niou, 2000).[7] To facilitate the presentation of these results, we define an index that is intended to measure one aspect of the quality of a voting rule, called the *minimax satisfaction index*.

Definition 11.2.1. *For any voting rule r, the* minimax satisfaction index (MSI) *of r is defined as*

$$MSI_r(m,n) = \min_{P \in L(\mathcal{X})^n} \max_{V \in P} \big(m + 1 - rank_V(r(P))\big)$$

where m is the number of alternatives, n is the number of voters, and $rank_V(r(P))$ is the position of $r(P)$ in vote V.

We note that in this chapter $m = 2^p$, where p is the number of issues. The MSI of a voting rule is not the final word on it. For example, the MSI for dictatorships is m, the maximum possible value, which is not to say that dictatorships are desirable. However, if the MSI of a voting rule is low, then this implies the existence of a paradox for it, namely, a profile that results in a winner that makes all voters unhappy.

The third type of multiple-election paradoxes (see Section 8.1) implicitly refer to such an index. We recall that the third type of multiple-election paradoxes state that if voters vote on issues separately and optimistically, then there exists a profile such that in each vote, the winner is ranked near the bottom; therefore this rule has a very low MSI.

[7] Even though Lacy and Niou (2000) have studied SSP, they actually did not examine whether there are any multiple-election paradoxes in SSP.

11.3 Multiple-Election Paradoxes for Strategic Sequential Voting

In this section, we show that over multi-binary-issue domains, for any natural number n that is sufficiently large (we will specify the number in our theorems), there exists an n-profile P such that $SSP_{\mathcal{O}}(P)$ is ranked almost in the bottom position in each vote in P. That is, the minimax satisfaction index is extremely low for the strategic sequential voting procedure.

We first calculate the MSI for $SSP_{\mathcal{O}}$ when the winner does not depend on the tie-breaking mechanism. That is, either n is odd, or n is even and there is never a tie in any stage of running the election sequentially. This is our main multiple-election paradox result.

Theorem 11.3.1. *For any $p \in \mathbb{N}$ ($p \geqslant 2$) and any $n \geqslant 2p^2 + 1$, $MSI_{SSP_{\mathcal{O}}}(m, n) = \lfloor p/2 + 2 \rfloor$.*[8] *Moreover, in the profile P that we use to prove the upper bound, the winner $SSP_{\mathcal{O}}(P)$ is Pareto-dominated by $2^p - (p+1)p/2$ alternatives.*

Proof of Theorem 11.3.1: The upper bound on $MSI_{SSP_{\mathcal{O}}}(m, n)$ is constructive, that is, we explicitly construct a paradox.

For any n-profile $P = (V_1, \ldots, V_n)$, we define the mapping $f_P : \mathcal{X} \to \mathbb{N}^n$ as follows: for any $c \in \mathcal{X}$, $f_P(c) = (h_1, \ldots, h_n)$ such that for any $i \leqslant n$, h_i is the number of alternatives that are ranked below c in V_i. For any $l \leqslant p$, we denote $\mathcal{X}_l = D_l \times \cdots \times D_p$ and $\mathcal{O}_l = X_l > X_{l+1} > \cdots > X_p$. For any vector $\vec{h} = (h_1, \ldots, h_n)$ and any $l \leqslant p$, we say that $\vec{h}$ is *realizable* over $\mathcal{X}_l$ (through a balanced binary tree) if there exists a profile $P_l = (V_1, \ldots, V_n)$ over $\mathcal{X}_l$ such that $f_{P_l}(SSP_{\mathcal{O}_l}(P_l)) = \vec{h}$. We first prove the following lemma.

[8] If n is even, then to prove $MSI_{SSP_{\mathcal{O}}}(m, n) \geqslant \lfloor p/2 + 2 \rfloor$, we restrict attention to profiles without ties.

Lemma 11.3.2. *For any l such that $1 \leqslant l < p$,*

$$\vec{h}_* = (\underbrace{0, \ldots, 0}_{\lfloor n/2 \rfloor - p + l}, \underbrace{1, \ldots, 1}_{p-l+1}, \underbrace{2^{p-l+1} - 1, \ldots, 2^{p-l+1} - 1}_{\lceil n/2 \rceil - 1})$$

is realizable over $\mathcal{X}_l$.

Proof of Lemma 11.3.2: We prove that there exists an n-profile P_l over $\mathcal{X}_l$ such that $SSP_{\mathcal{O}_l}(P_l) = 1_l \cdots 1_p$ and $\vec{h}_*$ is realized by P_l. For any $1 \leqslant i \leqslant p - l + 1$, we let $\vec{b}_i = 1_l \cdots 1_{p-i} 0_{p+1-i} 1_{p+2-i} \cdots 1_p$. That is, $\vec{b}_i$ is obtained from $1_l \cdots 1_p$ by flipping the value of X_{p+1-i}. We obtain $P_l = (V_1, \ldots, V_n)$ in the following steps.

1. Let $W_1, \ldots, W_n$ be null partial orders over $\mathcal{X}_l$. That is, for any $i \leqslant n$, the preference relation W_i is empty.

2. For any $j \leqslant \lfloor n/2 \rfloor - p + l$, we put $1_l \cdots 1_p$ in the bottom position in W_j; we put $\{\vec{b}_1, \ldots, \vec{b}_{p-l+1}\}$ in the top positions in W_j.

3. For any j with $\lfloor n/2 \rfloor + 2 \leqslant j \leqslant n$, we put $1_l \cdots 1_p$ in the top position of W_j, and we put $\{\vec{b}_1, \ldots, \vec{b}_{p-l+1}\}$ in the positions directly below the top.

4. For j with $\lfloor n/2 \rfloor - p + l + 1 \leqslant j \leqslant \lfloor n/2 \rfloor + 1$, we define preferences as follows. For any $i \leqslant p - l + 1$, in $W_{\lfloor n/2 \rfloor - p + l + i}$, we put $\vec{b}_i$ in the bottom position, $1_l \cdots 1_p$ in the second position from the bottom, and all the remaining b_j (with $j \neq i$) at the very top.

5. Finally, we complete the profile arbitrarily: for any $j \leqslant n$, we let V_j be an arbitrary extension of W_j.

Let $P_l = (V_1, \ldots, V_n)$. We note that for any $i \leqslant p - l + 1$, $\vec{b}_i$ beats any alternative in $\mathcal{X}_l \backslash \{1_l \cdots 1_p, \vec{b}_1, \ldots, \vec{b}_{p-l+1}\}$ in pairwise elections. Therefore, for any $i \leqslant p - l + 1$, the ith alternative that meets $1_l \cdots 1_p$ is $\vec{b}_i$, which loses to $1_l \cdots 1_p$ (just barely). It follows that $1_l \cdots 1_p$ is the winner, and it is easy to check that $f_{P_l}(1_l \cdots 1_p) = \vec{h}_*$. This completes the proof of the lemma. □

Because the majority rule is anonymous, for any permutation π over $1,\ldots,n$ and any $l<p$, if $(h_1,\ldots,h_n)$ is realizable over $\mathcal{X}_l$, then $(h_{\pi(1)},\ldots,h_{\pi(n)})$ is also realizable over $\mathcal{X}_l$. For any $k\in\mathbb{N}$, we define $H_k=\{\vec{h}\in\{0,1\}^n:\sum_{j\leqslant n}h_j\geqslant k\}$. That is, H_k is composed of all n-dimensional binary vectors in each of which at least k components are 1. We next show a lemma to derive a realizable vector over $\mathcal{X}_{l-1}$ from two realizable vectors over $\mathcal{X}_l$.

Lemma 11.3.3. *Let $l<p$, and let $\vec{h}_1$, $\vec{h}_2$ be vectors that are realizable over $\mathcal{X}_l$. For any $\vec{h}\in H_{\lfloor n/2\rfloor+1}$, $\vec{h}_1+(\vec{h}_2+\vec{1})\cdot\vec{h}$ is realizable over $\mathcal{X}_{l-1}$, where $\vec{1}=(1,\ldots,1)$, and for any $\vec{a}=(a_1,\ldots,a_n)$ and any $\vec{b}=(b_1,\ldots,b_n)$, we have $\vec{a}\cdot\vec{b}=(a_1b_1,\ldots,a_nb_n)$.*

Proof of Lemma 11.3.3: Without loss of generality, we prove the lemma for $\vec{h}=(\underbrace{0,\ldots,0}_{\lceil n/2\rceil-1},\underbrace{1,\ldots,1}_{\lfloor n/2\rfloor+1})$. Let P_1,P_2 be two profiles over $\mathcal{X}_l$, each of which is composed of n votes, such that $f(P_1)=\vec{h}_1$ and $f(P_2)=\vec{h}_2$. Let $P_1=(V_1^1,\ldots,V_n^1)$, $P_2=(V_1^2,\ldots,V_n^2)$, $\vec{a}=SSP_{\mathcal{O}_l}(P_1)$, $\vec{b}=SSP_{\mathcal{O}_l}(P_2)$. We define a profile $P=(V_1,\ldots,V_n)$ over $\mathcal{X}_{l-1}$ as follows.

1. Let $W_1,\ldots,W_n$ be n null partial orders over $\mathcal{X}_{l-1}$.

2. For any $j\leqslant n$ and any $\vec{e}_1,\vec{e}_2\in\mathcal{X}_l$, we let $(1_{l-1},\vec{e}_1)\succ_{W_j}(1_{l-1},\vec{e}_2)$ if $\vec{e}_1\succ_{V_j^1}\vec{e}_2$; and we let $(0_{l-1},\vec{e}_1)\succ_{W_j}(0_{l-1},\vec{e}_2)$ if $\vec{e}_1\succ_{V_j^2}\vec{e}_2$.

3. For any $\lceil n/2\rceil\leqslant j\leqslant n$, we let $(1_{l-1},\vec{a})\succ_{W_j}(0_{l-1},\vec{b})$.

4. Finally, we complete the profile arbitrarily: for any $j\leqslant n$, we let V_j be an (arbitrary) extension of W_j such that $(1_{l-1},\vec{a})$ is ranked as low as possible.

We note that $(1_{l-1},\vec{a})$ is the winner of the subtree in which $X_{l-1}=1_{l-1}$, $(0_{l-1},\vec{b})$ is the winner of the subtree in which $X_{l-1}=0_{l_1}$, and $(1_{l-1},\vec{a})$ beats $(0_{l-1},\vec{b})$ in their pairwise election (because the votes from $\lceil n/2\rceil$ to n rank $(1_{l-1},\vec{a})$ above $(0_{l-1},\vec{b})$). Therefore, $SSP_{\mathcal{O}_{l-1}}(P)=(1_{l-1},\vec{a})$.

Finally, we have that $f_P((1_{l-1}, \vec{a})) = \vec{h}_1 + (\vec{h}_2 + \vec{1}) \cdot \vec{h}$. This is because $(1_{l-1}, \vec{a})$ is ranked just as low as in the profile P_1 for voters 1 through $\lceil n/2 \rceil - 1$; for any voter j with $\lceil n/2 \rceil \leqslant j \leqslant n$, additionally, $(0_{l-1}, \vec{b})$ needs to be placed below $(1_{l-1}, \vec{a})$, which implies that also, all the alternatives $(0_{l-1}, \vec{b}')$ for which j ranked $\vec{b}'$ below $\vec{b}$ in P_2 must be below $(1_{l-1}, \vec{a})$ in j's new vote in P. This completes the proof of the lemma. □

Now we are ready to prove the main part of the theorem. It suffices to prove that for any $n \geqslant 2p^2 + 1$, there exists a vector $\vec{h}_p \in \mathbb{N}^n$ such that each component of $\vec{h}_p$ is no more than $\lfloor p/2 + 1 \rfloor$, and $\vec{h}_p$ is realizable over $\mathcal{X}$. We first prove the theorem for the case in which n is odd. We show the construction by induction in the proof of the following lemma.

Lemma 11.3.4. *Let n be odd. For any $l' < p$ (such that l' is odd),*

$$\vec{h}_{l'} = (\underbrace{\lfloor l'/2 \rfloor, \ldots, \lfloor l'/2 \rfloor}_{\lceil n/2 \rceil - (l'^2+1)/2}, \underbrace{\lceil l'/2 \rceil, \ldots, \lceil l'/2 \rceil}_{\lfloor n/2 \rfloor + (l'^2+1)/2})$$

is realizable over $\mathcal{X}_{p-l'+1}$, and if $l' < p$, then

$$\vec{h}_{l'+1} = (\underbrace{\lfloor l'/2 \rfloor, \ldots, \lfloor l'/2 \rfloor}_{l'+1}, \underbrace{\lceil l'/2 \rceil, \ldots, \lceil l'/2 \rceil}_{n-(l'+5)(l'+1)/2}, \underbrace{\lceil l'/2 \rceil + 1, \ldots, \lceil l'/2 \rceil + 1}_{(l'+3)(l'+1)/2})$$

is realizable over $\mathcal{X}_{p-l'}$.

Proof of Lemma 11.3.4: The base case in which $l' = 1$ corresponds to a single-issue majority election over two alternatives, where $\lceil n/2 \rceil - 1$ voters vote for one alternative, and $\lfloor n/2 \rfloor + 1$ vote for the other, so that only the latter get their preferred alternative.

Now, suppose the claim holds for some $l' \leqslant p - 2$; we next show that the claim also holds for $l' + 2$. To this end, we apply Lemma 11.3.3 twice. Let $l = p - l' + 1$.

First, let $\vec{h}_* = (\underbrace{1, \ldots, 1}_{l'}, \underbrace{2^{l'} - 1, \ldots, 2^{l'} - 1}_{l'+1}, \underbrace{0, \ldots, 0}_{\lfloor n/2 \rfloor - l' + 1}, \underbrace{2^{l'} - 1, \ldots, 2^{l'} - 1}_{\lceil n/2 \rceil - l' - 2})$

By Lemma 11.3.2, $\vec{h}_*$ is realizable over $\mathcal{X}_l$ (via a permutation of the voters). Let $\vec{h} = (\underbrace{1,\ldots,1}_{l'}, \underbrace{0,\ldots,0}_{l'+1}, \underbrace{1,\ldots,1}_{\lfloor n/2\rfloor - l'+1}, \underbrace{0,\ldots,0}_{\lceil n/2\rceil - l'-2})$.

Then, by Lemma 11.3.3, $\vec{h}_{l'} + (\vec{h}_* + \vec{1}) \cdot \vec{h}$ is realizable over $\mathcal{X}_{l-1}$. We have the following calculation.

$$\begin{aligned}
&\vec{h}_{l'} + (\vec{h}_* + \vec{1}) \cdot \vec{h} \\
=&(\underbrace{\lceil l'/2\rceil + 1, \ldots, \lceil l'/2\rceil + 1}_{l'}, \\
&\underbrace{\lfloor l'/2\rfloor, \ldots, \lfloor l'/2\rfloor}_{l'+1}, \underbrace{\lceil l'/2\rceil, \ldots, \lceil l'/2\rceil}_{\lceil n/2\rceil - (l'+3)(l'+1)/2}, \\
&\underbrace{\lceil l'/2\rceil + 1, \ldots, \lceil l'/2\rceil + 1}_{(l'+1)^2/2+1}, \underbrace{\lceil l'/2\rceil, \ldots, \lceil l'/2\rceil}_{\lfloor n/2\rfloor - l'-1})
\end{aligned}$$

The partition of the set of voters into these five groups uses the fact that $n \geqslant 2p^2 + 1$ implies $\lceil n/2\rceil - (l' + 3)(l' + 1)/2 \geqslant 0$. After permuting the voters in this vector, we obtain the following vector which is realizable over $\mathcal{X}_{l-1}$:

$$\vec{h}_{l'+1} =(\underbrace{\lfloor l'/2\rfloor, \ldots, \lfloor l'/2\rfloor}_{l'+1}, \underbrace{\lceil l'/2\rceil, \ldots, \lceil l'/2\rceil}_{n-(l'+5)(l'+1)/2}, \underbrace{\lceil l'/2\rceil + 1, \ldots, \lceil l'/2\rceil + 1}_{(l'+3)(l'+1)/2})$$

We next let $\vec{h}' = (\underbrace{1,\ldots,1}_{\lfloor n/2\rfloor+1}, \underbrace{0,\ldots,0}_{\lceil n/2\rceil-1})$ and

$$\vec{h}'_* = (\underbrace{1,\ldots,1}_{l'+1}, \underbrace{0,\ldots,0}_{\lfloor n/2\rfloor - l'}, \underbrace{2^{l'+1} - 1, \ldots, 2^{l'+1} - 1}_{\lceil n/2\rceil - 1})$$

By Lemma 11.3.2, the latter is realizable over $\mathcal{X}_{l-1}$. Thus, by Lemma 11.3.3, $\vec{h}_{l'+1} + (\vec{h}'_* + \vec{1}) \cdot \vec{h}'$ is realizable over $\mathcal{X}_{l-2}$. Through a permutation over the voters, we obtain the desired vector:

$$\vec{h}_{l'+2} = (\underbrace{\lfloor l'/2\rfloor + 1, \ldots, \lfloor l'/2\rfloor + 1}_{\lceil n/2\rceil - (l'+2)(l'+1)/2-1}, \underbrace{\lceil l'/2\rceil + 1, \ldots, \lceil l'/2\rceil + 1}_{\lfloor n/2\rfloor + (l'+2)(l'+1)/2+1})$$

which is realizable over $\mathcal{X}_{l-2}$. Therefore, the claim holds for $l'+2$. This completes the proof of the lemma. $\square$

If p is odd, from Lemma 11.3.4 we know that the theorem is true, by setting $l'=p$. If p is even, then we first set $l'=p-1$; then, the maximum component of $\vec{h}_{l'+1}$ is $\lceil l'/2\rceil+1=\lceil (p-1)/2\rceil+1=p/2+1$. Thus we have proved the upper bound in the theorem when n is odd.

When n is even, we have the following lemma (the proof is similar to the proof of Lemma 11.3.4, so we omitted its proof).

Lemma 11.3.5. *Let n be even. For any $l'<p$ (such that l' is odd),*

$$\vec{h}_{l'}=(\underbrace{\lfloor l'/2\rfloor,\ldots,\lfloor l'/2\rfloor}_{n/2-(l'^2-l'+1)/2},\underbrace{\lceil l'/2\rceil,\ldots,\lceil l'/2\rceil}_{n/2+(l'^2-l'+1)/2})$$

is realizable over $\mathcal{X}_{p-l'+1}$, and if $l'+1\leqslant p$, then

$$\vec{h}_{l'+1}=(\underbrace{\lfloor l'/2\rfloor,\ldots,\lfloor l'/2\rfloor}_{l'+1},\underbrace{\lceil l'/2\rceil,\ldots,\lceil l'/2\rceil}_{n-1-(l'+4)(l'+1)/2},\underbrace{\lceil l'/2\rceil+1,\ldots,\lceil l'/2\rceil+1}_{(l'+2)(l'+1)/2+1})$$

is realizable over $\mathcal{X}_{p-l'}$.

The upper bound in the theorem when n is even follows from Lemma 11.3.5. Moreover, we note that in the step from l' to $l'+1$ (respectively, from $l'+1$ to $l'+2$), no more than l' new alternatives are ranked lower than the winner in the profile that realizes $\vec{h}_{l'+1}$ (respectively, $\vec{h}_{l'+2}$). It follows that in the profile that realizes $\vec{h}_{l'+1}$ (respectively, $\vec{h}_{l'+2}$) in Lemma 11.3.4 or Lemma 11.3.5, the number of alternatives that are ranked lower than the winner by at least one voter is no more than $(l'+1)l'/2+l'+1=(l'+1)(l'+2)/2$ (respectively, $(l'+2)(l'+3)/2$), which equals $(p+1)p/2$ if $l'+1=p$ (respectively, $(p+1)p/2$ if $l'+2=p$). Therefore, in the profile that we use to obtain the upper bound, the winner under $SSP_{\mathcal{O}}$ is Pareto-dominated by $2^p-(p+1)p/2$ alternatives.

Finally, we show that $\lfloor p/2+2 \rfloor$ is a lower bound on $MSI_{SSP_{\mathcal{O}}}(m,n)$. Let P be an n-profile; let $SSP_{\mathcal{O}}(P) = \vec{a}$, and let $\vec{b}_1, \ldots, \vec{b}_p$ be the alternatives that $\vec{a}$ defeats in pairwise elections in rounds $1, \ldots, p$. It follows that in round j, more than half of the voters prefer $\vec{a}$ to $\vec{b}_j$, because we assume that there are no ties in the election. Therefore, summing over all votes, there are at least $p \times (\lfloor n/2 \rfloor + 1)$ occasions where $\vec{a}$ is preferred to one of $\vec{b}_1, \ldots, \vec{b}_p$. It follows that there exists some $V \in P$ in which $\vec{a}$ is ranked higher than at least $\lceil p \times (\lfloor n/2 \rfloor + 1)/n \rceil \geqslant \lfloor p/2+1 \rfloor$ of the alternatives $\vec{b}_1, \ldots, \vec{b}_p$. Thus $MSI_{SSP_{\mathcal{O}}}(m,n) \geqslant \lfloor p/2+2 \rfloor$.

(End of proof for Theorem 11.3.1.) □

We note that the number of alternatives is $m = 2^p$. Therefore, $\lfloor p/2+2 \rfloor$ is exponentially smaller than the number of alternatives, which means that there exists a profile for which every voter ranks the winner very close to the bottom. Moreover, $(p+1)p/2$ is still exponentially smaller than 2^p, which means that the winner is Pareto-dominated by almost every other alternative.

Naturally, we wish to avoid such paradoxes. One may wonder whether the paradox occurs only if the ordering of the issues is particularly unfortunate with respect to the preferences of the voters. If not, then, for example, perhaps a good approach is to randomly choose the ordering of the issues.[9] Unfortunately, our next result shows that we can construct a single profile that results in a paradox for *all* orderings of the issues. While it works for all orders, the result is otherwise somewhat weaker than Theorem 11.3.1: it does not show a Pareto-dominance result, it requires a number of voters that is at least twice the number of alternatives, the upper bound shown on the MSI is slightly higher than in Theorem 11.3.1, and unlike Theorem 11.3.1, no matching lower bound is shown.

[9] Of course, for any ordering of the issues, there exists a profile that results in the paradoxes in Theorem 11.3.1; but this does not directly imply that there exists a single profile that works for all orderings over the issues.

Theorem 11.3.6. *For any $p, n \in \mathbb{N}$ (with $p \geqslant 2$ and $n \geqslant 2^{p+1}$), there exists an n-profile P such that for any order $\mathcal{O}$ over $\{X_1, \ldots, X_p\}$, $SSP_{\mathcal{O}}(P) = 1_1 \cdots 1_p$, and any $V \in P$ ranks $1_1 \cdots 1_p$ somewhere in the bottom $p + 2$ positions.*

Proof of Theorem 11.3.6: We first prove a lemma.

Lemma 11.3.7. *For any $c \in \mathcal{X}$, $\mathcal{X}' \subset \mathcal{X}$ such that $c \notin \mathcal{X}'$, and any $n \in \mathbb{N}$ ($n \geqslant 2m = 2^{p+1}$), there exists an n-profile that satisfies the following conditions. Let $F = \mathcal{X} \backslash (\mathcal{X}' \cup \{c\})$.*

- *For any $c' \in \mathcal{X}'$, c defeats c' in their pairwise election.*
- *For any $c' \in \mathcal{X}'$ and $d \in F$, c' defeats d in their pairwise election.*
- *For any $V \in P$, c is ranked somewhere in the bottom $|\mathcal{X}'| + 2$ positions.*

Proof of Lemma 11.3.7: We let $P = (V_1, \ldots, V_n)$ be the profile defined as follows. Let $F_1, \ldots, F_{\lfloor n/2 \rfloor + 1}$ be a partition of F such that for any $j \leqslant \lfloor n/2 \rfloor + 1$, $|F_j| \leqslant \lceil 2m/n \rceil = 1$. For any $j \leqslant \lfloor n/2 \rfloor + 1$, we let $V_j = [(F \backslash F_j) \succ c \succ \mathcal{X}' \succ F_j]$. For any $\lfloor n/2 \rfloor + 2 \leqslant j \leqslant n$, we let $V_j = [\mathcal{X}' \succ F \succ c]$. It is easy to check that P satisfies all conditions in the lemma. □

Now, let $c = 1_1 \cdots 1_p$ and $\mathcal{X}' = \{0_1 1_2 \cdots 1_p, 1_1 0_2 1_3 \cdots 1_p, \ldots, 1_1 \cdots 1_{p-1} 0_p\}$. By Lemma 11.3.7, there exists a profile P such that c beats any alternative in $\mathcal{X}'$ in pairwise elections, any alternative in $\mathcal{X}'$ beats any alternative in $\mathcal{X} \backslash (\mathcal{X}' \cup \{c\})$ in pairwise elections, and c is ranked somewhere in the bottom $p + 2$ positions. This is the profile that we will use to prove the paradox.

Without loss of generality , we assume that $\mathcal{O} = X_1 > X_2 > \cdots > X_p$. (This is without loss of generality because all issues have been treated symmetrically so far.) c beats $1_1 \cdots 1_{p-1} 0_p$ in the first round; c will meet $1_1 \cdots 1_{p-2} 0_{p-1} 1_p$ in the next pairwise election, because $1_1 \cdots 1_{p-2} 0_{p-1} 1_p$ beats every other alternative in that branch (they are all in F), and c will win; and so on. It follows that $c = SSP_{\mathcal{O}}(P)$. Moreover, all voters rank c in the bottom $p+2$ positions.

(End of proof for Theorem 11.3.6.) □

11.4 Multiple-Election Paradoxes for SSP with Restrictions on Preferences

The paradoxes exhibited so far placed no restriction on the voters' preferences. While SSP is perfectly well defined for any preferences that the voters may have over the alternatives, we may yet wonder what happens if the voters' preferences over alternatives are restricted in a way that is natural with respect to the multi-issue structure of the setting. In particular, we may wonder if paradoxes are avoided by such restrictions. It is well known that natural restrictions on preferences sometimes lead to much more positive results in social choice and mechanism design—for example, single-peaked preferences allow for good strategy-proof mechanisms (Black, 1948; Moulin, 1980). In the next chapter we will see that we can characterize all strategy-proof voting rules when the voters' preferences are lexicographic, and their local preferences over issues are restricted.

In this section, we study the MSI for $SSP_{\mathcal{O}}$ for the following three cases: (1) voters' preferences are separable; (2) voters' preferences are $\mathcal{O}$-lexicographic; and (3) voters' preferences are $\mathcal{O}$-legal. For case (1), we show a mild paradox (and that this is effectively the strongest paradox that can be obtained); for case (2), we show a positive result; for case (3), we show a paradox that is nearly as bad as the unrestricted case.

Theorem 11.4.1. *For any $n \geqslant 2p$, when the profile is separable, the MSI for $SSP_{\mathcal{O}}$ is between $2^{\lceil p/2\rceil}$ and $2^{\lfloor p/2\rfloor+1}$.*

That is, the MSI of $SSP_{\mathcal{O}}$ when votes are separable is $\Theta(\sqrt{m})$. We still have that $\lim_{m\rightarrow\infty}\Theta(\sqrt{m})/m = 0$, so in that sense this is still a paradox. However, its convergence rate to 0 is much slower than for $\Theta(\log m)/m$, which corresponds to the convergence rate for the earlier paradoxes.

Proof of Theorem 11.4.1: Let $P = (V_1, \ldots, V_n)$. For any $i \leqslant p$, we let $d_i = maj(P|_{X_i})$. That is, d_i is the majority winner for the projection of the profile to the ith issue. Because any separable profile is compatible with any order over the issues, P is an $\mathcal{O}^{-1}$-legal profile. It follows from Corollary 11.1.6 that $SSP_{\mathcal{O}}(P) = (d_1, \ldots, d_p)$. Without loss of generality $(d_1, \ldots, d_p) = (1_1, \ldots, 1_p)$.

First, we prove the lower bound. Because for any $i \leqslant p$, at least half of the voters prefer 1_i to 0_i, the total number of times that a voter prefers 1 to 0 for an issue, counted across all voters and issues, is at least $p \cdot (n/2)$. Therefore, there exists $j \leqslant n$ such that voter j prefers 1 to 0 on at least $p/2$ issues, otherwise the total number of times that a voter prefers 1 to 0 for an issue, counted across all voters and issues, is no more than $n \cdot (p/2) - 1 < p \cdot (n/2)$, which is a contradiction. Formally put, there exists $j \leqslant n$ such that $|\{i \leqslant p : 1_i \succ_{V_j} 0_i\}| \geqslant p/2$. Without loss of generality for every $i \leqslant \lceil p/2 \rceil$, $1_i \succ_{V_j} 0_i$. It follows that for any $\vec{a} \in D_1 \times \cdots D_{\lceil p/2 \rceil}$, we have that $(1_1, \ldots, 1_p) \succ_{V_j} (\vec{a}, 1_{\lceil p/2 \rceil + 1}, \ldots, 1_p)$. Therefore, the minimax satisfaction index is at least $2^{\lceil p/2 \rceil}$.

Next, we prove the upper bound. We first show that there exists a set of n CP-nets $\mathcal{N}_1, \ldots, \mathcal{N}_n$ that satisfies the following two conditions.

1. For each $j \leqslant n$, the number of issues on which $\mathcal{N}_j$ prefers 1 to 0 is exactly $\lfloor p/2 \rfloor + 1$.
2. For each $i \leqslant p$, $maj(\mathcal{N}_1|_{X_i}, \ldots, \mathcal{N}_n|_{X_i}) = 1_i$.

The proof is by explicitly constructing the profile through the following n-step process. Informally, we will allocate $p(\lfloor n/2 \rfloor + 1)$ CPT entries "1 is preferred to 0", $\lfloor n/2 \rfloor + 1$ entries per issue, to n CP-nets as even as possible. Let $k_1 = \cdots = k_p = \lfloor n/2 \rfloor + 1$. In the jth step, we let $I_j = \{i_1, \ldots, i_{\lfloor p/2 \rfloor + 1}\}$ be the set of indices of the highest k's. Then, for any $i \in I_j$, we let $\mathcal{N}_j|_{X_i} = [1_i \succ 0_i]$ and $k_i \leftarrow k_i - 1$; for any $i \notin I_j$, we let $\mathcal{N}_j|_{X_i} = [0_i \succ 1_i]$. Because of the assumption that $n \geqslant 2p$, we have that $n(\lfloor p/2 \rfloor + 1) \geqslant p(\lfloor n/2 \rfloor + 1)$, which means that after n steps, for all $i \leqslant p$, $k_i \leqslant 0$.

It left to show that there exist extensions of $\mathcal{N}_1, \ldots, \mathcal{N}_n$ such that in each of these extensions, $1_1 \cdots 1_p$ is ranked within bottom $2^{\lfloor p/2 \rfloor + 1}$ positions. To show this, we use the following lemma.

Lemma 11.4.2. *For any partial order W and any alternative c, we let $|Down_W(c)| = \{c' : c \succeq_W c'\}$, that is, $|Down_W(c)|$ is the set of all alternatives (including c) that are less preferred to c in W. There exists a linear order V such that V extends W and c is ranked in the $|Down_W(c)|$th position from the bottom.*

The proof of Lemma 11.4.2 is quite straightforward: for every alternative d such that $d \notin \text{Down}_W(c)$, we put $d \succ c$ in the partial order. This does not violate transitivity, which means that the ordering relation obtained in this way is a partial order, denoted by W'. Then, let V be an arbitrary linear order that extends W'. It follows that c is ranked at the $\text{Down}_W(c)$th position from the bottom in V.

We note that for any $j \leqslant n$, the number of entries in $\mathcal{N}_j$ where $1 \succ 0$ is no more than $\lfloor p/2 + 1 \rfloor$. Therefore, for any $j \leqslant n$, $|\text{Down}_{\succ_{\mathcal{N}_j}}(1_1 \cdots 1_p)| \leqslant 2^{\lfloor p/2+1 \rfloor}$ (we recall that $\succ_{\mathcal{N}_j}$ is the partial order that $\mathcal{N}_j$ encodes). Let $V_1, \ldots, V_n$ be extensions of $\mathcal{N}_1, \ldots, \mathcal{N}_n$, respectively, where for all $j \leqslant n$, $1_1 \cdots 1_p$ is ranked as low as possible in any V_j. It follows from Lemma 11.4.2 that for any $j \leqslant n$, $1_1 \cdots 1_p$ is ranked in the $2^{\lfloor p/2+1 \rfloor}$th position from the bottom in V_j. This proves the upper bound.

(End of proof for Theorem 11.4.1.) □

Theorem 11.4.3. *For any $p \in \mathbb{N}$ ($p \geqslant 2$) and any $n \geqslant 5$, when the profile is $\mathcal{O}$-lexicographic, $MSI(SSP_{\mathcal{O}}) = 3 \cdot 2^{p-2} + 1$. Moreover, $SSP_{\mathcal{O}}(P)$ is ranked somewhere in the top 2^{p-1} positions in at least $n/2$ votes.*

Naturally $\lim_{m\to\infty}(3m/4+1)/m = 3/4$, so in that sense there is no paradox when votes are $\mathcal{O}$-lexicographic.

Proof of Theorem 11.4.3: The proof is for profiles without ties. The other cases can be proved similarly. Without loss of generality $SSP_{\mathcal{O}}(P) = 1_1 \cdots 1_p$ and for every $j \leqslant \lfloor n/2 \rfloor + 1$, $1_1 \succ_{V_j|X_1} 0_1$. It follows that in $V_1, \ldots, V_{\lfloor n/2 \rfloor+1}$, $1_1 \cdots 1_p$ is ranked within top $2^{p-1} = m/2$ positions. Because in at least $\lfloor n/2 \rfloor + 1$ votes $1_1 : 1_2 \succ 0_2$, there exists a vote $V \in P$ such that $1_1 \succ_{V|X_1} 0_1$ and $1_1 : 1_2 \succ_{V|X_2:1_1} 0_2$. It follows that $1_1 \cdots 1_p$ is ranked in the $(3 \cdot 2^{p-2} + 1)$th position from the bottom. This proves that when the profile is $\mathcal{O}$-lexicographic, $MSI(SSP_{\mathcal{O}}) \geqslant 3 \cdot 2^{p-2} + 1$.

We next prove that $3 \cdot 2^{p-2} + 1$ is also an upper bound. Consider the profile $P = (V_1, \ldots, V_n)$ defined as follows. For any $j \leqslant \lfloor n/2 \rfloor + 1$, $1_1 \succ_{V_j|X_1} 0_1$; for any $\lfloor n/2 \rfloor + 2 \leqslant j \leqslant n$, $1_2 \succ_{V_j|X_2:1_1} 0_2$; for $j = 1, 2$, $1_2 \succ_{V_j|X_2:1_1} 0_2$; for any $3 \leqslant j \leqslant n$ and any $3 \leqslant i \leqslant p$, $1_i \succ_{V_j|X_i:1_1\cdots 1_{i-1}} 0_j$; for any local preferences of any voter that is not defined above, let 0 be preferred to 1.

We note that for any $i \leqslant p$, more than $n/2$ votes in $P|_{X_i:1_1\cdots 1_{i-1}}$ prefer 1_i to 0_i, which means that $SSP_{\mathcal{O}}(P) = 1_1 \cdots 1_p$. It is easy to check that in any vote, $1_1 \cdots 1_p$ is ranked somewhere within the bottom $3 \cdot 2^{p-2} + 1$ positions.

(End of proof for Theorem 11.4.3.) □

Under the previous two restrictions (separability and $\mathcal{O}$-lexicographicity), $SSP_{\mathcal{O}}$ coincides with $Seq(Maj, \ldots, Maj)$ (by Corollary 11.1.6 and Proposition 11.1.4, respectively). Therefore, Theorems 11.4.1 and 11.4.3 also apply to sequential voting rules as well as issue-by-issue voting rules.

Finally, we study the MSI for $SSP_{\mathcal{O}}$ when the profile is $\mathcal{O}$-legal. Theorem 11.4.6 shows that it is nearly as bad as the unrestricted case (Theorem 11.3.1). The proof of Theorem 11.4.6 is the most involved proof in this chapter. The idea of the proof is similar to that of the proof for Theorem 11.3.1, but now we cannot apply Lemma 11.3.3, because $\mathcal{O}$-legality must be preserved. We start with a simpler result that shows the idea of the construction.

Theorem 11.4.4. *There exists a way to break ties in $SSP_{\mathcal{O}}$ such that the following is true. Let $SSP'_{\mathcal{O}}$ be the rule corresponding to $SSP_{\mathcal{O}}$ plus the tiebreaking mechanism. For any $p \in \mathbb{N}$, there exists an $\mathcal{O}$-legal profile that consists of two votes, such that in one of the two votes, no more than $\lceil p/2 \rceil$ alternatives are ranked lower than the winner $SSP'_{\mathcal{O}}(P)$; and in the other vote, no more than $\lfloor p/2 \rfloor$ alternatives are ranked lower than $SSP'_{\mathcal{O}}(P)$.*

Proof of Theorem 11.4.4: The proof is by induction on p. When $p = 2$, let the CPT of $\mathcal{N}_1$ be $0_1 \succ 1_1, 0_1 : 1_2 \succ 0_2, 1_1 : 1_2 \succ 0_2$; let the CPT of $\mathcal{N}_2$ be $1_1 \succ 0_1, 0_1 : 0_2 \succ 1_2, 1_1 : 0_2 \succ 1_2$; $V_1 = [0_1 1_2 \succ 0_1 0_2 \succ 1_1 1_2 \succ 1_1 0_2]$; $V_2 = [1_1 0_2 \succ 0_1 0_2 \succ 1_1 1_2 \succ 0_1 1_2]$. In the first step, ties are broken in favor of $1_1 1_2$. Given 1_1, ties are broken in favor of 1_2; given 0_1, ties are broken in favor of 1_2.

Suppose the claim is true for $p = l$. Next we construct $\mathcal{N}_1$ and $\mathcal{N}_2$ for $p = l + 1$. Let $\mathcal{N}'_1$, $\mathcal{N}'_2$, V'_1, V'_2 be the CP-nets and the votes for the case of $p = l$, where the multi-issue domain is $D_2 \times \cdots \times D_{l+1}$. Without loss of generality $|\text{Down}_{V_1}(1_2 \cdots 1_{l+1})| \leqslant \lceil l/2 \rceil$ and $|\text{Down}_{V_2}(1_2 \cdots 1_{l+1})| \leqslant \lfloor l/2 \rfloor$. We recall that for any vote V and any alternative c, $\text{Down}_V(c)$ (defined in Lemma 11.4.2) is the set of all alternatives that are ranked below c in V, including c. Let $\vec{e} \in D_2 \times \cdots \times D_{l+1}$ be an arbitrary alternative such that $1_2 \cdots 1_{l+1} \succ_{V_2} \vec{e}$. Such an $\vec{e}$ always exists, because if on the contrary $1_2 \cdots 1_{l+1}$ is in the bottom of V_2, it must be ranked higher than at least l other alternatives in V_1 to win the election, which contradicts the assumption that $|\text{Down}_{V_1}(1_2 \cdots 1_{l+1})| \leqslant \lceil l/2 \rceil$. We will explain later why we choose $\vec{e}$ in such a way.

Let $\mathcal{N}^*_1$ (respectively, $\mathcal{N}^*_2$) be the separable CP-net (we recall that a CP-net is separable if its graph has no edges) $D_2 \times \cdots \times D_{l+1}$ in which $\vec{e}$ is in the top (respectively, bottom) position. For $i = 1, 2$, we let $\mathcal{N}_i$ be a CP-net over $D_1 \times \cdots \times$

D_{l+1}, defined as follows:

- $0_1 \succ_{\mathcal{N}_i} 1_1$.
- The sub-CP-net of $\mathcal{N}_i$ restricted on $X_1 = 1_1$ is $\mathcal{N}_i'$;
- The sub-CP-net of $\mathcal{N}_i$ restricted on $X_1 = 0_1$ is $\mathcal{N}_i^*$;

Let V_1, V_2 be the extension of $\mathcal{N}_1$ and $\mathcal{N}_2$ respectively, that satisfy the following conditions:

- For any $\vec{b}, \vec{d} \in D_2 \times \cdots \times D_{l+1}$ such that $\vec{b} \neq \vec{e}$, and any $i = 1, 2$, we have $(0_1, \vec{b}) \succ_{V_i} (1_1, \vec{d})$. This condition can be satisfied, because we have $0_1 \succ_{\mathcal{N}_i} 1_1$.
- For any $\vec{b}, \vec{d} \in D_2 \times \cdots \times D_{l+1}$, and any $i = 1, 2$, we have that $(1_1, \vec{b}) \succ_{V_i} (1_1, \vec{d})$ if and only if $\vec{b} \succ_{V_i'} \vec{d}$. This condition says that if we focus on the order of the alternatives whose X_1 component is 1_1 in V_i, then it is the same as in V_i'.
- For any $\vec{d} \in D_2 \times \cdots \times D_{l+1}$, we have that $(0_1, \vec{e}) \succ_{V_1} (1_1, \vec{d})$.
- $(1_1, \ldots, 1_{l+1}) \succ_{V_2} (0_1, \vec{e}) \succ_{V_2} (1_1, \vec{e})$.

We let the tie-breaking mechanism be defined as follows: in the first step, ties are broken in favor of 1_1; in the subgame in which $X_1 = 1_1$, ties are broken in the same way as for the profile (V_1', V_2') (such that $1_2 \cdots 1_{l_1}$ is the winner for the profile); in the subgame in which $X_1 = 0_1$, ties are broken in such a way that $\vec{e}$ is the winner (because $\vec{e}$ is ranked in the top position in one vote, and in the bottom position in the other, there exists a tie-breaking mechanism under which $\vec{e}$ is the winner).

We note that $1_1 \cdots 1_p \succ_{V_1} \vec{d}$ if and only if $\vec{d} = (1_1, \vec{d}')$ for some $\vec{d}' \in D_2 \times \cdots \times D_{l+1}$ such that $1_2 \cdots 1_p \succ_{V_1'} \vec{d}'$. It follows that $|\text{Down}_{V_1}(1_1 \cdots 1_{l+1})| = |\text{Down}_{V_1'}(1_2 \cdots 1_{l+1})|$. We also note that $1_1 \cdots 1_{l+1} \succ_{V_2} \vec{b}$ if and only if $\vec{b} = (0_1, \vec{e})$ or $\vec{b} = (1_1, \vec{b}')$ for some

$\vec{b}' \in D_2 \times \cdots \times D_{l+1}$ such that $1_2 \cdots 1_p \succ_{V_2'} \vec{b}'$. It follows that $|\mathrm{Down}_{V_2}(1_1 \cdots 1_{l+1})| = |\mathrm{Down}_{V_2'}(1_2 \cdots 1_{l+1})| + 1$. Therefore, we have the following inequalities.

$$|\mathrm{Down}_{V_1}(1_1 \cdots 1_{l+1})| \leqslant \lfloor (l+1)/2 \rfloor$$

$$|\mathrm{Down}_{V_2}(1_1 \cdots 1_p)| \leqslant \lfloor l/2 \rfloor + 1 \leqslant \lceil (l+1)/2 \rceil$$

Here the trick to choose $\vec{e}$ such that $1_2 \cdots 1_{l+1} \succ_{V_2'} \vec{e}$ is crucial, because we force $0_1 \succ_{\mathcal{N}_2} 1_1$ and $1_1 \cdots 1_{l+1} \succ_{V_2} (0_1, \vec{e})$, which implies that $1_1 \cdots 1_{l+1} \succ_{V_2} (0_1, \vec{e}) \succ_{V_2} (1_1, \vec{e})$ (since V_2 extends $\mathcal{N}_2$). If we chose $\vec{e}$ such that $\vec{e} \succ_{V_2'} 1_2 \cdots 1_{l+1}$, then we would have that $|\mathrm{Down}_{V_2}(1_1 \cdots 1_{l+1})| = |\mathrm{Down}_{V_2'}(1_2 \cdots 1_{l+1})| + 2$, which does not prove the claim for $p = l+1$.

Next, we verify that $SSP_{\mathcal{O}}(V_1, V_2) = 1_1 \cdots 1_{l+1}$. We note that $(0_1, \vec{e}) \succ_{V_1} 1_1 \cdots 1_{l+1}$. Therefore, in the first step voter 1 will vote for 0_1. Meanwhile, $1_1 \cdots 1_{l+1} \succ_{V_2} (0_1, \vec{e})$, which means that in the first step voter 2 will vote for 1_1. Because ties are broken in favor of 1_1 in the first step, we will fix $X_1 = 1_1$. Then, in the following steps (step $2, \ldots, l+1$), $1_2, \ldots, 1_{l+1}$ will be the winners by induction hypothesis, which means that $SSP_{\mathcal{O}}(V_1, V_2) = 1_1 \cdots 1_{l+1}$.

Therefore, the claim is true for $p = l+1$. This means that the claim is true for any $p \in \mathbb{N}$.

Example 11.4.5. *Let us show an example of the above construction from $p = 2$ to $p = 3$. In $\mathcal{N}_1$, we have $0_1 \succ 1_1$, $1_1 : \mathcal{N}_1^*$, and $0_1 : \mathcal{N}_1'$, where $\mathcal{N}_1'$ is $0_2 \succ 1_2, 0_2 : 1_3 \succ 0_3, 1_2 : 1_3 \succ 0_3$. (We note that $\mathcal{N}_1'$ is defined over $D_2 \times D_3$.) V_1 restricted to 1_1 is $V_1' = [0_2 1_3 \succ 0_2 0_3 \succ 1_2 1_3 \succ 1_2 0_3]$ (which is, again, over $D_2 \times D_3$). Let $\vec{e} = 0_2 1_3$. Therefore, we have the following construction:*

$V_1 = 0_1 0_2 1_3 \succ 0_1 1_2 1_3 \succ 0_1 0_2 0_3 \succ 0_1 1_2 0_3 \succ 1_1 0_2 1_3 \succ 1_1 0_2 0_3 \succ 1_1 1_2 1_3 \succ 1_1 1_2 0_3$

$V_2 = 0_1 1_2 0_3 \succ 0_1 0_2 0_3 \succ 0_1 1_2 1_3 \succ 1_1 1_2 0_3 \succ 1_1 0_2 0_3 \succ 1_1 1_2 1_3 \succ 0_1 0_2 1_3 \succ 1_1 0_2 1_3$

Ties are broken in a way such that if we are in the branch in which $X_1 = 1_1$, then $1_2 1_3$ is the winner; and if we are in the branch in which $X_1 = 0_1$, then $\vec{e} = 0_2 1_3$

is the winner. In the first step, ties are broken in favor of 1_1*. Then, the sub-game winners are* $1_1 1_2 1_3$ *and* $0_1 0_2 1_3$*. Since exactly one vote (*V_1*) prefers* $0_1 0_2 1_3$ *to* $1_1 1_2 1_3$*, and the other vote* V_2 *prefers* $1_1 1_2 1_3$ *to* $0_1 0_2 1_3$*, the winner is* $1_1 1_2 1_3$*.*

(End of proof for Theorem 11.4.4.) □

We emphasize that, unlike any of our other results, Theorem 11.4.4 is based on a specific tie-breaking mechanism. The next theorem concerns the more general and complicated case in which n can be either odd or even, and the winner does not depend on the tie-breaking mechanism. That is, there are no ties in the election. The situation is almost the same as in Theorem 11.3.1.

Theorem 11.4.6. *For any* $p, n \in \mathbb{N}$ *with* $n \geqslant 2p^2 + 2p + 1$*, there exists an* $\mathcal{O}$*-legal profile such that in each vote, no more than* $\lceil p/2 \rceil + 4$ *alternatives are ranked lower than* $SSP_{\mathcal{O}}(P)$*. Moreover,* $SSP_{\mathcal{O}}(P)$ *is Pareto-dominated by at least* $2^p - 4p^2$ *alternatives.*

Of course, the lower bound on the MSI from Theorem 11.3.1 still applies when the profile is $\mathcal{O}$-legal, so together with Theorem 11.4.6 this proves that the MSI for $SSP_{\mathcal{O}}$ when the profile is $\mathcal{O}$-legal is $\Theta(\log m)$, just as in the unrestricted case.

Proof of Theorem 11.4.6: For simplicity, we prove the theorem for the case in which $n = 2p^2 + 2p + 1$. The proof for the case in which $n > 2p^2 + 2p + 1$ is similar. For any $l \leqslant p$, we let $\mathcal{X}_l = \{0_l, 1_l\} \times \{0_{l+1}, 1_{l+1}\} \times \cdots \times \{0_p, 1_p\}$; let $\mathcal{O}_l = X_l > X_{l+1} > \cdots > X_p$. We first prove the following claim by induction.

Claim 11.4.1. *For any* $l \leqslant p$*, there exists a* $\mathcal{O}_l$*-legal profile* $P_l = A_l \cup B_l \cup \hat{A}_l \cup \hat{B}_l \cup \{c^l\}$ *over* $\mathcal{X}_l$*, where* $A_l = \{a^l_1, \ldots, a^l_{p^2}\}$*,* $B_l = \{b^l_1, \ldots, b^l_{p^2}\}$*,* $\hat{A}_l = \{\hat{a}^l_1, \ldots, \hat{a}^l_p\}$*,* $\hat{B}_l = \{\hat{b}^l_1, \ldots, \hat{b}^l_p\}$*, that satisfies the following conditions.*

- $SSP_{\mathcal{O}_l}(P_l) = 1_l \cdots 1_p$*.*

- *For any* $V \in P_l$*,* $|Down_V(1_l \cdots 1_p)| \leqslant \lceil (p - l + 1)/2 \rceil + 5$*.*

- *For any* $(p-l)p \leqslant j \leqslant p^2$, $|Down_{a^l_j}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 3$, $|Down_{b^l_j}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 3$.

- *For any* $p-l \leqslant j \leqslant p$, $|Down_{\hat{a}^l_j}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 3$, $|Down_{\hat{b}^l_j}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 3$.

- *If* $p-l+1$ *is odd, then*

  - *for any* $V_B \in B$, $|Down_{V_B}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 4$*;*
  - *for any* $(p-l)p \leqslant j \leqslant p^2$, $|Down_{b^l_j}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 2$*;*
  - *and for any* $p-l \leqslant j \leqslant p$, $|Down_{\hat{b}^l_j}(1_l \cdots 1_p)| \leqslant \lceil (p-l+1)/2 \rceil + 2$.

- $1_l \cdots 1_p$ *is ranked higher than* $1_l \cdots 1_{p-2}0_{p-1}0_p$ *in all votes in* P_l*.*

Proof of Claim 11.4.1: We prove the claim by induction on l. When $l = p-1$, we let all votes in P_{p-1} be $1_{p-1}1_p \succ 1_{p-1}0_p \succ 0_{p-1}1_p \succ 0_{p-1}0_p$. It is easy to check that P_{p-1} satisfies all the conditions in the claim. Suppose the claim is true for $l \leqslant p$, we next prove that the claim is also true for $l-1$. We show the existence of P_{l-1} by construction for the following two cases.

Case 1: $p-l+1$ is even.

We let $\mathcal{N}^l_1, \ldots, \mathcal{N}^l_{p-l+1}, \mathcal{N}^l_A, \mathcal{N}^l_B$ be separable CP-nets over $\mathcal{X}_l$, defined as follows.

- Let $1_l \cdots 1_{p-2}0_{p-1}0_p$ be in the bottom position of $\mathcal{N}^l_A$; let $1_l \cdots 1_{p-2}0_{p-1}0_p$ be in the top position of $\mathcal{N}^l_B$.

- For any $1 \leqslant i \leqslant p-l-1$, let $1_l \cdots 1_{l+i-2}0_{l+i-1}1_{l+i} \cdots 1_{p-2}0_{p-1}0_p$ be in the top position of $\mathcal{N}^l_i$; let $1_l \cdots 1_{p-2}1_{p-1}0_p$ be in the top position of $\mathcal{N}^l_{p-l}$; let $1_l \cdots 1_{p-2}0_{p-1}1_p$ be in the top position of $\mathcal{N}^l_{p-l+1}$.

For any linear order V over $\mathcal{X}_l$, we let the *composition* of V and $\mathcal{N}$ (where $\mathcal{N} \in \{\mathcal{N}^l_1, \ldots, \mathcal{N}^l_{p-l+1}, \mathcal{N}^l_A, \mathcal{N}^l_B\}$) be a partial order O^{l-1} over $\mathcal{X}_{l-1}$, defined as follows.

- The restriction of O^{l-1} on $X_{l-1} = 1_{l-1}$ is V. That is, for any $\vec{d_1}, \vec{d_2} \in \mathcal{X}_l$ such that $\vec{d_1} \succ_V \vec{d_2}$, we let $(1_{l-1}, \vec{d_1}) \succ_{O^{l-1}} (1_{l-1}, \vec{d_2})$.

- The restriction of O^{l-1} on $X_{l-1} = 0_{l-1}$ is the partial order encoded by $\mathcal{N}$. That is, for any $\vec{d_1}, \vec{d_2} \in \mathcal{X}_l$ such that $\vec{d_1} \succ_{\mathcal{N}} \vec{d_2}$, we let $(0_{l-1}, \vec{d_1}) \succ_{O^{l-1}} (0_{l-1}, \vec{d_2})$.

- For any $\vec{d} \in \mathcal{X}_l$, we let $(0_{l-1}, \vec{d}) \succ_{O^{l-1}} (1_{l-1}, \vec{d})$.

- If $\mathcal{N} \in \{\mathcal{N}^l_1, \ldots, \mathcal{N}^l_{p-l+1}, \mathcal{N}^l_A\}$, we let $1_{l-1}1_l \cdots 1_p \succ_{O^{l-1}} 0_{l-1}1_l \cdots 1_{p-2}0_{p-1}0_p$.

We are now ready to define P_{l-1}. Any $V \in P_{l-1}$ has a counterpart in P_l. For example, the counterpart of $\hat{a}^{l-1}_1$ is $\hat{a}^l_1$. For any $V \in P_{l-1}$, V is defined to be the extension of the composition of V's counterpart in P_l and some $\mathcal{N}$ (where $\mathcal{N} \in \{\mathcal{N}^l_1, \ldots, \mathcal{N}^l_{p-l+1}, \mathcal{N}^l_A, \mathcal{N}^l_B\}$), in which $1_{l-1} \cdots 1_p$ is ranked as low as possible. Next we specify which $\mathcal{N}$ that each $V \in P_{l-1}$ corresponds to in the following table.

Table 11.2: From P_l to P_{l-1}.

for all	**votes in P_{l-1}**	**is composed of**	
$1 \leqslant j \leqslant p$	$\hat{a}^{l-1}_j$	$\hat{a}^l_j$	$\mathcal{N}^l_A$
$j \leqslant (p-l)p$	a^{l-1}_j	a^l_j	$\mathcal{N}^l_A$
$(p-l)p+1 \leqslant j \leqslant (p-l+1)p$	a^{l-1}_j	a^l_j	$\mathcal{N}^l_{j-(p-l)p}$
$(p-l+1)p+1 \leqslant j \leqslant p^2$	a^{l-1}_j	a^l_j	$\mathcal{N}^l_A$
	$\hat{b}^{l-1}_{p-l+2}$	$\hat{b}^l_{p-l+2}$	$\mathcal{N}^l_A$
$j \neq p-l+2$	$\hat{b}^{l-1}_j$	$\hat{b}^l_j$	$\mathcal{N}^l_B$
$j \leqslant p^2$	b^{l-1}_j	b^l_j	$\mathcal{N}^l_B$
	c^{l-1}	c^l	$\mathcal{N}^l_B$

It follows that P_{l-1} is $\mathcal{O}_{l-1}$-legal. By Lemma 11.4.2, we have the following calculation.

- For any $1 \leqslant j \leqslant p$, $|\text{Down}_{\hat{a}^{l-1}_j}(1_{l-1} \cdots 1_p)| = |\text{Down}_{\hat{a}^l_j}(1_l \cdots 1_p)| + 1$. This is because for any $\vec{d} \in \mathcal{X}_l$ such that $\vec{d} \in \text{Down}_{\hat{a}^l_j}(1_l \cdots 1_p)$, $1_{l-1} \cdots 1_p$ is ranked higher than $(1_{l-1}, \vec{d})$ in $\hat{a}_j^{l-1}$; and moreover, $1_{l-1} \cdots 1_p$ is ranked higher than $0_{l-1}1_l \cdots 1_{p-2}0_{p-1}0_p$ in $\hat{a}_j^{l-1}$.

- For any $1 \leqslant j \leqslant p$, $|\text{Down}_{a^{l-1}_{(p-l)p+j}}(1_{l-1} \cdots 1_p)| = |\text{Down}_{a^{l}_{(p-l)p+j}}(1_l \cdots 1_p)| + 3$. This is because for any $\vec{d} \in \mathcal{X}_l$ such that $\vec{d} \in \text{Down}_{a_{(p-l)p+j}}(1_l \cdots 1_p)$, $1_{l-1} \cdots 1_p$ is ranked higher than $(1_{l-1}, \vec{d})$ in $a_{(p-l)p+j}$; and moreover, $1_{l-1} \cdots 1_p$ is ranked higher than $0_{l-1}1_l \cdots 1_{p-2}0_{p-1}0_p$ in $a_{(p-l)p+j}$.

- For any $j \leqslant (p-l)p$ or $(p-l+1)p+1 \leqslant j \leqslant p^2$, $|\text{Down}_{a_j^{l-1}}(1_{l-1} \cdots 1_p)| = |\text{Down}_{a_j^{l}}(1_l \cdots 1_p)| + 1$.

- $|\text{Down}_{\hat{b}^{l-1}_{p-l+2}}(1_{l-1} \cdots 1_p)| = |\text{Down}_{\hat{b}^{l}_{p-l+2}}(1_l \cdots 1_p)| + 1$.

- For any $V_B \in (B_{l-1} \cup \hat{B}_{l-1} \cup \{c\}) \backslash \{\hat{b}_{p-l+2}\}$,
$$|\text{Down}_{V_B}(1_{l-1} \cdots 1_p)| = |\text{Down}_{V_B^l}(1_l \cdots 1_p)|,$$
where V_B^l is the counterpart of V_B in P_l.

We next prove that $SSP_{\mathcal{O}_{l-1}}(P_{l-1}) = 1_{l-1} \cdots l_p$. We note that $P_{l-1}|_{X_{l-1}=1_{l-1}} = P_l$. Therefore, if in the first step 1_{l-1} is chosen, then the winner is $1_{l-1} \cdots 1_p$. We also note that $P_{l-1}|_{X_{l-1}=0_{l-1}}$ is separable (and the CP-nets are $\mathcal{N}_1^l, \ldots, \mathcal{N}_{p-l+1}^l$, p^2+p copies of $\mathcal{N}_A^l$ and p^2+p copies of $\mathcal{N}_B^l$). Therefore, if in the first step 0_{l-1} is chosen, then the winner is $0_{l-1}1_l \cdots 1_{p-2}1_{p-1}1_p$. Because exactly p^2+p-1 votes in P_{l-1} prefer $0_{l-1}1_l \cdots 1_{p-2}1_{p-1}1_p$ to $1_{l-1} \cdots 1_p$ (those votes corresponds to $\mathcal{N}_B^l$ in the construction), we have that 1_{l-1} is the winner in the first step. Therefore, $SSP_{\mathcal{O}_{l-1}}(P_{l-1}) = 1_{l-1} \cdots l_p$. It is also easy to verify that P_{l-1} satisfies all conditions in the claim.

Case 2: $p-l+1$ is odd. The construction is similar as in the even case. The only difference is that we switch the role of A_l and B_l (also $\hat{A}_l$ and $\hat{B}_l$). □

The theorem follows from Claim 11.4.1 by letting $l = 1$, and it is easy to check that in P_1 in Claim 11.4.1 ($l = 1$), no more than $4p^2$ alternatives have been ranked lower than $SSP_{\mathcal{O}}(P_1)$ in any vote, which means that $SSP_{\mathcal{O}}(P_1)$ is Pareto-dominated by at least $2^p - 4p^2$ alternatives.

(End of proof for Theorem 11.4.6.) □

11.5 Summary

In this chapter, we considered a complete-information game-theoretic analysis of sequential voting on binary issues, which we called strategic sequential voting. Specifically, given that voters have complete information about each other's preferences and their preferences are strict, the game can be solved by a natural backward induction process (WSDSBI), which leads to a unique solution. We showed that under some conditions on the preferences, this process leads to the same outcome as the truthful sequential voting, but in general it can result in very different outcomes. We analyzed the effect of changing the order over the issues that voters vote on and showed that, in some elections, every alternative can be made a winner by voting according to an appropriate order over the issues.

Most significantly, we showed that strategic sequential voting is prone to multiple-election paradoxes; to do so, we introduced a concept called minimax satisfaction index, which measures the degree to which at least one voter is made happy by the outcome of the election. We showed that the minimax satisfaction index for strategic sequential voting is exponentially small, which means that there exists a profile for which the winner is ranked almost in the bottom positions in all votes; even worse, the winner is Pareto-dominated by almost every other alternative. We showed that changing the order of the issues in sequential voting cannot completely avoid the paradoxes. These negative results indicate that the solution of the sequential game can be extremely undesirable for every voter. We also showed that multiple-election paradoxes can be avoided to some extent by restricting voters' preferences to be separable or lexicographic, but the paradoxes still exist when the voters' preferences are $\mathcal{O}$-legal.

12

Strategy-Proof Voting Rules over Restricted Domains

We have seen in Chapter 7 and Chapter 11 that in some voting games, strategic behavior sometimes leads to extremely undesirable outcomes. Therefore, we may want to prevent voters' strategic behavior. However, the Gibbard-Satterthwaite theorem tells us that for any voting rule that satisfies some natural properties, there must exist at least one voter who has an incentive to misreport her preferences, if the voters are allowed to use any linear order to represent their preferences. To circumvent the Gibbard-Satterthwaite theorem, researchers in Computational Social Choice have investigated the possibility of using computational complexity to prevent voters' strategic behavior. Chapter 4 showed that for some common voting rules computational complexity can provide some protection from manipulation, while Chapter 5 gave some evidence that computational complexity does not seem to be a very strong barrier against manipulation.

In fact, there is another, older, line of research on circumventing the Gibbard-Satterthwaite theorem. This line, which has been pursued mainly by economists, is to restrict the domain of preferences. That is, we assume that voters' preferences always lie in a restricted class. An example of such a class is that of *single-peaked* preferences (Black, 1948). For single-peaked preferences, desirable strategy-proof rules exist, such as the *median* rule. Other strategy-proof rules are also possible in this preference domain: for example, it is possible to add some artificial (*phantom*) votes before running the median rule. In fact, this characterizes all strategy-proof rules for single-peaked preferences (Moulin, 1980). On the other hand, preferences have to be significantly restricted to obtain such positive results: Aswal et al. (2003) extend the Gibbard-Satterthwaite theorem, showing that if the preference domain is *linked*, then with three or more alternatives the only strategy-proof voting rule that satisfies non-imposition is a dictatorship.

In this chapter we will focus on exploring the possibility of using domain restriction to circumvent the Gibbard-Satterthwaite theorem in multi-issue domains. The problem of characterizing strategy-proof voting rules in multi-issue domains has already received significant attention. Strategy-proof voting rules for high-dimensional single-peaked preferences (where each dimension can be seen as an issue) have been characterized (Border and Jordan, 1983; Barbera et al., 1993, 1997; Nehring and Puppe, 2007). Barbera et al. (1991) characterized strategy-proof voting rules when the voters' preferences are separable, and each issue is binary (that is, the domain for each issue has two elements). Ju (2003) studied multi-issue domains where each issue can take three values: "good", "bad", and "null", and characterized all strategy-proof voting rules that satisfy *null-independence*, that is, if a voter votes "null" on an issue i, then her preferences over other issues do not affect the value of issue i.

The prior research that is closest to ours was performed by Le Breton and Sen (1999). They proved that if the voters' preferences are separable, and the restricted preference domain of the voters satisfies a *richness* condition, then, a voting rule is strategy-proof if and only if it is an issue-by-issue voting rule, in which each issue-wise voting rule is strategy-proof over its respective domain.

Despite its elegance, the work by Le Breton and Sen is limited by the restrictiveness of separable preferences: as we have argued in Chapter 8, in general, a voter's preferences on one issue depend on the decision taken on other issues. On the other hand, one would not necessarily expect the preferences for one issue to depend on every other issue. Therefore, it seems that sequential voting (Section 8.3) is better than issue-by-issue voting. While the assumption of sequential voting that there exists an ordering $\mathcal{O}$ over issues such that all voters' preferences are $\mathcal{O}$-legal is still restrictive, it is much less restrictive than assuming that preferences are separable. Chapter 9 and Chapter 10 concerned how to design new voting rules when voters use a much more expressive voting language (i.e., possibly cyclic CP-nets), but in this chapter, we only study the setting where all voters' preferences are $\mathcal{O}$-legal, and w.l.o.g. we fix $\mathcal{O} = X_1 > X_2 > \cdots > X_p$.

The main theorem of this chapter is the following: over *lexicographic* preference domains (where earlier issues dominate later issues in terms of importance to the voters), the class of strategy-proof voting rules that satisfy non-imposition is exactly the class of voting rules that can be decomposed into multiple strategy-proof local rules, one for each issue and each setting of the issues preceding it. Technically, it is exactly the class of all *conditional rule nets (CR-nets)*, defined later in this chapter but analogous to CP-nets, whose local (issue-wise) entries are strategy-proof voting rules. CR-nets represent how the voting rule's behavior on one issue depends on the decisions made on all issues preceding it. Conceptually, this is similar to how acyclic CP-nets represent how a voter's preferences on one issue depend on the decisions made on all issues preceding it.

12.1 Conditional Rule Nets (CR-Nets)

In this section, we give the motivation and formal definition of CR-nets. In a sequential voting rule, the local voting rule that is used for a given issue is always the same, that is, the local voting *rule* does not depend on the decisions made on earlier issues (though, of course, the voters' *preferences* for this issue do depend on those decisions).

However, in many cases, it makes sense to let the local voting rules depend on the values of preceding issues. For example, let us consider again the setting in Example 8.0.2, where a group of people must make a joint decision on the menu for dinner, and the menu is composed of two issues: the main course (**M**) and the wine (**W**). Let us suppose that the caterer is collecting the votes and making the decision based on some rule. Suppose the order of voting is $\mathbf{M} > \mathbf{W}$. Suppose the main course is determined to be beef. One would expect that, conditioning on beef being selected, most voters prefer red wine (e.g., $r \succ p \succ w$). Still, it can happen that even conditioned on beef being selected, surprisingly, slightly more than half the voters vote for white wine ($w \succ p \succ r$), and slightly less than half vote for red ($r \succ p \succ w$). In this case, the caterer, who knows that in the general population most people prefer red to white given a meal of beef, may "overrule" the preference for white wine among the slight majority of the voters, and select red wine anyway. While this may appear somewhat snobbish on the part of the caterer, in fact she may be acting in the best interest of social welfare if we take the non-voting agents (who are likely to prefer red given beef) into account.

To model voting rules where the local rules depend on the values chosen for earlier issues, we introduce *conditional rule nets (CR-nets)*. A CR-net is defined similarly to a CP-net—the difference is that CPTs are replaced by conditional rule tables (CRTs), which specify a local voting rule over D_i for each issue X_i and each setting of the parents of X_i.[1]

Definition 12.1.1. *An (acyclic)* conditional rule net (CR-net) $\mathcal{M}$ *over* $\mathcal{X}$ *is composed of the following two parts.*

[1] It is not clear how a cyclic CR-net could be useful, so we only define acyclic CR-nets.

1. A directed acyclic graph *G over $\{X_1, \ldots, X_p\}$.*

2. A set of conditional rule tables *(CRTs) in which, for any variable X_i and any setting $\vec{d}$ of $Par_G(X_i)$, there is a* local conditional voting rule $\mathcal{M}|_{X:\vec{d}}$ *over D_i.*

A CR-net encodes a voting rule over all $\mathcal{O}$-legal profiles (we recall that we fix $\mathcal{O} = X_1 > \cdots > X_p$ in this chapter). For any $1 \leqslant i \leqslant p$, in the ith step, the value d_i is determined by applying $\mathcal{M}|_{X_i:d_1\cdots d_{i-1}}$ (the local rule specified by the CR-net for the ith issue given that the earlier issues take the values $d_1 \cdots d_{i-1}$) to $P|_{X_i:d_1\cdots d_{i-1}}$ (the profile of preferences over the ith issue, given that the earlier issues take the values $d_1 \cdots d_{i-1}$). Formally, for any $\mathcal{O}$-legal profile P, $\mathcal{M}(P) = (d_1, \ldots, d_p)$ is defined as follows: $d_1 = \mathcal{M}|_{X_1}(P|_{X_1})$, $d_2 = \mathcal{M}|_{X_2:d_1}(P|_{X_2:d_1})$, etc. Finally, $d_p = \mathcal{M}|_{X_p:d_1\cdots d_{p-1}}(P|_{X_p:d_1\cdots d_{p-1}})$.

A CR-net $\mathcal{M}$ is *separable* if there are no edges in the graph of $\mathcal{M}$. That is, the local voting rule for any issue is independent of the values of all other issues (which corresponds to a sequential voting rule).

12.2 Restricting Voters' Preferences

We now consider restrictions on preferences. A restriction on preferences (for a single voter) rules out some of the possible preferences in $L(\mathcal{X})$. Following the convention of Le Breton and Sen (1999), a *preference domain* is a set of all admissible profiles, which represents the restricted preferences of the voters. Usually a preference domain is the Cartesian product of the sets of restricted preferences for individual voters. A natural way to restrict preferences in a multi-issue domain is to restrict the preferences on individual issues. For example, we may decide that $r \succ w \succ p$ is not a reasonable preference for wine (regardless of the choice of main course), and therefore rule it out (assume it away). More generally, which preferences are considered reasonable for one issue may depend on the decisions for the other issues. Hence, in general, for each i, for each setting $\vec{d_i}$ of the issues before issue X_i, there is a set of "reasonable" (or: possible, admissible) preferences over X_i, which we call $\mathcal{L}|_{X_i:\vec{d_i}}$. Formally, *admissible conditional preference sets*, which encode all possible conditional preferences of voters, are defined as follows.

Definition 12.2.1. *An* admissible conditional preference set $\mathcal{L}$ *over* $\mathcal{X}$ *is composed of multiple* local conditional preference sets*, denoted by* $\mathcal{L}|_{X_i:\vec{d_i}}$*, such that for any* $1 \leqslant i \leqslant p$ *and any* $\vec{d_i} \in D_1 \times \cdots \times D_{i-1}$*,* $\mathcal{L}|_{X_i:\vec{d_i}}$ *is a set of (not necessarily all) linear orders over* D_i*.*

That is, for any $1 \leqslant i \leqslant p$ and any $\vec{d_i} \in D_1 \times \cdots \times D_{i-1}$, $\mathcal{L}|_{X_i:\vec{d_i}}$ encodes the voter's local language over issue i, given the preceding issues taking values $\vec{d_i}$. In other words, if $\mathcal{L}$ is the admissible conditional preference set for a voter, then we require the voter's preferences over X_i given $\vec{d_i}$ to be in $\mathcal{L}|_{X_i:\vec{d_i}}$.

An admissible conditional preference set restricts the possible CP-nets, preferences, and lexicographic preferences. We note that Le Breton and Sen (1999) defined a similar structure, which works specifically for separable votes.

Now we are ready to define the restricted preferences of a voter over $\mathcal{X}$. Let $\mathcal{L}$ be the admissible conditional preference set for the voter. A voter's admissible vote can be generated in the following two steps: first, a CP-net $\mathcal{N}$ is constructed such that for any $1 \leqslant i \leqslant p$ and any $\vec{d_i} \in D_1 \times \cdots \times D_{i-1}$, the restriction of $\mathcal{N}$ on X_i given $\vec{d_i}$ is chosen from $\mathcal{L}|_{X_i:\vec{d_i}}$; second, an extension of $\mathcal{N}$ is chosen as the voter's vote. By restricting the freedom in either of the two steps (or both), we obtain a set of restricted preferences for the voter. Hence, we have the following definitions.

Definition 12.2.2. *Let* $\mathcal{L}$ *be an admissible conditional preference set over* $\mathcal{X}$*.*

- *CPnets*$(\mathcal{L}) = \{\mathcal{N} : \mathcal{N}$ *is a CP-net over*$\mathcal{X}$*, and* $\forall i, \forall \vec{d_i} \in D_1 \times \cdots \times D_{i-1}, \mathcal{N}|_{X_i:\vec{d_i}} \in \mathcal{L}|_{X_i:\vec{d_i}}\}$.
- *Pref*$(\mathcal{L}) = \{V : V \sim \mathcal{N}, \mathcal{N} \in$ *CPnets*$(\mathcal{L})\}$.
- $LD(\mathcal{L}) = \{Lex(\mathcal{N}) : \mathcal{N} \in$ *CPnets*$(\mathcal{L})\}$.

That is, CPnets($\mathcal{L}$) is the set of all CP-nets over $\mathcal{X}$ corresponding to preferences that are consistent with the admissible conditional preference set $\mathcal{L}$. Pref($\mathcal{L}$) is the set of all linear orders that are consistent with the admissible conditional preference set $\mathcal{L}$. $LD(\mathcal{L})$, which we call the *lexicographic preference domain*, is the subset of linear orders in Pref($\mathcal{L}$) that are lexicographic. For any $L \subseteq \text{Pref}(\mathcal{L})$, we say that L *extends* $\mathcal{L}$ if for any CP-net in CPnets($\mathcal{L}$), there exists at least one linear order in L consistent with that CP-net. It follows that $LD(\mathcal{L})$ extends $\mathcal{L}$; in this case, for any CP-net $\mathcal{N}$ in CPnets($\mathcal{L}$), there exists exactly one linear order in $LD(\mathcal{L})$ that extends $\mathcal{N}$. Lexicographic preference domains are natural extensions of admissible conditional preference sets, but they are also quite restrictive, since any CP-net only has one lexicographic extension.

We now define a notion of richness for admissible conditional preference sets. This notion says that for any issue, given any setting of the earlier issues, each value of the current issue can be the most-preferred one.[2]

Definition 12.2.3. *An admissible conditional preference set $\mathcal{L}$ is* rich *if for each $1 \leqslant i \leqslant p$, each valuation $\vec{d_i}$ of the preceding issues, and each $a_i \in D_i$, there exists $V^i \in \mathcal{L}|_{X_i:\vec{d_i}}$ such that a_i is ranked in the top position of V^i.*

We remark that richness is a natural requirement, and it is also a very weak restriction in the following sense. It only requires that when a voter is asked about her (local) preferences over X_i given $\vec{d_i}$, she should have the freedom to at least specify her most preferred local alternative in D_i at will. We note that $|\mathcal{L}|_{X_i:\vec{d_i}}|$ can be as small as $|D_i|$ (by letting each alternative in D_i be ranked in the top position exactly once), which is in sharp contrast to $|L(D_i)| = |D_i|!$ (when all local orders are allowed).

A CR-net $\mathcal{M}$ is *locally strategy-proof* if all its local conditional rules are strategy-

[2] This is *not* the same richness notion as the one proposed by Le Breton and Sen, which applies to preferences over all alternatives rather than to admissible conditional preference sets.

proof over their respective local domains (we recall that the voters' local preferences must be in the corresponding local conditional preference set). That is, for any $1 \leqslant i \leqslant p, \vec{d_i} \in D_1 \times \cdots \times D_{i-1}$, $\mathcal{M}|_{X_i:\vec{d_i}}$ is strategy-proof over $\prod_{j=1}^{n} \mathcal{L}_j|_{X_i:\vec{d_i}}$.

12.3 Strategy-Proof Voting Rules in Lexicographic Preference Domains

In this section, we present our main theorem, which characterizes strategy-proof voting rules that satisfy non-imposition, when the voters' preferences are restricted to lexicographic preference domains. Our main theorem states the following: if each voter's preferences are restricted to the lexicographic preference domain for a rich admissible conditional preference set, then a voting rule that satisfies non-imposition is strategy-proof if and only if it is a locally strategy-proof CR-net. We recall that there are at least two issues with at least two possible values each, and the lexicographic preference domain for a rich admissible conditional preference set $\mathcal{L}$ is composed of all lexicographic extensions of the CP-nets that are constructed from $\mathcal{L}$.

Theorem 12.3.1. *For any $1 \leqslant j \leqslant n$, suppose $\mathcal{L}_j$ is a rich admissible conditional preference set, and voter j's preferences are restricted to the lexicographic preference domain of $\mathcal{L}_j$. Then, a voting rule r that satisfies non-imposition is strategy-proof if and only if r is a locally strategy-proof CR-net.*

Proof of Theorem 12.3.1: In this proof, for any $1 \leqslant i \leqslant p$, we let X_{-i} denote $\mathcal{I} \backslash \{X_i\}$. For any $1 \leqslant j \leqslant n$, any profile P of n votes, we let P_{-j} denote the profile that consists of all votes in P except the vote by voter j.

Before formally proving the theorem, let us first sketch the idea behind the proof. The "if" part is easy. The "only if" part is proved by induction on p (the number of issues). More precisely, suppose the theorem holds for p issues. For $p+1$ issues, let r be a strategy-proof voting rule that satisfies non-imposition. We first prove that r

can be decomposed in the following way: there exists a local rule r_1 over D_1 and a voting rule $r_{X_{-1}:a_1}$ over $D_2 \times \cdots \times D_{p+1}$ for each $a_1 \in D_1$ that satisfy the following two conditions.

1. For any profile P, the first component of $r(P)$ is determined by applying r_1 to the projection of P on X_1, and

2. the remaining components are determined by applying $r_{X_{-1}:a_1}$ to the restriction of P on the remaining issues given $X_1 = a_1$, where a_1 is the first component of $r(P)$, which is just determined by r_1.

Moreover, we prove that r_1 and $r_{X_{-1}:a_1}$ (for all $a_1 \in D_1$) satisfy non-imposition and strategy-proofness. Therefore, by the induction hypothesis, for each $a_1 \in D_1$, $r_{X_{-1}:a_1}$ is a locally strategy-proof CR-net over $D_2 \times \cdots \times D_{p+1}$. It follows that r is a locally strategy-proof CR-net over $D_1 \times \cdots \times D_{p+1}$, in which the (unconditional) rule for X_1 is r_1, and given any $a_1 \in D_1$, the sub-CR-net conditioned on $X_1 = a_1$ is $r_{X_{-1}:a_1}$.

We now formally prove the theorem. We will use Lemma 12.3.2, which states that any strategy-proof rule r satisfies monotonicity, that is, for any profile P, if each voter changes her vote by ranking $r(P)$ higher, then the winner is still $r(P)$.

Lemma 12.3.2 (Known)**.** *Any strategy-proof voting rule satisfies monotonicity.*

Proof of Lemma 12.3.2: Suppose for the sake of contradiction r is strategy-proof but does not satisfy monotonicity. It follows that there exists a profile P, i, and V_i' such that V_i' is obtained from V_i by raising $r(P)$, and $r(P_{-i}, V_i') \neq r(P)$. If $r(P_{-i}, V_i') \succ_{V_i'} r(P)$, then we must have that $r(P_{-i}, V_i') \succ_{V_i} r(P)$, which means that voter i has incentive to falsely report that her true preferences are V_i'; if $r(P) \succ_{V_i'} r(P_{-i}, V_i')$, then when voter i's true preferences are V_i' and the other voters' profile is P_{-i}, she has incentive to falsely report that her preferences are V_i. In either case there is a manipulation, which contradicts the assumption that r is strategy-proof. □

First, we prove the "only if " part by induction on p. When $p = 1$, the theorem is immediate. Now, suppose that the theorem holds when $p = k$. When $p = k + 1$, for any strategy-proof rule r that satisfies non-imposition, over $\mathcal{X}_{k+1} = D_1 \times \cdots \times D_{k+1}$, we prove that this rule can be decomposed into two parts: first, it applies a local voting rule r_1 for X_1, and subsequently, it applies a rule $r|_{X_{-1}:a_1}$ for X_{-1}, which depends on the outcome of r_1. Thus, we have the property that for any $P \in \prod_{j=1}^{n} LD(\mathcal{L}_j)$, we have $r(P) = (r_1(P|_{X_1}), r|_{X_{-1}:r_1(P|_{X_1})}(P|_{X_{-1}:r_1(P|_{X_1})}))$. Then, we will show that the induction assumption can be applied to the second part.

To prove these, we claim that for any strategy-proof voting rule r satisfying non-imposition, and any $P \in \prod_{j=1}^{n} LD(\mathcal{L}_j)$, the value of issue X_1 for the winning alternative only depends on the restriction of the profile to X_1. That is, we show that for any pair of profiles $P, Q \in \prod_{j=1}^{n} LD(\mathcal{L}_j)$, where $P = (V_1, \ldots, V_n)$, $Q = (W_1, \ldots, W_n)$ and $P|_{X_1} = Q|_{X_1}$, we must have $r(P)|_{X_1} = r(Q)|_{X_1}$. Suppose on the contrary that $r(P)|_{X_1} \neq r(Q)|_{X_1}$. For any $0 \leqslant j \leqslant n$, we define $P_j = (W_1, \ldots, W_j, V_{j+1}, \ldots, V_n)$. It follows that $P_0 = P$ and $P_n = Q$. We claim that for any $0 \leqslant j \leqslant n - 1$, $r(P_j)|_{X_1} = r(P_{j+1})|_{X_1}$. For the sake of contradiction, suppose $r(P_j)|_{X_1} \neq r(P_{j+1})|_{X_1}$ for some $j \leqslant n - 1$. Let $a_1 = r(P_j)|_{X_1}$ and $b_1 = r(P_{j+1})|_{X_1}$. If $a_1 \succ_{V_{j+1}|_{X_1}} b_1$, then, because $V_{j+1}|_{X_1} = W_{j+1}|_{X_1}$, (P_{j+1}, V_{j+1}) is a successful manipulation; on the other hand, if $b_1 \succ_{V_{j+1}|_{X_1}} a_1$, then, (P_j, W_{j+1}) is a successful manipulation. This contradicts the strategy-proofness of r. Thus, we have shown that the value of issue X_1 for the winning alternative only depends on the restriction of the profile to X_1.

Therefore, we can define a voting rule r_1 over D_1 as follows. For any $P^1 \in \prod_{j=1}^{n} \mathcal{L}_j|_{X_1}$, $r_1(P^1) = r(P)|_{X_1}$, where $P \in \prod_{j=1}^{n} LD(\mathcal{L}_j)$ and $P|_{X_1} = P^1$. Such a P exists because $LD(\mathcal{L}_j)$ extends $\mathcal{L}_j$ for all j, and this is well-defined by the observation from the previous paragraph. r_1 satisfies non-imposition because r satisfies non-imposition.

Next, we prove that r_1 is strategy-proof. If we assume for the sake of contradiction that r_1 is not strategy-proof, then there exists a successful manipulation $(P^1, \hat{V}_l^1)$ over D_1, where voter l is the manipulator, and $P^1 = (V_1^1, \ldots, V_n^1)$. Let $\mathcal{N}_1, \ldots, \mathcal{N}_n, \hat{\mathcal{N}}_l$ be $n+1$ CP-nets satisfying the following conditions.

- For any $1 \leqslant j \leqslant n$, $\mathcal{N}_j|_{X_1} = V_j^1$; $\hat{\mathcal{N}}_l|_{X_1} = \hat{V}_l^1$.
- For any $1 \leqslant j \leqslant n$, $\mathcal{N}_j \in \text{CPnets}(\mathcal{L}_j)$, $\hat{\mathcal{N}}_l \in \text{CPnets}(\mathcal{L}_l)$.

For $1 \leqslant j \leqslant n$, let V_j be the lexicographic extension of $\mathcal{N}_j$. Let $\hat{V}_l$ be the lexicographic extension of $\hat{\mathcal{N}}_l$. Let $P = (V_1, \ldots, V_n)$. We note that the X_1 component of $r(P_{-l}, \hat{V}_l)$ is $r_1(P_{-l}^1, \hat{V}_l^1) \succ_{V_l^1} r_1(P^1)$, which is the X_1 component of $r(P)$. Because V_l is the lexicographic extension of $\mathcal{N}_l$, and $\mathcal{N}_l|_{X_1} = V_l^1$, we have that $r(P_{-l}, \hat{V}_l) \succ_{V_l} r(P)$, which means that $(P, \hat{V}_l)$ is a successful manipulation. This contradicts the strategy-proofness of r. So, we have shown that r_1 is strategy-proof.

We next show that the second part of r can be written as $r|_{X_{-1}:r_1(P|_{X_1})}(P|_{X_{-1}:r_1(P|_{X_1})})$. That is, the rule for the remaining issues X_{-1} only depends on the outcome for X_1. For any $V \in \mathit{Legal}(\mathcal{O})$ and any $a_1 \in D_1$, we let $V|_{X_{-1}:a_1}$ denote the linear preference over D_{-1} that is compatible with the restriction of V to the set of alternatives whose X_1 component is a_1, that is, for any $\vec{a}_{-1}, \vec{b}_{-1} \in D_{-1}$, $\vec{a}_{-1} \succeq_{V|_{X_{-1}:a_1}} \vec{b}_{-1}$ if and only if $(a_1, \vec{a}_{-1}) \succeq_V (a_1, \vec{b}_{-1})$. For any $\mathcal{O}$-legal profile P, $P|_{X_{-1}:a_1}$ is composed of $V|_{X_{-1}:a_1}$ for all $V \in P$. For any CP-net $\mathcal{N}$, we let $\mathcal{N}|_{X_{-1}:a_1}$ denote the sub-CP-net of $\mathcal{N}$ conditioned on $X_1 = a_1$. It follows that if $V \sim \mathcal{N}$, then, $V|_{X_{-1}:a_1} \sim \mathcal{N}|_{X_{-1}:a_1}$.

Now, we claim that for any pair of profiles $P_1, P_2 \in \prod_{j=1}^n LD(\mathcal{L}_j)$, $P_1 = (V_1, \ldots, V_n)$ and $P_2 = (W_1, \ldots, W_n)$, such that $a_1 = r_1(P_1) = r_1(P_2)$ and $P_1|_{X_{-1}:a_1} = P_2|_{X_{-1}:a_1}$, we must have $r(P_1) = r(P_2)$. To prove this, we construct a profile P such that $r(P_1) = r(P) = r(P_2)$. For any $1 \leqslant j \leqslant n$, we let $V_j^{a_1} \in \mathcal{L}_j|_{X_1}$ be an arbitrary linear order over D_1 in which a_1 is in the top position. Let $P = (Q_1, \ldots, Q_n) \in \prod_{j=1}^n LD(\mathcal{L}_j)$

be the profile in which for any $1 \leqslant j \leqslant n$, Q_j is the lexicographic extension of the CP-net $\mathcal{N}_j$ that satisfies the following conditions.

- $\mathcal{N}_j|_{X_1} = V_j^{a_1}$.
- $\mathcal{N}_j|_{X_{-1}:a_1} = \hat{\mathcal{N}}_j|_{X_{-1}:a_1}$, where $\hat{\mathcal{N}}_j$ is the CP-net that V_j extends.

Let $\vec{a} = (a_1, \vec{a}_{-1}) = r(P_1)$. For any $1 \leqslant j \leqslant n$ and any $\vec{b} \in \mathcal{X}$ with $\vec{b} \succ_{Q_j} \vec{a}$, we have that the X_1 component of $\vec{b}$ must be a_1, because Q_j is lexicographic, and a_1 is in the top position of $Q_j|_{X_1}$. We let $\vec{b} = (a_1, \vec{b}_{-1})$. It follows that $\vec{b}_{-1} \succ_{Q_j|_{X_{-1}:a_1}} \vec{a}_{-1}$. We note that $Q_j|_{X_{-1}:a_1}$ is the lexicographic extension of $\mathcal{N}_j|_{X_{-1}:a_1}$, $V_j|_{X_{-1}:a_1}$ is the lexicographic extension of $\hat{\mathcal{N}}_j|_{X_{-1}:a_1}$, and $\mathcal{N}_j|_{X_{-1}:a_1} = \hat{\mathcal{N}}_j|_{X_{-1}:a_1}$. Therefore, $Q_j|_{X_{-1}:a_1} = V_j|_{X_{-1}:a_1}$, which means that $\vec{b}_{-1} \succ_{V_j|_{X_{-1}:a_1}} \vec{a}_{-1}$. Hence, we have $\vec{b} \succ_{V_j} \vec{a}$. By Lemma 12.3.2, we have $r(P) = r(P_1)$. By similar reasoning, $r(P) = r(P_2)$, which means that $r(P_1) = r(P) = r(P_2)$. It follows that for any $a_1 \in D_1$, there exists a voting rule $r|_{X_{-1}:a_1}$ over $D_2 \times \cdots \times D_p$ such that for any $P \in \prod_{j=1}^{n} LD(\mathcal{L}_j)$,

$$r(P) = (r_1(P|_{X_1}), r|_{X_{-1}:r_1(P|_{X_1})}(P|_{X_{-1}:r_1(P|_{X_1})}))$$

At this point, we have shown that r can be decomposed as desired. We next show that for any $a_1 \in D_1$, $r|_{X_{-1}:a_1}$ is strategy-proof over $\prod_{j=1}^{n} LD(\mathcal{L}_j|_{X_{-1}:a_1})$. Suppose for the sake of contradiction that there exists a successful manipulation $(P^{-1}, \hat{V}_l^{-1})$, where voter l is the manipulator, and $P^{-1} = (V_1^{-1}, \ldots, V_n^{-1})$. Let $\mathcal{N}_1, \ldots, \mathcal{N}_n, \hat{\mathcal{N}}_l$ be $n+1$ CP-nets satisfying the following conditions.

- For any $1 \leqslant j \leqslant n$, $top(\mathcal{N}_j|_{X_1}) = a_1$. That is, a_1 is ranked in the top position in the restriction of $\mathcal{N}_j$ to X_1. Also, $top(\hat{\mathcal{N}}_l|_{X_1}) = a_1$.
- For any $1 \leqslant j \leqslant n$, $\mathcal{N}_j|_{X_{-1}:a_1}$ is the CP-net over D_{-1} that V_j^{-1} extends; $\hat{\mathcal{N}}_l|_{X_{-1}:a_1}$ is the CP-net over D_{-1} that $\hat{V}_l^{-1}$ extends.
- For any $1 \leqslant j \leqslant n$, $\mathcal{N}_j \in \text{CPnets}(\mathcal{L}_j)$; $\hat{\mathcal{N}}_l \in \text{CPnets}(\mathcal{L}_l)$.

The existence of these CP-nets is guaranteed by the richness of $\mathcal{L}_j$ for any $1 \leqslant$

$j \leqslant n$. For any $1 \leqslant j \leqslant n$, let V_j be the lexicographic extension of $\mathcal{N}_j$. Let $\hat{V}_l$ be the lexicographic extension of $\hat{\mathcal{N}_l}$. Let $P = (V_1, \ldots, V_n)$. We note that

$$\begin{aligned} r(P) &= (r_1(P|_{X_1}), r|_{X_{-1}:r_1(P|_{X_1})}(P|_{X_{-1}:r_1(P|_{X_1})})) \\ &= (a_1, r|_{X_{-1}:a_1}(P|_{X_{-1}:a_1})) = (a_1, r|_{X_{-1}:a_1}(P^{-1})) \\ &\prec_{V_l} (a_1, r|_{X_{-1}:a_1}(P^{-1}_{-l}, \hat{V}_l)) = r(P_{-l}, \hat{V}_l) \end{aligned}$$

This contradicts the strategy-proofness of r. Hence, we have shown that for any $a_1 \in D_1$, $r|_{X_{-1}:a_1}$ is strategy-proof over $\prod_{j=1}^{n} LD(\mathcal{L}_j|_{X_{-1}:a_1})$.

Moreover, because r satisfies non-imposition, for any $a_1 \in D_1$, $r|_{X_{-1}:a_1}$ satisfies non-imposition. Hence, for any $a_1 \in D_1$, we can apply the induction assumption to $r|_{X_{-1}:a_1}$ and conclude that it is a locally strategy-proof CR-net over D_{-1}. It follows that r is a locally strategy-proof CR-net over $\mathcal{X}$, completing the first part of the proof.

We next prove the "if" part. If the proposition does not hold, then there exists a locally strategy-proof CR-net $\mathcal{M}$ for which there is a successful manipulation $(P, \hat{V}_l)$. Let $i \leqslant p$ be the smallest natural number such that $\mathcal{M}(P)|_{X_i} \neq \mathcal{M}(P_{-l}, \hat{V}_l)|_{X_i}$. Let $\vec{d}_i$ be the first $i-1$ components of $\mathcal{M}(P)$ and $\mathcal{M}(P_{-l}, \hat{V}_l)$. Because $\mathcal{M}|_{X_i:\vec{d}_i}$ is strategy-proof, we have the following calculation.

$$\begin{aligned} \mathcal{M}(P)|_{X_i} &= \mathcal{M}|_{X_i:\vec{d}_i}(P|_{X_i:\vec{d}_i}) \\ &\succ_{V_l|_{X_i:\vec{d}_i}} \mathcal{M}|_{X_i:\vec{d}_i}(P_{-1}, \hat{V}_l|_{X_i:\vec{d}_i}) \\ &= \mathcal{M}(P_{-l}, \hat{V}_l)|_{X_i} \end{aligned}$$

Because V_l is lexicographic, for any $\vec{y}, \vec{z} \in D_{i+1} \times \cdots \times D_p$, we have

$$(\vec{d}_i, \mathcal{M}|_{X_i:\vec{d}_i}(P), \vec{y}) \succ_{V_l} (\vec{d}_i, \mathcal{M}|_{X_i:\vec{d}_i}(P_{-1}, \hat{V}_l), \vec{z})$$

Therefore, $\mathcal{M}(P) \succ_{V_l} \mathcal{M}(P_{-1}, \hat{V}_l)$, which contradicts the assumption that $(P, \hat{V}_l)$ is a successful manipulation. Hence, locally strategy-proof CR-nets are strategy-proof for lexicographic preferences. □

It follows from Theorem 12.3.1 that any sequential voting rule that is composed of locally strategy-proof voting rules is strategy-proof over lexicographic preference domains, because a sequential voting rule is a separable CR-net. Specifically, when the multi-issue domain is binary (that is, for any $1 \leqslant i \leqslant p$, $|D_i| = 2$), the sequential composition of majority rules is strategy-proof when the profiles are lexicographic. It is interesting to view this in the context of previous works on the strategy-proofness of sequential composition of majority rules: Lacy and Niou (2000) and Le Breton and Sen (1999) showed that the sequential composition of majority rules is strategy-proof when the profile is restricted to the set of all separable profiles; on the other hand, Lang and Xia (2009) showed that this rule is not strategy-proof when the profile is restricted to the set of all $\mathcal{O}$-legal profiles.

The restriction to lexicographic preferences is still limiting. Next, we investigate whether there is any other preference domain for the voters on which the set of strategy-proof voting rules that satisfy non-imposition is equivalent to the set of all locally strategy-proof CR-nets. The answer to this question is "No," as shown in the next result. More precisely, over any preference domain that extends an admissible conditional preference set, the set of strategy-proof voting rules satisfying non-imposition and the set of locally strategy-proof CR-nets satisfying non-imposition are identical *if and only if* the preference domain is lexicographic.

Theorem 12.3.3. *For any $1 \leqslant j \leqslant n$, suppose $\mathcal{L}_j$ is a rich admissible conditional preference set, $L_j \subseteq$ Pref$(\mathcal{L}_j)$, and L_j extends $\mathcal{L}_j$. If there exists $1 \leqslant j \leqslant n$ such that L_j is not the lexicographic preference domain of $\mathcal{L}_j$, then there exists a locally strategy-proof CR-net $\mathcal{M}$ that satisfies non-imposition and is not strategy-proof over $\prod_{j=1}^{n} L_j$.*

Proof of Theorem 12.3.3: If, for some $j \leqslant n$, there is a $V'_j \in LD(\mathcal{L}_j)$ that is not in L_j, then there must also be a $V_j \in L_j$ that is not in $LD(\mathcal{L}_j)$, because some vote in L_j

must extend the CP-net that V'_j extends. Hence, if $\prod_{j=1}^n LD(\mathcal{L}_j) \neq \prod_{j=1}^n LD(\mathcal{L}_j)$, there must exist some $j \leqslant n$, $V_j \in L_j$ such that V_j is not in $LD(\mathcal{L}_j)$. For this V_j, there must exist $i \leqslant p$, $\vec{a}_{i-1} \in D_1 \times \cdots \times D_{i-1}$, $a_i, b_i \in D_i$, $\vec{a}_{i+1}, \vec{b}_{i+1} \in D_{i+1} \times \cdots \times D_p$ such that $a_i \succ_{V_j|_{X_i:\vec{a}_{i-1}}} b_i$, and $(\vec{a}_{i-1}, b_i, \vec{b}_{i+1}) \succ_{V_j} (\vec{a}_{i-1}, a_i, \vec{a}_{i+1})$. Now, let us define a CR-net $\mathcal{M}$ as follows.

- $\mathcal{M}|_{X_i:\vec{a}_{i-1}}$ is the plurality rule that only counts voter 1 and voter j's votes; ties are broken in the order $b_i \succ a_i \succ D_i - \{a_i, b_i\}$.

- Any other local conditional voting rule is a dictatorship by voter 1.

Now, let $\mathcal{N}_1 \in \text{CPnets}(\mathcal{L}_1)$ be a CP-net such that $top(\mathcal{N}_1) = (\vec{a}_{i-1}, a_i, \vec{a}_{i+1})$, and for any $k \geqslant i+1$, $top(\mathcal{N}_1|_{X_k:\vec{a}_{i-1}b_ib_{i+1}\cdots b_{k-1}}) = b_k$. Here $top(\mathcal{N}_1)$ is the top-ranked alternative in $\mathcal{N}_1$. Let $\mathcal{N}'_j \in \text{CPnets}(\mathcal{L}_j)$ be a CP-net such that $top(\mathcal{N}'_j) = (\vec{a}_{i-1}, b_i, \vec{b}_{i+1})$. Let $V_1 \in L_1$ be such that $V_1 \sim \mathcal{N}_1$, and let $V'_j \in L_j$ be such that $V'_j \sim \mathcal{N}'_j$. Such V_1 and V'_j must exist, because L_1 extends $\mathcal{L}_1$, and L_j extends $\mathcal{L}_j$. For any profile $P = (V_1, \ldots, V_j, \ldots, V_n) \in \prod_{j=1}^n LD(\mathcal{L}_j)$ (that is, for any $l \neq 1, j$, V_l is chosen arbitrarily, because $\mathcal{M}(P)$ does not depend on them), it follows that $\mathcal{M}(P) = (\vec{a}_{i-1}, a_i, \vec{a}_{i+1})$, and $\mathcal{M}(P_{-j}, V'_j) = (\vec{a}_{i-1}, b_i, \vec{b}_{i+1})$, which means that (P, V'_j) is a successful manipulation for voter j. So, $\mathcal{M}$ is not strategy-proof (and it satisfies non-imposition). □

12.4 Summary

In this chapter, we studied strategy-proof voting rules when the voters' preferences are represented by acyclic CP-nets that follow a common order over issues. We showed that if each voter's preferences are restricted to a lexicographic preference domain, then a voting rule satisfying non-imposition is strategy-proof if and only if it is a locally strategy-proof CR-net. We then proved that this characterization only works for lexicographic domains.

Our characterization is quite positive; however, beyond that, it is still not clear how much we can hope for desirable strategy-proof voting rules in multi-issue domains.[3] Of course, it is well known that it is difficult to obtain strategy-proofness in voting settings in general, and this does not mean that we should abandon voting as a general method. Similarly, difficulties in obtaining desirable strategy-proof voting rules in multi-issue domains should not prevent us from studying voting rules for multi-issue domains altogether. From a mechanism design perspective, strategy-proofness is a very strong criterion, which corresponds to implementation in dominant strategies. It may well be the case that rules that are not strategy-proof still result in good outcomes in practice—or, more formally, in (say) Bayes-Nash equilibrium.

[3] In fact, we also proved two impossibility theorems, which (informally) both state that as soon as we go beyond lexicographic domains, there are no strategy-proof voting rules, except CR-nets where local rules are dictatorships. These results are omitted due to their heavy technicality and notation. They can be found on my homepage.

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-

ally tractable mechanisms to select the outcome thereafter. For example, the popular XOR-language used in combinatorial auctions has a close relationship with a language that has been investigated in combinatorial voting called *GAI-networks* (Gonzales et al., 2008). See also Conitzer (2010). Therefore, exploring richer connections between combinatorial voting and combinatorial auctions can help in designing good voting/auction rules for both of them.

2. **Designing voting rules based on "local" voting rules.** Our H-composition framework leaves several computational challenges. Some of them have been resolved in Conitzer et al. (2011b), where we proved that for several choice set functions, the winner under H-composition is NP-hard to compute. Future work includes designing heuristic, approximation, or fixed-parameter tractable algorithms that would work well under certain natural assumptions, for example, when the voters' preferences share some common structure.

3. **Other principled approaches.** We have shown that the MLE approach taken in Chapter 10 allowed us to define genuinely new families of voting correspondences for multi-issue domains. However, the computational aspects of determining the winners under MLE correspondences are still not completely clear. For example, we only characterized the complexity of computing winners under MLEs of distance-based threshold models with thresholds 1 and p (the number of issues). It would be interesting to identify the complexity for other thresholds (however, we conjecture that it is at least NP-hard). Another promising principled approach that has not yet been applied to combinatorial voting is *distance-rationalizable voting rules* (Meskanen and Hannu, 2008; Elkind et al., 2009a, 2010b,c, 2011). To define a voting rule in the distance-rationalizability framework, a distance (metric) is defined for any pair of profiles, and a winner is associated to some profiles where the voters reach a

consensus (for some notion of consensus, for example, profiles with a Condorcet winner). Then, a voting rule can be viewed as the function that selects the winner in the closest consensus profile. The distance-based rationalizable framework can be easily adopted to design new rules for combinatorial voting: we only need to define a natural distance between profiles represented by some compact language (for example, some distance that is based on the Hamming distance between CP-nets over multi-binary-issue domains), and a set of profiles where voters reach a consensus. The main question here is the quality of the voting rule that is distance-rationalized in this way, especially the computational complexity.

There are also many interesting topics for future research about the game-theoretic aspects of combinatorial voting. For example, is there any criterion for the selection of the order over the issues in sequential voting games? Perhaps more importantly, how can we get around the multiple-election paradoxes in sequential voting games? For example, Theorem 11.4.3 shows that if the voters' preferences are lexicographic, then we can avoid the paradoxes. It is not clear if there are other ways to avoid the paradoxes (paradoxes occur even if we restrict voters' preferences to be separable or $\mathcal{O}$-legal, as shown in Theorem 11.4.1 and Theorem 11.4.6). Another approach is to consider other, non-sequential voting procedures for multi-issue domains. What are good examples of such procedures? Will these avoid paradoxes? What is the effect of strategic behavior for such procedures? How should we even define "strategic behavior" for such procedures, or for sequential voting with non-binary issues, or for voting rules in general? How can we extend these results to incomplete-information settings? Also, beyond proving paradoxes for individual rules, is it possible to show a general impossibility result that shows that under certain minimal conditions, paradoxes cannot be avoided? Can we find other domain restrictions to obtain strategy-proof voting rules in multi-issue domains?

Bibliography

Ahn, D. S. and Oliveros, S. (2011), "Combinatorial Voting," *Econometrica*, Forthcoming.

Ailon, N., Charikar, M., and Newman, A. (2005), "Aggregating Inconsistent Information: Ranking and Clustering," in *Proceedings of the Annual Symposium on Theory of Computing (STOC)*, pp. 684–693.

Alon, N. (2006), "Ranking tournaments," *SIAM Journal of Discrete Mathematics*, 20, 137–142.

Altman, A. (2007), "The Axiomatic Approach to Ranking Systems," Ph.D. thesis, Technion.

Arrow, K. J. (1950), "A Difficulty in the Concept of Social Welfare," *Journal of Political Economy*, 58, 328–346.

Aswal, N., Chatterji, S., and Sen, A. (2003), "Dictatorial domains," *Economic Theory*, 22, 45–62.

Aziz, H. (2009), "Algorithmic and Complexity Aspects of Simple Coalitional Games," Ph.D. thesis, University of Warwick.

Bachrach, Y., Betzler, N., and Faliszewski, P. (2010), "Probabilistic Possible Winner Determination," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 697–702, Atlanta, GA, USA.

Bachrach, Y., Elkind, E., and Falisewski, P. (2011), "Coalitional Voting Manipulation: A Game-Theoretic Perspective," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Baharad, E. and Neeman, Z. (2002), "The asymptotic strategyproofness of scoring and Condorcet consistent rules," *Review of Economic Design*, 4, 331–340.

Barbera, S., Sonnenschein, H., and Zhou, L. (1991), "Voting by Committees," *Econometrica*, 59, 595–609.

Barbera, S., Gul, F., and Stacchetti, E. (1993), "Generalized Median Voter Schemes and Committees," *Journal of Economic Theory*, 61, 262–289.

Barbera, S., Masso, J., and Neme, A. (1997), "Voting under Constraints," *Journal of Economic Theory*, 76, 298–321.

Bartholdi, III, J. and Orlin, J. (1991), "Single transferable vote resists strategic voting," *Social Choice and Welfare*, 8, 341–354.

Bartholdi, III, J., Tovey, C., and Trick, M. (1989a), "The computational difficulty of manipulating an election," *Social Choice and Welfare*, 6, 227–241.

Bartholdi, III, J., Tovey, C., and Trick, M. (1989b), "Voting schemes for which it can be difficult to tell who won the election," *Social Choice and Welfare*, 6, 157–165.

Bartholdi, III, J., Tovey, C., and Trick, M. (1992), "How hard is it to control an election?" *Math. Comput. Modelling*, 16, 27–40, Formal theories of politics, II.

Battaglini, M. (2005), "Sequential voting with abstention," *Games and Economic Behavior*, 51, 445–463.

Baumeister, D., Roos, M., and Rothe, J. (2011), "Computational Complexity of Two Variants of the Possible Winner Problem," in *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Taipei, Taiwan.

Baumeister, D. B. and Rothe, J. (2010), "Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules," in *Proceedings of The 19th European Conference on Artificial Intelligence (ECAI)*, Lisbon, Portugal.

Bender, E. A. and Williamson, S. G. (2006), *The Foundations of Combinatorics with Applications*, Dover.

Betzler, N. (2010), "A Multivariate Complexity Analysis of Voting Problems," Ph.D. thesis, University of Jena.

Betzler, N. and Dorn, B. (2010), "Towards a dichotomy for the Possible Winner problem in elections based on scoring rules," *Journal of Computer and System Sciences*, 76, 812–836.

Betzler, N., Fellows, M. R., Guo, J., Niedermeier, R., and Rosamond, F. A. (2009a), "Fixed-parameter algorithms for Kemeny rankings," *heoretical Computer Science*, 410, 4554–4570.

Betzler, N., Hemmann, S., and Niedermeier, R. (2009b), "A multivariate complexity analysis of determining possible winners given incomplete votes," in *Proceedings of the 21st International Joint Conference on Artifical intelligence*, pp. 53–58.

Betzler, N., Niedermeier, R., and Woeginger, G. (2011), “Unweighted coalitional manipulation under the Borda rule is NP-hard,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Black, D. (1948), “On the rationale of group decision-making,” *Journal of Political Economy*, 56, 23–34.

Border, K. C. and Jordan, J. S. (1983), “Straightforward Elections, Unanimity and Phantom Voters,” *The Review of Economic Studies*, 50, 153–170.

Boutilier, C., Brafman, R., Domshlak, C., Hoos, H., and Poole, D. (2004), “CP-nets: A tool for representing and reasoning with conditional ceteris paribus statements,” *Journal of Artificial Intelligence Research*, 21, 135–191.

Bouveret, S. (2007), “Allocation et Partage Equitables de Ressources Indivisibles: Modélisation, Complexité et Algorithmique,” Ph.D. thesis, ISAE Toulouse and University of Toulouse.

Brams, S. J., Kilgour, D. M., and Zwicker, W. S. (1998), “The paradox of multiple elections,” *Social Choice and Welfare*, 15, 211–236.

Brandt, F. (2009), “Some remarks on Dodgson’s voting rule,” *Mathematical Logic Quarterly*, 55, 460–463.

Brandt, F., Fischer, F., and Harrenstein, P. (2007), “The computational complexity of choice sets,” in *Theoretical Aspects of Rationality and Knowledge (TARK)*, Brussels, Belgium.

Brandt, F., Fischer, F., and Harrenstein, P. (2009), “The Computational Complexity of Choice Sets,” *Mathematical Logic Quarterly*, 55, 444–459.

Brandt, F., Bril, M., Hemaspaandra, E., and Hemaspaandra, L. A. (2010a), “Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates,” in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 715–722.

Brandt, F., Fischer, F., Harrenstein, P., and Mair, M. (2010b), “A computational analysis of the tournament equilibrium set,” *Social Choice and Welfare*, 34, 597–609.

Brandt, F., Brill, M., and Seedig, H. G. (2011), “On the Fixed-Parameter Tractability of Composition-Consistent Tournament Solutions,” in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Brucker, P. (2007), *Scheduling Algorithms*, Springer Publishing Company, Incorporated.

Caragiannis, I., Covey, J. A., Feldman, M., Homan, C. M., Kaklamanis, C., Karanikolas, N., Procaccia, A. D., and Rosenschein, J. S. (2009), "On the approximability of Dodgson and Young elections," in *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1058–1067.

Caragiannis, I., Kaklamanis, C., Karanikolas, N., and Procaccia, A. D. (2010), "Socially Desirable Approximations for Dodgson's Voting Rule," in *Proceedings of the 11th ACM Conference on Electronic Commerce*.

Caragiannis, I., Lai, J. K., and Procaccia, A. D. (2011), "Towards More Expressive Cake Cutting," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Chapman, B. (1998), "More Easily Done than Said: Rules, Reasons, and Rational Social Choice," *Oxford Journal of Legal Studies*, 18, 293–329.

Charon, I. and Hudry, O. (2000), "Slater orders and Hamiltonian paths of tournaments," *Electronic Notes in Discrete Mathematics*, 5, 60–63, 6th International Conference on Graph Theory.

Chen, Y. and Pennock, D. M. (2010), "Designing Markets for Prediction," *AI Magazine*, 31, 42–52.

Chen, Y., Fortnow, L., Lambert, N., Pennock, D. M., and Wortman, J. (2008a), "Complexity of Combinatorial Market Makers," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 190–199, Chicago, IL, USA.

Chen, Y., Goel, S., and Pennock, D. M. (2008b), "Pricing combinatorial markets for tournaments," in *STOC '08: Proceedings of the 40th annual ACM Symposium on Theory of Computing*, pp. 305–314.

Chen, Y., Lai, J., Parkes, D., and Procaccia, A. (2010), "Truth, Justice, and Cake Cutting," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, USA.

Chernoff, H. (1952), "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations," *Annals of Mathematical Statistics*, 23, 493–507.

Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaitre, M., Maudet, N., Padget, J., Phelps, S., Rodríguez-Aguilar, J. A., and Sousa, P. (2006), "Issues in Multiagent Resource Allocation," *Informatica*, 30, 3–31.

Chevaleyre, Y., Endriss, U., Lang, J., and Maudet, N. (2007), "A Short Introduction to Computational Social Choice," in *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science (SOFSEM-2007)*.

Chevaleyre, Y., Lang, J., Maudet, N., and Ravilly-Abadie, G. (2009), "Compiling the votes of a subelectorate," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 97–102, Pasadena, CA, USA.

Chevaleyre, Y., Lang, J., Maudet, N., Monnot, J., and Xia, L. (2010a), "New Candidates Welcome! Possible Winners with respect to the Addition of New Candidates," Tech. Rep. 302, Université Paris-Dauphine.

Chevaleyre, Y., Lang, J., Maudet, N., and Monnot, J. (2010b), "Possible Winners when New Candidates Are Added: The Case of Scoring Rules," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 762–767, Atlanta, GA, USA.

Clarke, E. H. (1971), "Multipart pricing of public goods," *Public Choice*, 11, 17–33.

Cohler, Y. J., Lai, J. K., Parkes, D. C., and Procaccia, A. D. (2011), "Optimal Envy-Free Cake Cutting," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, San Francisco, CA, USA.

Coleman, T. and Teague, V. (2007), "On the complexity of manipulating elections," in *Proceedings of the thirteenth Australasian symposium on Theory of computing (CATS)*, pp. 25–33.

Condorcet, M. d. (1785), "Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix," Paris: L'Imprimerie Royale.

Conitzer, V. (2006a), "Computational aspects of preference aggregation," Ph.D. thesis, Carnegie Mellon University.

Conitzer, V. (2006b), "Computing Slater Rankings Using Similarities among Candidates," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 613–619, Boston, MA, USA, Early version appeared as IBM RC 23748, 2005.

Conitzer, V. (2008), "Anonymity-Proof Voting Rules," in *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pp. 295–306, Shanghai, China.

Conitzer, V. (2009), "Eliciting Single-Peaked Preferences Using Comparison Queries," *Journal of Artificial Intelligence Research*, 35, 161–191, Early version in AAMAS-07.

Conitzer, V. (2010), "Comparing Multiagent Systems Research in Combinatorial Auctions and Voting," *Annals of Mathematics and Artificial Intelligence*, To appear.

Conitzer, V. (2011), "Should Social Network Structure Be Taken into Account in Elections?" *Mathematical Social Sciences*, to appear.

Conitzer, V. and Sandholm, T. (2002), "Vote Elicitation: Complexity and Strategy-Proofness," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 392–397, Edmonton, AB, Canada.

Conitzer, V. and Sandholm, T. (2003), "Universal Voting Protocol Tweaks to Make Manipulation Hard," in *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 781–788, Acapulco, Mexico.

Conitzer, V. and Sandholm, T. (2005a), "Common Voting Rules as Maximum Likelihood Estimators," in *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pp. 145–152, Edinburgh, UK.

Conitzer, V. and Sandholm, T. (2005b), "Communication Complexity of Common Voting Rules," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 78–87, Vancouver, BC, Canada.

Conitzer, V. and Sandholm, T. (2006), "Nonexistence of Voting Rules That Are Usually Hard to Manipulate," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 627–634, Boston, MA, USA.

Conitzer, V. and Yokoo, M. (2010), "Using Mechanism Design to Prevent False-Name Manipulations," *AI Magazine*, 31, 65–77, Special Issue on Algorithmic Game Theory.

Conitzer, V., Davenport, A., and Kalagnanam, J. (2006), "Improved Bounds for Computing Kemeny Rankings," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 620–626, Boston, MA, USA.

Conitzer, V., Sandholm, T., and Lang, J. (2007), "When Are Elections with Few Candidates Hard to Manipulate?" *Journal of the ACM*, 54, Article 14, 1–33, Early versions in AAAI-02 and TARK-03.

Conitzer, V., Lang, J., and Xia, L. (2009a), "How hard is it to control sequential elections via the agenda?" in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 103–108, Pasadena, CA, USA.

Conitzer, V., Rognlie, M., and Xia, L. (2009b), "Preference Functions That Score Rankings and Maximum Likelihood Estimation," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 109–115, Pasadena, CA, USA.

Conitzer, V., Immorlica, N., Letchford, J., Munagala, K., and Wagman, L. (2010), "False-Name-Proofness in Social Networks," in *Proceedings of the Sixth Workshop on Internet and Network Economics (WINE)*, Stanford, CA, USA.

Conitzer, V., Walsh, T., and Xia, L. (2011a), "Dominating Manipulations in Voting with Partial Information," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, San Francisco, CA, USA.

Conitzer, V., Lang, J., and Xia, L. (2011b), "Hypercubewise Preference Aggregation in Multi-issue Domains," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Davies, J., Katsirelos, G., Narodytska, N., and Walsh, T. (2011), "Complexity of and Algorithms for BordaManipulation," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, San Francisco, CA, USA.

Dekel, E. and Piccione, M. (2000), "Sequential Voting Procedures in Symmetric Binary Elections," *Journal of Political Economy*, 108, 34–55.

Desmedt, Y. and Elkind, E. (2010), "Equilibria of plurality voting with abstentions," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 347–356, Cambridge, MA, USA.

Dobzinski, S. and Procaccia, A. D. (2008), "Frequent Manipulability of Elections: The Case of Two Voters," in *Proceedings of the Fourth Workshop on Internet and Network Economics (WINE)*, pp. 653–664, Shanghai, China.

Domshlak, C. and Brafman, R. I. (2002), "CP-nets - reasoning and consistency testing," in *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning (KR-02)*, pp. 121–132, Toulouse, France.

Drissi-Bakhkhat, M. and Truchon, M. (2004), "Maximum Likelihood Approach to Vote Aggregation with Variable Probabilities," *Social Choice and Welfare*, 23, 161–185.

Durrett, R. (1991), *Probability: Theory and Examples.*

Dutta, B. and Sen, A. (1993), "Implementing generalized Condorcet social choice functions via backward induction," *Social Choice and Welfare*, 10, 149–160.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001), "Rank aggregation methods for the web," in *Proceedings of the 10th World Wide Web Conference*, pp. 613–622.

Elkind, E. and Lipmaa, H. (2005), "Hybrid Voting Protocols and Hardness of Manipulation," in *Annual International Symposium on Algorithms and Computation (ISAAC)*, pp. 24–26.

Elkind, E., Faliszewski, P., and Slinko, A. (2009a), "On distance rationalizability of some voting rules," in *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 108–117.

Elkind, E., Faliszewski, P., and Slinko, A. (2009b), "Swap Bribery," in *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pp. 299–310.

Elkind, E., Faliszewski, P., and Slinko, A. (2010a), "Cloning in Elections," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 768–773, Atlanta, GA, USA.

Elkind, E., Faliszewski, P., and Slinko, A. (2010b), "Good Rationalizations of Voting Rules," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 774–779, Atlanta, GA, USA.

Elkind, E., Faliszewski, P., and Slinko, A. (2010c), "On the role of distances in defining voting rules," in *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 375–382, Toronto, Canada.

Elkind, E., Faliszewski, P., and Slinko, A. (2011), "Rationalizations of Condorcet-consistent rules via distances of hamming type," *Social Choice and Welfare. To appear.*

Endriss, U., Grandi, U., and Porello, D. (2010a), "Complexity of judgment aggregation: safety of the agenda," in *Proceedings of the Ninth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 359–366.

Endriss, U., Grandi, U., and Porello, D. (2010b), "Complexity of Winner Determination and Strategic Manipulation in Judgment Aggregation," in *Proceedings of the 3rd International Workshop on Computational Social Choice (COMSOC-2010).*

Ephrati, E. and Rosenschein, J. S. (1991), "The Clarke Tax as a Consensus Mechanism Among Automated Agents," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 173–178, Anaheim, CA, USA.

Erdös, P. and Moser, L. (1964), "On the Representation of Directed Graphs as Unions of Orderings," *Math. Inst. Hung. Acad. Sci.*, 9, 125–132.

Estivie, S. (2007), "Allocation de Ressources Multi-Agents: Théorie et Pratique," Ph.D. thesis, Université Paris Dauphine, LAMSADE.

Faliszewski, P. (2008), "Manipulation of Elections: Algorithms and Infeasibility Results," Ph.D. thesis, University of Rochester.

Faliszewski, P. and Procaccia, A. D. (2010), "AI's War on Manipulation: Are We Winning?" *AI Magazine*, 31, 53–64.

Faliszewski, P., Hemaspaandra, E., and Schnoor, H. (2008), "Copeland voting: ties matter," in *Proceedings of AAMAS '08*, pp. 983–990.

Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. A. (2009a), "How hard is bribery in elections?" *Journal of Artificial Intelligence Research*, 35, 485–532.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., and Rothe, J. (2009b), "Llull and Copeland voting computationally resist bribery and constructive control," *Journal of Artificial Intelligence Research*, 35, 275–341.

Faliszewski, P., Hemaspaandra, E., and Schnoor, H. (2010a), "Manipulation of copeland elections," in *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 367–374.

Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. A. (2010b), "Using complexity to protect elections," *Commun. ACM*, 53, 74–82.

Faliszewski, P., Hemaspaandra, E., and Hemaspaandra, L. A. (2011a), "Multimode Control Attacks on Elections," *Journal of Artificial Intelligence Research*, 40, 305–351.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L. A., and Rothe, J. (2011b), "The shield that never was: Societies with single-peaked preferences are more open to manipulation and control," *Information and Computation*, 209, 89–107.

Farfel, J. and Conitzer, V. (2011), "Aggregating value ranges: preference elicitation and truthfulness," *Autonomous Agents and Multi-Agent Systems*, pp. 127–150.

Farquharson, R. (1969), *Theory of Voting*, Yale University Press.

Fortune, S., Hopcroft, J. E., and Wyllie, J. (1980), "The Directed Subgraph Homeomorphism Problem," *Theoretical Computer Science*, 10, 111–121.

Friedgut, E., Kalai, G., and Nisan, N. (2008), "Elections can be Manipulated Often," in *Proceedings of the Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 243–249.

Fudenberg, D. and Tirole, J. (1991), *Game theory*, MIT Press, Cambridge, MA.

Ghosh, S., Mundhe, M., Hernandez, K., and Sen, S. (1999), "Voting for movies: the anatomy of a recommender system," in *Proceedings of the third annual conference on Autonomous Agents*, pp. 434–435.

Gibbard, A. (1973), "Manipulation of voting schemes: a general result," *Econometrica*, 41, 587–601.

Gibbard, A. (1977), "Manipulation of schemes that mix voting with chance," *Econometrica*, 45, 665–681.

Goldsmith, J., Lang, J., Truszczynski, M., and Wilson, N. (2008), "The Computational Complexity of Dominance and Consistency in CP-Nets," *Journal of Artificial Intelligence Research*, 33, 403–432.

Gonzales, C., Perny, P., and Queiroz, S. (2008), "Preference Aggregation with Graphical Utility Models," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 1037–1042.

Gonzalez, T. and Sahni, S. (1978), "Preemptive Scheduling of Uniform Processor Systems," *Journal of the ACM*, 25, 92–101.

Grandi, U. and Endriss, U. (2011), "Binary Aggregation with Integrity Constraints," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*.

Gretlei, R. J. (1983), "Dominance elimination procedures on finite alternative games," *International Journal of Game Theory*, 12, 107–113.

Groves, T. (1973), "Incentives in Teams," *Econometrica*, 41, 617–631.

Guo, M. (2010), "Computationally Feasible Approaches to Automated Mechanism Design," Ph.D. thesis, Duke University.

Hazon, N. and Elkind, E. (2010), "Complexity of safe strategic voting," in *Proceedings of the First Symposium on Algorithmic Game Theory (SAGT-08)*, pp. 210–221.

Hemaspaandra, E. and Hemaspaandra, L. A. (2007), "Dichotomy for Voting Systems," *Journal of Computer and System Sciences*, 73, 73–83.

Hemaspaandra, E., Hemaspaandra, L. A., and Rothe, J. (1997), "Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP," *Journal of the ACM*, 44, 806–825.

Hemaspaandra, E., Spakowski, H., and Vogel, J. (2005), "The complexity of Kemeny elections," *Theoretical Computer Science*, 349, 382–391.

Hodge, J. K. (2006), "Permutations of separable preference orders," *Discrete Applied Mathematics*, 154, 1478–1499.

Hudry, O. (2006), "Improvements of a Branch and Bound Method to Compute the Slater Orders of Tournaments," Tech. rep., ENST.

Ianovski, E., Yu, L., Elkind, E., and Wilson, M. C. (2011), "The Complexity of Safe Manipulation under Scoring Rules," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Isaksson, M., Kindler, G., and Mossel, E. (2010), "The Geometry of Manipulation: A Quantitative Proof of the Gibbard-Satterthwaite Theorem," in *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pp. 319–328, Washington, DC, USA.

Ju, B.-G. (2003), "A characterization of strategy-proof voting rules for separable weak orderings," *Social Choice and Welfare*, 21, 469–499.

Kadane, J. B. (1972), "On division of the question," *Public Choice*, 13, 47–54.

Kemeny, J. (1959), "Mathematics without Numbers," *Daedalus*, 88, 575–591.

Konczak, K. and Lang, J. (2005), "Voting procedures with incomplete preferences," in *Multidisciplinary Workshop on Advances in Preference Handling*.

Koutsoupias, E. and Papadimitriou, C. H. (1999), "Worst-case equilibria," in *Symposium on Theoretical Aspects in Computer Science*, pp. 404–413.

Lacy, D. and Niou, E. M. (2000), "A Problem with Referendums," *Journal of Theoretical Politics*, 12, 5–31.

Laffond, G., Laslier, J. F., and Breton, M. L. (1995), "Condorcet choice correspondences: A set-theoretical comparison," *Mathematical Social Sciences*, 30, 23–35.

Lang, J. (2007), "Vote and Aggregation in Combinatorial Domains with Structured Preferences," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 1366–1371, Hyderabad, India.

Lang, J. and Xia, L. (2009), "Sequential composition of voting rules in multi-issue domains," *Mathematical Social Sciences*, 57, 304–324.

Laslier, J.-F. (1997), *Tournament Solutions and Majority Voting*, Springe.

Le Breton, M. and Sen, A. (1999), "Separable Preferences, Strategyproofness, and Decomposability," *Econometrica*, 67, 605–628.

LeGrand, R. (2008), "Computational Aspects of Approval Voting and Declared-Strategy Voting," Ph.D. thesis, Washington University in St. Louis.

Li, M., Vo, Q. B., and Kowalczyk, R. (2011), "Majority-rule-based preference aggregation on multi-attribute domains with structured preferences," in *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*.

Lipton, R., Markakis, E., Mossel, E., and Saberi, A. (2004), "On approximately fair allocations of indivisible goods," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 125–131, New York, NY, USA.

List, C. and Puppe, C. (2009), *The handbook of rational and social choice*, chap. Judgment aggregation: a survey, Oxford University Press.

May, K. (1952), "A set of independent necessary and sufficient conditions for simple majority decision," *Econometrica*, 20, 680–684.

McGarvey, D. C. (1953), "A Theorem on the Construction of Voting Paradoxes," *Econometrica*, 21, 608–610.

McKelvey, R. D. and Niemi, R. G. (1978), "A multistage game representation of sophisticated voting for binary procedures," *Journal of Economic Theory*, 18, 1 – 22.

Meskanen, T. and Hannu, N. (2008), *Power, freedom, and voting*, chap. Closeness counts in social choice, Springer-Verlag Berlin Heidelberg.

Mossel, E. and Tamuz, O. (2010), "Truthful fair division," in *Proceedings of the Third International Symposium on Algorithmic game theory*, pp. 288–299.

Moulin, H. (1979), "Dominance Solvable Voting Schemes," *Econometrica*, 47, 1137–51.

Moulin, H. (1980), "On strategy-proofness and single peakedness," *Public Choice*, 35, 437–455.

Moulin, H. (1991), *Axioms of Cooperative Decision Making*, Cambridge University Press.

Narodytska, N., Walsh, T., and Xia, L. (2011), "Manipulation of Nanson's and Baldwin's rule," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, San Francisco, CA, USA.

Nehring, K. and Puppe, C. (2007), "Efficient and strategy-proof voting rules: A characterization," *Games and Economic Behavior*, 59, 132–153.

Niou, E. M. S. (1987), "A note on Nanson's rule," *Public Choice*, 54, 191–193.

Obraztsova, S. and Elkind, E. (2011), "On the Complexity of Voting Manipulation under Randomized Tie-Breaking," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Obraztsova, S., Elkind, E., and Hazon, N. (2011), "Ties Matter: Complexity of Voting Manipulation Revisited," in *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Taipei, Taiwan.

Pearl, J. (1988), *Probabilistic reasoning in intelligent systems: Networks of plausible inference*, Morgan Kaufmann Publishers Inc.

Peleg, B. (1979), "A note on manipulability of large voting schemes," *Theory and Decision*, 11, 401–412.

Pennock, D. M. and Xia, L. (2011), "Price Updating in Combinatorial Prediction Markets with Bayesian Networks," in *Proceedings of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Barcelona, Spain.

Pennock, D. M., Horvitz, E., and Giles, C. L. (2000), "Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 729–734, Austin, TX, USA.

Pinedo, M. L. (2008), *Scheduling: Theory, Algorithms, and Systems*, Springer.

Pini, M. S. (2007), "Reasoning with Preferences and Uncertainty," Ph.D. thesis, University of Padova.

Pini, M. S., Rossi, F., Venable, K. B., and Walsh, T. (2007), "Incompleteness and incomparability in preference aggregation," in *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence (IJCAI)*, Hyderabad, India.

Procaccia, A. (2008), "Computational Voting Theory: Of the Agents, By the Agents, For the Agents," Ph.D. thesis, Hebrew University of Jerusalem.

Procaccia, A. D. (2009), "Thou shalt covet thy neighbor's cake," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 239–244.

Procaccia, A. D. and Rosenschein, J. S. (2007a), "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)*, pp. 718–720, Honolulu, HI, USA.

Procaccia, A. D. and Rosenschein, J. S. (2007b), "Junta Distributions and the Average-Case Complexity of Manipulating Elections," *Journal of Artificial Intelligence Research (JAIR)*, 28, 157–181.

Purrington, K. and Durfee, E. H. (2007), "Making social choices from individuals' CP-nets," in *Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pp. 1122–1124.

Rossi, F., Venable, K. B., and Walsh, T. (2004), "mCP nets: Representing and reasoning with preferences of multiple agents," in *Proceedings of AAAI-04*, pp. 729–734.

Satterthwaite, M. (1975), "Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions," *Journal of Economic Theory*, 10, 187–217.

Scarsini, M. (1998), "A strong paradox of multiple elections," *Social Choice and Welfare*, 15, 237–238.

Schwartz, T. (1970), "On the Possibility of Rational Policy Evaluation," *Theory and Decision*, 1, 89–106.

Schwartz, T. (1977), "Collective Choice, Separation of Issues and Vote Trading," *The American Political Science Review*, 71, 999–1010.

Shoham, Y. and Leyton-Brown, K. (2009), *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*, Cambridge University Press.

Slinko, A. (2002), "On asymptotic strategy-proofness of classical social choice rules," *Theory and Decision*, 52, 389–398.

Slinko, A. (2004), "How large should a coalition be to manipulate an election?" *Mathematical Social Sciences*, 47, 289–293.

Slinko, A. and White, S. (2008), "Non-dictatorial social choice rules are safely manipulable," in *Proceedings of the Second International Workshop on Computational Social Choice (COMSOC)*, pp. 403–413.

Sloth, B. (1993), "The Theory of Voting and Equilibria in Noncooperative Games," *Games and Economic Behavior*, 5, 152–169.

Steinhaus, H. (1948), "The problem of fair division," *Econometrica*, 16, 101–104.

Truchon, M. (2008), "Borda and the maximum likelihood approach to vote aggregation," *Mathematical Social Sciences*, 55, 96–102.

Uckelman, J. (2009), "More than the Sum of its Parts: Compact Preference Representation over Combinatorial Domains," Ph.D. thesis, University of Amsterdam.

Vickrey, W. (1961), "Counterspeculation, Auctions, and Competitive Sealed Tenders," *Journal of Finance*, 16, 8–37.

Wagman, L. and Conitzer, V. (2008), "Optimal False-Name-Proof Voting Rules with Costly Voting," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 190–195, Chicago, IL, USA.

Walsh, T. (2007), "Uncertainty in preference elicitation and aggregation," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 3–8, Vancouver, BC, Canada.

Walsh, T. (2009), "Where are the really hard manipulation problems? The phase transition in manipulating the veto rule," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 324–329, Pasadena, CA, USA.

Walsh, T. (2010), "An Empirical Study of the Manipulability of Single Transferable Voting," in *Proceedings of the European Conference on Artificial Intelligence (ECAI)*, pp. 257–262.

Wing, J. M. (2006), "Computational thinking," *Communications of the ACM*, 49, 33–35.

Wooldridge, M. J. (2009), *An Introduction to MultiAgent Systems*, John Wiley & Sons, 2nd edn.

Xia, L. (2010), "Computational Social Choice: Strategic and Combinatorial Aspects," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 2000–2001, Atlanta, GA, USA.

Xia, L. and Conitzer, V. (2008a), "Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 196–201, Chicago, IL, USA.

Xia, L. and Conitzer, V. (2008b), "Generalized Scoring Rules and the Frequency of Coalitional Manipulability," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 109–118, Chicago, IL, USA.

Xia, L. and Conitzer, V. (2008c), "A Sufficient Condition for Voting Rules to Be Frequently Manipulable," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 99–108, Chicago, IL, USA.

Xia, L. and Conitzer, V. (2009), "Finite Local Consistency Characterizes Generalized Scoring Rules," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 336–341, Pasadena, CA, USA.

Xia, L. and Conitzer, V. (2010a), "Compilation Complexity of Common Voting Rules," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 915–920, Atlanta, GA, USA.

Xia, L. and Conitzer, V. (2010b), "Stackelberg Voting Games: Computational Aspects and Paradoxes," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 921–926, Atlanta, GA, USA.

Xia, L. and Conitzer, V. (2010c), "Strategy-proof Voting Rules over Multi-Issue Domains with Restricted Preferences," in *Proceedings of the Sixth Workshop on Internet and Network Economics (WINE)*, pp. 402–414, Stanford, CA, USA.

Xia, L. and Conitzer, V. (2011a), "Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders," *Journal of Artificial Intelligence Research*, 41, 25–67.

Xia, L. and Conitzer, V. (2011b), "A Maximum Likelihood Approach towards Aggregating Partial Orders," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Xia, L. and Lang, J. (2009), "A Dichotomy Theorem on the Existence of Efficient or Neutral Sequential Voting Correspondences," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 342–347, Pasadena, CA, USA.

Xia, L. and Pennock, D. M. (2011), "An Efficient Monte-Carlo Algorithm for Pricing Combinatorial Prediction Markets for Tournaments," in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI)*, Barcelona, Catalonia, Spain.

Xia, L., Lang, J., and Ying, M. (2007a), "Sequential voting rules and multiple elections paradoxes," in *Theoretical Aspects of Rationality and Knowledge (TARK)*, pp. 279–288, Brussels, Belgium.

Xia, L., Lang, J., and Ying, M. (2007b), "Strongly decomposable voting rules on multiattribute domains," in *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pp. 776–781, Vancouver, BC, Canada.

Xia, L., Zuckerman, M., Procaccia, A. D., Conitzer, V., and Rosenschein, J. (2009), "Complexity of unweighted coalitional manipulation under some common voting rules," in *Proceedings of the Twenty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 348–353, Pasadena, CA, USA.

Xia, L., Conitzer, V., and Procaccia, A. D. (2010), "A Scheduling Approach to Coalitional Manipulation," in *Proceedings of the ACM Conference on Electronic Commerce (EC)*, pp. 275–284, Cambridge, MA, USA.

Xia, L., Lang, J., and Monnot, J. (2011), "Possible Winners When New Alternatives Join: New Results Coming Up!" in *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Taipei, Taiwan.

Yokoo, M., Sakurai, Y., and Matsubara, S. (2004), "The Effect of False-name Bids in Combinatorial Auctions: New Fraud in Internet Auctions," *Games and Economic Behavior*, 46, 174–188.

Young, H. P. (1975), "Social Choice Scoring Functions," *SIAM Journal on Applied Mathematics*, 28, 824–838.

Young, H. P. (1988), "Condorcet's Theory of Voting," *American Political Science Review*, 82, 1231–1244.

Young, H. P. (1995), "Optimal Voting Rules," *Journal of Economic Perspectives*, 9, 51–64.

Zuckerman, M., Procaccia, A. D., and Rosenschein, J. S. (2009), "Algorithms for the coalitional manipulation problem," *Artificial Intelligence*, 173, 392–412.

Zuckerman, M., Lev, O., and Rosenschein, J. S. (2011), "An Algorithm for the Coalitional Manipulation Problem under Maximin," in *Proceedings of the Tenth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Taipei, Taiwan.

Biography

Lirong Xia was born on July 29, 1982 in Fengxin, Jiangxi Province, China. In 2004 he received a B.E. from the Department of Computer Science and Technology, Tsinghua University, Beijing, China. From 2004 to 2007 he worked as a research associate in the Department of Computer Science and Technology at Tsinghua University. In 2007 he joined the Ph.D. program of the Department of Computer Science at Duke University, and was awarded a James B. Duke Fellowship. He received an M.A. from the Department of Economics, Duke University in 2010. He will join Harvard Center for Research on Computation and Society (CRCS) as a postdoctoral researcher in September, 2011. He was awarded the Outstanding Research Initiation Project Award from the Department of Computer Science, Duke University in 2009, and was a finalist of the 2010 Facebook Ph.D. Fellowship Program.