A Deep Exploration of the Complexity Border of Strategic Voting Problems

Yongjie Yang

Dissertation zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultäten
der Universität des Saarlandes

Saarbrücken
2015

Tag des Kolloquiums	2. November 2015
Dekan	Prof. Dr. Markus Bläser Saarland University Saarbrücken, Germany
Vorsitzender des Prüfungsausschusses	Prof. Dr. Hans-Peter Lenhof
Berichterstatter	Prof. Dr. Jiong Guo Prof. Dr. Raimund Seidel Prof. Dr. Piotr Faliszewski
Akademischer Beisitzer	Dr. Asad Sayeed

Zusammenfassung

Abstimmungen werden auf verschiedene Gebiete angewendet. Leider kann es bei einer Abstimmung einzelne Teilnehmer geben, die Vorteile daraus ziehen, die Wahl durch strategisches Verhalten zu manipulieren. Eine Möglichkeit diesem Problem zu begegnen ist es, die Berechnungskomplexität als Hindernis gegen strategisches Verhalten zu nutzen. Die Annahme ist, dass falls es $\mathcal{NP}$-schwer ist, um strategisches Verhalten erfolgreich anzuwenden, der strategisch Handelnde vielleicht den Plan aufgibt die Abstimmung zu attackieren.

Diese Arbeit befasst sich mit strategischem Vorgehen in eingeschränkten Abstimmungen in dem Sinne, dass die vorgegebenen Abstimmungen kombinatorischen Einschränkungen unterliegen. Ziel ist es herauszufinden, wie sich die Komplexität des strategischen Handelns von dem sehr eingeschränkten zu dem generellen Fall ändert.

Kapitel 1 gibt einen Überblick zu der Arbeit.

Kapitel 2 diskutiert die Verhaltenskontrolle in ,,k-peaked" Abstimmungen. Insbesondere wird die Komplexität der Verhaltenskontrolle unter r-Approval, Condorcet, Copeland und Maximin Abstimmunssystemen studiert. k-peaked Abstimmungen verallgemeinern ,,single-peaked" Abstimmungen in der Art, dass höchstens k-Peaks in jeder Abstimmung auftauchen.

Kapitel 3 diskutiert die gleichen Probleme wie Kapitel 2, jedoch in Abstimmungen mit beschränkter single-peaked Breite. Intuitiv knnen, in einer Abstimmung mit single-peaked Breite k die Kandidaten gruppiert werden, wobei die Größe jeder Gruppe durch begrenzt ist k, und für jede Gruppe alle Wähler die gleiche Präferenz über alle Kandidaten in dieser Gruppe haben, im Vergleich zu Kandidaten, die nicht der Gruppe zugehören. Darüberhinaus, falls man jede Gruppe als ein Kandidat betrachtet, dann ist die Abstimmung single-peaked.

Kapitel 4 beschäftigt sich mit Bestechungsproblemen mittels Abstandseinschränkungen. In diesem Szenario darf ein korrumpierter Wähler eine neue Stimme abgeben die nahe an der Originalstimme liegen muss. In dieser Arbeit werden die bekannte Hamming-Distanz und die Kendall-Tau-Distanz verwendet um die Ähnlichkeit zu messen.

In Kapitel 5 betrachten wir Abstimmungen, die in einem Wettbewerb durchgeführt werden. Dabei sind die Kandidaten als Knoten in einem gerichteten Graphen dargestellt, wo eine Kante von einem Knoten a nach einem Knoten b die Bedeutung „a schlägt b". In diesem Zusammenhang heißt es, dass mehr Wähler für a als für b gestimmt haben. Die Gewinner werden in wohldefinierten Wettbewerbsverfahren ermittelt, z.B., Landau-Menge, Bank-Menge, usw. Besonderer Schwerpunkt liegt auf den Problemen bei der Ermittlung möglicher Gewinner. Die Fragestellung dabei ist, ob eine Untermenge von Knoten (Kandidaten) durch Hinzufügen oder Umkehren der Kanten in der Landau-Menge (Bank-Menge) hinzugefgt werden kann .

In Kapitel 6 werden gewichtete und ungewichtete Borda-Manipulationsprobleme betrachtet. Insbesondere leiten wir kombinatorische Algorithmen ab für den Fall einer erheblich eingeschränkten Kandidatenmenge und für den Fall einer eingeschränkten Menge von Maniupulatoren.

Kapitel 7 fasst die Ergebnisse zusammen und liefert Anhaltspunkte für weitere zukünftige Fragestellungen.

Preface

Voting has found applications in a variety of areas. Unfortunately, in a voting activity there may exist strategic individuals who have incentives to attack the election by performing some strategic behavior. One possible way to address this issue is to use computational complexity as a barrier against the strategic behavior. The point is that if it is $\mathcal{NP}$-hard to successfully perform a strategic behavior, the strategic individuals may give up their plan of attacking the election.

This thesis is concerned with strategic behavior in restricted elections, in the sense that the given elections are subject to some combinatorial restrictions. The goal is to find out how the complexity of the strategic behavior changes from the very restricted case to the general case.

In Chapter 1, we provide an overview of this thesis.

Chapter 2 is devoted to discussing control behavior in k-peaked elections. In particular, the complexity of control behavior for r-Approval, Condorcet, Copeland$^\alpha$ and Maximin is studied. k-peaked elections generalize single-peaked elections in the way that at most k peaks occur in each vote.

Chapter 3 is devoted to discussing the same problems as studied in Chapter 2, but in elections with bounded single-peaked width. Intuitively, in an election with single-peaked width k, the candidates can be grouped together, where the size of each group is bounded by k, and for each group, every voter has the same preferences over all candidates in this group compared to candidates not in the group. Moreover, if considering each group as a candidate, the election is single-peaked.

Chapter 4 is concerned with bribery problems with distance restrictions. In this scenario, every bribed voter can recast a new vote which needs to be as close as to its original vote. In this thesis, we adopt the prominent Hamming distance and the Kendall-Tau distance to measure the closeness.

In Chapter 5, we study elections which are performed on tournaments. In this scenario the candidates are represented by vertices, and there is an arc from a vertex a to a vertex b if a beats b in a pairwise comparison. Here, “a beats b” means that there are more voters who prefer a to b. The winners are selected according to some well-defined tournament solutions, e.g, Uncovered set, Banks set, etc. We focus on the possible winner(s) problems with respect to the Uncovered set and the Banks set. The input is a partial tournament and a vertex subset of the partial tournament, and the question is whether the given subset of vertices (candidates) can be included in the Uncovered set (Banks set) by adding/reversing some arcs.

In Chapter 6, we study the weighted and unweighted Borda manipulation problems. In particular, we derive combinatorial algorithms for both the case where the number of candidates is considerably small, and the case where the number of manipulators is considerably small.

Chapter 7 summarizes our results and provides some directions for future research.

Acknowledgments

It is a great pleasure to express my heartfelt thanks to all the people who have supported and helped me over the past four years.

My first special thanks goes to my supervisor Prof. Jiong Guo. I thank him for giving me the opportunity to work in his “Efficient Algorithms for Hard Problems” group which consists of many talented researchers like Matthias Mnich, Ondřej Suchý, Peng Sun and Yash Raj Shrestha. I also thank Prof. Jiong Guo for bringing me to the avenue of Computational Social Choice.

I thank my colleague Yash Raj Shrestha who shared the same office with me in the past four years. We are more like brothers other than colleagues. Besides sharing the office, we share happiness and sadness. I thank Matthias Mnich, Ondřej Suchý and Peng Sun for their helpful discussions. I also thank many excellent researchers from MPI (Max-Planck-Institut für Informatik) for their insightful discussions. Especially, I thank Geevarghese Philip and Kunal Dutta for the discussion of many interesting graph problems, and Marek Košta for the discussion of several kernelization algorithms.

I thank my master supervisors Prof. Jianxin Wang in Central South University and Prof. Jianer Chen in Texas A&M University. I thank their recommendation for my PhD study in Prof. Jiong Guo’s group. I also thank them for bringing me to the Parameterized Complexity Theory, from where I learnt many algorithm design techniques. These techniques have been proved very useful in exploring many problems studied in this thesis.

My PhD study was supported by CSC (China Scholarship Council), DFG (Deutsche Forschungsgemeinschaft) and MMCI (Cluster of Excellence “Multimodal Computing and Interaction”). I am thankful for their support.

I thank the anonymous reviewers of the publications that are incorporated in this thesis for their constructive comments.

I thank my wife Fang Xu. She made a lot of sacrifices in order to be with me. I appreciate everything that she has done for me. I thank my little sweetie daughter Emily Xinyan Yang whose birth made my life different.

I thank my parents, my grandparents, my brother and my sister for their encouragements and supports.

Finally, I thank the readers of this thesis for their interest in my work.

Contents

List of Tables

List of Figures

1

Introduction

Voting plays an important role in our daily life. You may have been involved in the following situations several times: deciding where to have a picnic; when should an exam to be held; who deserves to be the new leader; which brand of computer should you buy for your company. In all these situations, you are making collective decisions with other people involved. After the final decision is made, some people are happy with the result while others are not. If the result is predicable, the unhappy guys would have incentive to change the result to make themselves better off, perhaps by acting some strategic behavior. To achieve their goal, they need to figure out in a reasonable time how to act a strategic behavior to make themselves better off. Computational Social Choice (COMSOC) is concerned with this interesting topic of exploring the computation cost of performing a successful strategic behavior.

1.1 A Small Map

Life is full of making decisions: when to sell your stocks, where to hold your birthday party, how to get to school, with whom should you collaborate. In many real-world settings, however, making a decision is not only the business of yourself. For instance, making the decision of who is qualified as the president of a country involves all citizens who have the right to vote. In this situation, a collective decision is needed. With the rapid development of social media, people are more and more frequently involved in scenarios where collective decisions are needed to be done. Moreover, apart from arising in human communities, collective decision making also arises in many scenarios where no human being directly participates. For instance, in a multiagent system a collective decision would be made by a set of agents which might be robots, computers or something else.

One common and natural way to make a collective decision is by means of voting. In a voting activity, every voter casts his (for simplicity, we take the gender "male" for voters throughout this thesis) vote (or ballot) according to his preference over a given set of potential decisions (candidates), and then a designed voting rule works on the votes to make the final decision. Generally, different voting rules would lead to different results, while a good voting rule should be helpful in maximizing the benefits of the whole community involved. However, even we have a good voting rule in hand, there is no guarantee to get the expected result. One reason is the existence of strategic individuals who have incentive to change the result by acting some strategic behavior. To check this, take a look at the following example.

> A community is going to select a new leader from candidates A and B. Suppose that there are 10 community members and everyone is asked to give his vote to either A or B according to which candidate is preferred by himself. The candidate who gets the majority votes wins. Suppose further that 6 community members would give their votes to A, and 4 community members would give their votes to B. Then, A will be the winner. However, if the candidate B knows this information in advance, he can make himself the winner by carrying out the following strategy: persuade another candidate C who is very similar to A to compete with him and A. The similarity between A and C implies that every community member who prefers A to B also prefers C to B, and who prefers B to A also prefers B to C. Then in this case, it might be that 3 community members give their votes to A, 3 community members give their votes to C, and 4 community members give their votes to B, resulting in B to be the winner.

Due to the above discussion, a natural question arises: is there a strategy-proof voting rule? That is, a voting rule under which no one can change the result by performing a strategic behavior. The answer is "Yes": just consider the voting rule which always selects a fixed candidate as the winner, no matter what or how the voters vote. Nevertheless, a pertinent answer to this question is "no". This is because that all these voting rules which are strategy-proof cannot be reasonable voting rules (voting rules that satisfy a set of desirable criteria). In fact, there have been several impossibility theorems established in the middle of the last century, which state that under any reasonable voting rule the winners can be changed by performing a certain strategic behavior, say, misreporting true preferences (see e.g., [134, 135, 202, 225] for further details). Since the establishment of those impossibility theorems, the question of how to prevent voting from being attacked by means of strategic behavior had tantalized researchers for many years. A prominent answer to this question was given by Bartholdi, Tovey and Trick in the early 1990s [154, 155, 156, 157]. To address this issue of preventing voting from being attacked, they adopted the complexity as a barrier against strategic behavior. The point is that if performing a successful strategic behavior is $\mathcal{NP}$-hard, the strategic individuals may give up attacking the voting. Their work also sparked researchers to model more strategic behavior occurring in the real-world settings, such as the swap bribery by Elkind, Faliszewski and Slinko [98], the model of destructive control by Hemaspaandra, Hemaspaandra and Rothe [146], and the coalition weighted manipulation by Conitzer, Sandholm and Lang [68], to name a few. The efforts of the researchers finally foster the birth of the emerging area—Computational Social Choice (COMSOC).

Over the last decade, COMSOC has witnessed a significant development with more and more researchers joining the community. Moreover, quite a few papers are published in AI (stands for artificial intelligence) top conferences (e.g., AAAI, AAMAS, IJCAI, ECAI) and AI journals (e.g., Artificial Intelligence, Journal of Artificial Intelligence Research) every year. On the one hand, COMSOC fosters the development of many other research areas such as multiagent systems, political elections, recommendation systems, machine learning etc., due to its importance and practicability in these areas [209, 211, 212, 213, 229, 245]. On the other hand, COMSOC has independent interest on its own right. In the following, we first briefly introduce several hot topics in COMSOC. For each topic, we list a number of remarkable papers for the interested readers to get further details. See also Figure 1.1 for a small map of COMSOC. We hope that this brief introduction is helpful for readers who are getting ready to embark on related researches to quickly get the landscape of COMSOC. Then, we give a brief introduction to the problems studied in this thesis.

Since the conducted work of Bartholdi, Tovey and Trick [154, 155, 156, 157], COMSOC has been attracting an astonishing amount of attention from the theoretical computer science, artificial intelligence and social choice theory communities. Nowadays, COMSOC has been dominated by the following research directions: designing algorithms for voting problems, analyzing the complexity of voting problems, proposing

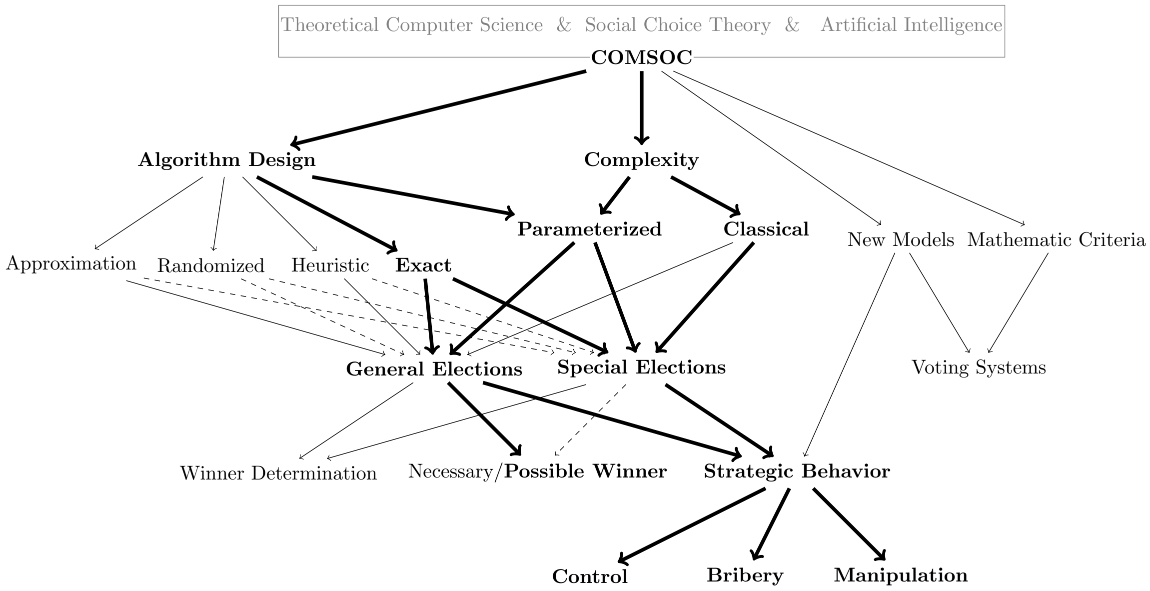


Figure 1.1: This is a small map of COMSOC showing research topics that have been receiving or will probably receive a considerable amount of attention of researchers. Hot topics that have been extensively studied by researchers are represented by plain arcs and dark arcs. The dashed arcs mean that the corresponding topics have not been extensively studied so far as this thesis is written, but are prominent topics for future research from our perspective. The lines of research that this thesis follows are represented by dark arcs.

new voting models and characterizing voting systems with mathematic criteria. The development of these directions also fosters each other.

Designing algorithms for voting problems lies in the core of COMSOC. On the one hand, faster algorithms are needed to solve many voting problems. For instance, a central question in a voting system is how to calculate the winners as fast as possible. On the other hand, algorithms can tell us, in another way different from the way that complexity theory does, whether a strategic voting problem is really hard to solve in real-world applications. For instance, even though the Borda manipulation problem has been proved $\mathcal{NP}$-hard [26, 75], Davies et al. [73] derived several heuristic algorithms for the problem, and showed that these algorithms perform quite well in elections that are created randomly. The algorithms designed for voting problems include non-exact algorithms, such as approximation algorithms, randomized algorithms and heuristic algorithms, as well as exact algorithms and parameterized algorithms[i]. See Figure 1.1 for an illustration. In spite of the importance of algorithm design for voting problems, many lines of research have not been extensively investigated, as indicated by the dashed arcs in Figure 1.1. But since COMSOC is still in its developing period, this does not indicate that these topics are not important. Conversely, we believe that

[i]In fact, a parameterized algorithm can also output exact solutions. However, parameterized complexity puts strong emphasize on parameters. Another reason we distinguish between parameterized algorithms and exact algorithms is that parameterized complexity has been commonly recognized as an independent research area.

these currently less-studied lines of research will attract a considerable amount of attention in the near future. The reasons are as follows. Randomized algorithms for voting problems have not attracted much attention since in many cases people desire to get determined results. However, as the proposal of many randomized voting systems very recently (see, e.g., [11, 13, 14, 217]), designing randomized algorithms for related voting problems will be of particular importance. Non-exact algorithms on special elections might also receive attention in the future due to the emerging of many hardness results of voting problems in special elections (see, e.g., [252, 253] the work wherein is part of the thesis, and [111]). For some other representative work on these research directions, we refer to [54, 57, 58, 74, 176, 248, 257, 258] for approximation algorithms, [72, 73, 143, 176, 181, 240] for heuristic algorithms, and [23, 79, 121, 163, 199, 241] for parameterized algorithms.

Analyzing complexity of voting problems is another hot topic in COMSOC. In this direction, researchers focus on the classical complexity of voting problems, where the main task is to prove whether the voting problem in hand is $\mathcal{NP}$-hard or polynomial-time solvable, as well as the parameterized complexity of voting problems, where the main task is to prove whether the voting problem in hand is $\mathcal{W}$-hard or $\mathcal{FPT}$. One aim of this line of research is to provide worst-case based evidence of the hardness of voting problems, such as election control, manipulation and bribery. The classical complexity of voting problems has been prevalent in the last decade, while the parameterized complexity counterpart has received a considerable amount of attention recently. For some representative work on this topic, we refer to [10, 79, 84, 99, 108, 109, 110, 112, 118, 184, 185, 199].

Proposing new voting models, such as new voting systems or new models of strategic voting behavior, is also an active topic in COMSOC. New voting models naturally emerge as new observations on real-world applications arise. For instance, based on the observation that in many real-world settings of partition of voters, one wants the two parts of the partition to be of (almost) equal size, or is partitioning into more than two parts, or has groups of actors who must be placed in the same part of the partition, Erdélyi, Hemaspaandra and Hemaspaandra [100] recently proposed several voter partition models which better capture many real-world applications. Some recently proposed voting models can be also found in [37, 63, 100, 141, 166, 207].

Characterizing voting systems with mathematic criteria has been a long studied topic in social choice theory. This line of research aims at providing useful guideline for people who desire to arise a voting. Some voting systems are better than others in a specific situation, according to the mathematic criteria that the voting systems hold. This line of research also provides critical properties of voting systems, which are useful in deriving algorithms or analyzing complexity of voting problems. We refer to [124, 204, 222, 255] for some representative early work, and refer to [9, 41, 96, 127, 141, 151, 227, 231, 247] for recent developments of this line of research. In particular, we refer to Figure 9.3 in [231] and Table 9.2 in [227] for summaries of many well-studied mathematic criteria of single-winner voting systems, and refer to Table 1 in [96] for a summary of some recently proposed mathematic criteria of multiwinner voting systems.

The voting problems that have been extensively studied in the literature include winner determination problem, possible/necessary winner problem and strategic behavior such as manipulation, control and bribery. These problems have been extensively studied in general elections (the domain of the preferences of the voters is not restricted) since the seminal work of Bartholdi, Tovey and Trick [154, 155, 156, 157]. Recently, voting problems in special elections (the domain of the preferences of the voters is restricted in some way) have attracted a considerable attention, see, e.g., [111, 113, 188]. Winner determination is intrinsic in voting: each voting ends up with a set of winners (or a single winner) being elected. Hardness of winner determination of a voting system impedes the practical applications of the voting system. Fortunately, only few of common voting systems are $\mathcal{NP}$-hard to determine the winners. Among them are the Dodgson voting system [118, 156, 145], the Kemeny voting system [147, 156] and the Young voting system [221]. The possible winner and the necessary winner problems were initialized by Konczak and Lang [170] in 2005. Both problems arise in the scenario where the information (normally refers to the information of votes) of the election is incomplete. The possible winners are then defined as all candidates that win an election which is extended from the given incomplete election, while the necessary winners are defined as all candidates that win every full extension of the given election. We refer to [170] for further details on the possible winner and the necessary winner problems. The manipulation problem is studied as early as the social choice theory. However, algorithmic and complexity analysis of manipulation problems were first studied by Bartholdi, Tovey and Trick [154] in 1989. In the manipulation problem, a set of voters (manipulators) who have not cast their votes yet attempt to change the winners by casting their votes in some way. The control problems were also introduced by Bartholdi, Tovey and Trick [157] in 1992, where an external agent (a strategic individual) wants to change the winners by modifying the vote set or the candidate set. In particular, Bartholdi, Tovey and Trick [157] considered the modification operations vote/candidate deletion/addition/partition. Later, several other modification operations were also studied by researchers [110, 112, 146]. We defer further detailed discussion on control to the next section. The study of the bribery problems was initialized by Faliszewski, Hemaspaandra and Hemaspaandra [107] in 2006. Many variants of the bribery problems studied in [107] were proposed and extensively studied by researchers [50, 98, 106, 108, 190]. In general, the bribery problems are concerned with how an external agent changes the winners by bribing the voters. A bribed voter need to recast his vote in the external agent's favor.

This thesis is devoted to making a contribution to this emerging area by exploring the complexity of strategic voting problems in some prominent voting systems. In particular, this thesis is concerned with strategic voting problems in a restricted way, in the sense that the given voting profiles satisfy some combinatorial properties. This study of restricted strategic voting problems is motivated by the observation that in many real-world settings the voters may cast their votes based on some common principles, which in turn leads to a voting profile that satisfies several combinatorial properties. For example, imagine a voting where residents who live on the same street are asked to vote for the location of a supermarket. If the votes cast by the voters are represented by linear orders over the candidates (potential locations of the supermarket), it is natural that every voter would rank the candidate which is nearest to his residence in the highest position. Moreover, the farther the other candidate located away from this ideal candidate, the lower it is ranked. The consequence of the above setting is an election that fits into the category of single-peaked domain, which has been extensively studied in the literature [34, 44, 60, 103, 111, 113, 129, 178, 232]. The lines of research that this thesis follows are depicted in Figure 1.1. A detailed description of the structure of this thesis is given in Section 1.4. Before that, in the following sections, we give definitions and notations that will be used throughout this thesis, as well as a brief introduction to the technique toolkit that we adopt to study the voting problems in this thesis.

1.2 Problem Statement

1.2.1 Voting Systems

In this chapter we shall formally introduce the definitions and notations on voting systems that will serve us throughout this thesis. We may introduce some additional notations in the latter chapters, on an ad hoc basis.

Multiset. A *multiset* $S = \{s_1, s_2, ..., s_{|S|}\}$ is a generalization of a set where objects of S are allowed to appear more than one time in S, that is, $s_i = s_j$ is allowed for $i \neq j$. An *element* of S is one copy of some object. We use $s \in_+ S$ to denote that s is an element of S. The *cardinality* of S denoted by $|S|$ is the number of elements contained in S. For two multisets A and B, we use $A \uplus B$ to denote the multiset containing all elements in A and B. Moreover, we use $A \boxminus B$ to denote the multiset containing for each object s, $\max\{0, n_1 - n_2\}$ copies of s, where n_1 and n_2 are the numbers of copies of s in A and B, respectively. A multiset B is a *submultiset* of a multiset A if for every object s that occurs n times in B, A contains at least n copies of s. We use $B \sqsubseteq A$ to denote that B is a submultiset of A.

Example. Consider multisets $A = \{1, 1, 1, 2, 3, 3, 4\}$ and $B = \{1, 2, 3\}$. Then we have

- The cardinalities of A and B are $|A| = 7, |B| = 3$, respectively;
- $A \uplus B = \{1, 1, 1, 1, 2, 2, 3, 3, 3, 4\}$;
- $A \uplus B = \{1, 1, 3, 4\}$; and
- $B \sqsubseteq A$.

Voting System. A *voting system* can be specified by a set $\mathcal{C}$ of *candidates*, a multiset[ii] $\Pi_{\mathcal{V}} = (\pi_{v_1}, \pi_{v_2}, ..., \pi_{v_n})$ of *votes* cast by a corresponding set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ of *voters* (π_{v_i} is cast by v_i), and a *voting correspondence* φ[iii] which maps the *election* $\mathcal{E} = (\mathcal{C}, \Pi_{\mathcal{V}}, \mathcal{V})$ to a nonempty set of candidates $\varphi(\mathcal{E})$, the *winners*. For simplicity, we often discard $\mathcal{V}$ from the above notation for election $\mathcal{E}$ when $\Pi_{\mathcal{V}}$ is sufficient to determine the winners (we will discuss a weighted voting scenario in Chapter 6, where each voter has a positive weight which is indispensable to determine the winners. In this case, we retain $\mathcal{V}$ in the notation. This is the only case we do so throughout this thesis). If there is only one winner, we call it a *unique winner*; otherwise we call them *co-winners*. Moreover, each vote $\pi_v \in_+ \Pi_{\mathcal{V}}$ is defined as a linear order over the candidates. Throughout this thesis, we interchangeably use the terms "vote" and "voter". The linear order of a vote is also called the *preference* of the vote over the candidates. For convenience, we use $\succ_v$ to denote the preference of the vote cast by the voter v. Therefore, for a voter v who prefers the candidate a to b to c, the vote will be written as $\pi_v : a \succ_v b \succ_v c$. We say that the voter v casts vote π_v with preference $a \succ_v b \succ_v c$. In context where $\succ_v$ is clearly known to be whose preference, we drop v from $\succ_v$. In many places in this thesis (especially in Sections 2 and 3), for ease of exposition, we also use curve braces with candidates listed inside to represent the preferences of the voters. For instance, $\pi_v = (a, b, c, d)$ is saying that the preference of the vote π_v cast by the voter v is $a \succ b \succ c \succ d$.

A candidate c is preferred to another candidate c' by a vote π_v if $c \succ_v c'$. We also say that c is *ranked above* c' in the vote. The *position* of a candidate c in a vote π_v, denoted as $pos_{\succ_v}(c)$ (or simply $pos_v(c)$), is defined as $|\{c' \mid c' \succ_v c\}| + 1$, the number of candidates that are ranked above c in the vote plus one.

For a vote π_v and a subset $C \subseteq \mathcal{C}$, let $\pi_v(C)$ denote the *partial vote* of π_v restricted to C, such that in $\pi_v(C)$ every two distinct candidates in C preserve the same order as in π_v. For example, for a vote π_v with preference $a \succ b \succ c \succ d \succ e$, the partial vote $\pi_v(\{b, d, e\})$ over the candidates b, d, e has preference $b \succ d \succ e$. For a multiset Π

[ii] In some literature, the votes are enclosed in a list other than a multiset. The reason for using multiset is twofold. On the one hand, from the mathematic point of view, multiset allows us to use operations generalized from the set theory. On the other hand, all the voting correspondences considered in this thesis are anonymous, which means that the winners do not change if we change the order of the voters; thus, we do not need the terminology "list" to emphasize the order of the voters.

[iii] A similar concept is voting rule which maps an election to a single candidate. A voting correspondence can be modified to a voting rule by using a tie-breaking method.

of votes and a subset $C \subseteq \mathcal{C}$, let $\Pi(C)$ be the multiset of votes obtained from Π by replacing each $\pi \in_+ \Pi$ by $\pi(C)$.

For two candidates c and c' in an election $\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V})$, let $N_\mathcal{E}(c, c')$ denote the number of votes which prefer c to c'. We drop the index $\mathcal{E}$ when it is clear from context. If $N_\mathcal{E}(c, c') > N_\mathcal{E}(c', c)$, we say c *beats* c' by $N_\mathcal{E}(c, c')$ in $\mathcal{E}$; otherwise if $N_\mathcal{E}(c, c') = N_\mathcal{E}(c', c)$ we say c *ties* c' in $\mathcal{E}$.

Voting Correspondences. We mainly study the following voting correspondences in this thesis.

- Positional scoring correspondences. Every candidate gets a specific score from each vote according to the position of the candidate in the vote. More specifically, a positional scoring voting correspondence is defined by a scoring vector $\vec{\lambda} = \langle \lambda_1, \lambda_2, ..., \lambda_m \rangle$ with $\lambda_1 \geq \lambda_2 \geq, ..., \geq \lambda_m$, where m is the number of candidates and each λ_i is a real number. The candidate ranked in the i-th position in a vote gets λ_i points from this vote. The winners are the candidates with the highest score. Following are some well-known positional scoring correspondences.

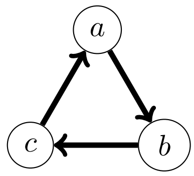
Name	Scoring Vectors
Borda	$\langle m-1, m-2, ..., 0 \rangle$
r-Approval	$\langle 1, ..., 1, 0, ..., 0 \rangle$ with exactly r many 1's.
Plurality	$\langle 1, 0, 0, ..., 0 \rangle$
Veto	$\langle 1, 1, ..., 1, 0 \rangle$

- Condorcet[iv]. A candidate in an election is a *Condorcet winner* if it beats every other candidate in the election. A candidate in an election is a *weak Condorcet winner* if it ties or beats every other candidate in the election. Note that an election may not have a Condorcet winner or a weak Condorcet winner. See Figure 1.2 for an example. However, if an election has a Condorcet winner, the Condorcet winner is unique.

- Maximin. The maximin score of a candidate c in an election $\mathcal{E}$ with candidate set $\mathcal{C}$ is defined as $\min_{c' \in \mathcal{C} \setminus \{c\}} N_\mathcal{E}(c, c')$. The winners are the candidates with the highest Maximin score.

- Copeland$^\alpha$. Each candidate is compared with every other candidate. In each comparison, the one which beats its rival gets one point and its rival gets zero points. If they are tied, both get α points. The winners are the candidates

[iv]Strictly speaking, Condorcet is not a voting correspondence since there could be no Condorcet winner or weak Condorcet winner in an election. Nevertheless, the concept of Condorcet winner and weak Condorcet winner plays significant role in many common voting correspondences, such as Kemeny, Young and Dodgson voting correspondences [22]. Moreover, complexity of making a given distinguished candidate (not) a Condorcet winner (weak Condorcet winner) by performing some strategic behavior has been widely studied in the literature. We list it here since we shall also study the complexity of strategic behavior with respect to Condorcet winner and weak Condorcet winner.



$a \succ b \succ c$
$b \succ c \succ a$
$c \succ a \succ b$

Figure 1.2: An election with three candidates a, b, c and three votes with preferences shown on the right side of the figure. The comparison between every two candidates is shown on the left-hand. An arc from a candidate c to another candidate c' means that c beats c'. It is clear that this election contains neither a Condorcet winner nor a weak Condorcet winner.

with the highest score. We remark that Copeland$^{0.5}$ is commonly referred to as Copeland, and Copeland^1, developed by the thirteenth-century mystic Llull, is referred to Llull voting in the literature [112].

For readers who are interested in voting correspondences, we refer to [231] for an excellent summary. Moreover, we refer to [179, 234] for two further auxiliary references for voting correspondences, where economic and political aspects of many voting correspondences are discussed with concrete examples.

1.2.2 Strategic Behavior

In this section, we introduce the strategic voting problems which will be studied in this thesis. We do not give all definitions and notations which will be used or studied in this thesis since listing all these definitions and notations is cumbersome and will tax the reader. Instead, we choose problems which we believe to be significant for readers to grasp the main idea of strategic voting problems quickly. Further definitions and notations concerning concrete problems are given in Chapters 2-6, on an ad hoc basis. We refer to [22] for a comprehensive survey for other strategic voting problems which are not considered in this thesis but have been also widely studied in the literature.

1.2.2.1 Control

Election control models the scenario where there is an external agent (e.g., the chairman of a committee) who attempts to influence the result of the election by doing some tricks. There would be two goals that the external agent wants to reach. One goal is to make a given distinguished candidate win the election. The other goal is to make the given distinguished candidate lose the election. The former case is called a *constructive control* and the latter case is called a *destructive control*. Moreover, the tricks involved in a control attack include adding some new, unregistered votes to the registered votes, deleting votes from the registered votes, adding new candidates to the election or deleting candidates from the election. The complexity of constructive control problems were first studied by Bartholdi, Tovey and Trick [157] in 1992, and the complexity of destructive control problems were first studied by Hemaspaandra, Hemaspaandra and Rothe [146] in 2007. In the following, we give the formal definitions of the control problems.

Problem definitions. Let φ be a designed voting correspondence. We first define the constructive control problems. The destructive counterpart is defined analogously. In all strategic voting problems studied in this thesis, we distinguish between the unique-winner model and the nonunique-winner model. In the *unique-winner model* of constructive control, we are asked to make a given distinguished candidate the unique winner (with the assumption that the distinguished candidate is not the unique winner in advance). However, in the *nonunique-winner model* of constructive control, we are only asked to make the distinguished candidate a winner (with the assumption that the distinguished candidate is not a winner in advance), in the sense that the distinguished candidate is the unique winner or one of the co-winners in the final election (obtained from the original election by performing a certain control behavior). The unique-winner model is indicated by "UNI" and the nonunique-winner model is indicated by "NON".

Constructive Control by Adding Votes under φ (CCAV-φ-UNI/NON)

Input: An election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is not the unique winner/a winner in $\mathcal{E}$, a multiset $\Pi_{\mathcal{T}}$ of unregistered votes and an integer $0 \leq \mathcal{R} \leq |\Pi_{\mathcal{T}}|$.

Question: Are there at most $\mathcal{R}$ votes $\Pi_{\mathcal{T}'}$ in $\Pi_{\mathcal{T}}$ such that p is the unique winner/a winner in the election $\mathcal{E}' = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}} \uplus \Pi_{\mathcal{T}'})$ under the voting correspondence φ?

Constructive Control by Deleting Votes under φ (CCDV-φ-UNI/NON)

Input: An election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is not the unique winner/a winner in $\mathcal{E}$, and an integer $0 \leq \mathcal{R} \leq |\Pi_{\mathcal{V}}|$.

Question: Are there at most $\mathcal{R}$ votes $\Pi_{\mathcal{T}}$ in $\Pi_{\mathcal{V}}$ such that p is the unique winner/a winner in the election $\mathcal{E}' = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}} \uplus \Pi_{\mathcal{T}})$ under the voting correspondence φ?

Constructive Control by Deleting Candidates under φ (CCDC-φ-UNI/NON)

Input: An election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is not the unique winner/a winner in $\mathcal{E}$, and an integer $0 \leq \mathcal{R} \leq |\mathcal{C}|$.

Question: Are there at most $\mathcal{R}$ candidates $C \subseteq \mathcal{C}$ such that p is the unique winner/a winner in the election $\mathcal{E}' = ((\mathcal{C} \cup \{p\}) \setminus C, \Pi_{\mathcal{V}}((\mathcal{C} \cup \{p\}) \setminus C))$ under the voting correspondence φ?

Constructive Control by Adding Candidates under φ (CCAC-φ-UNI/NON)

Input: An election $(\mathcal{C} \cup \mathcal{D} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is not the unique winner/a winner in $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}(\mathcal{C} \cup \{p\}))$, and an integer $0 \leq \mathcal{R} \leq |\mathcal{D}|$.

Question: Are there at most $\mathcal{R}$ candidates $D \subseteq \mathcal{D}$ such that p is the unique winner/a winner in the election $\mathcal{E}' = ((\mathcal{C} \cup D \cup \{p\}), \Pi_{\mathcal{V}}(\mathcal{C} \cup D \cup \{p\}))$ under the voting correspondence φ?

The destructive control problems are defined in the similar way with two differences. First, instead of making the distinguished candidate p the unique winner/a winner, the destructive control is to prevent the distinguished candidate from being the unique winner/a winner (corresponding to the unique-winner model/nonunique-winner model). Second, instead of assuming the distinguished candidate is not the unique winner/a winner in the given election $\mathcal{E}$, we assume that the distinguished candidate is the unique winner/a winner in $\mathcal{E}$ in the destructive control problems.

Even though Condorcet is not regarded as a voting correspondence in most of the literature (since there could exist no Condorcet winner), we still define the same problems as above for Condorcet. In particular, in the unique-winner model, the objective is to make the distinguished candidate the Condorcet winner or not the Condorcet winner, depending on whether the constructive control or the destructive control is discussed. On the other hand, in the nonunique-winner model, the objective is to make the distinguished candidate a weak Condorcet winner or not a weak Condorcet winner, depending on whether the constructive control or the destructive control is discussed. The study of Condorcet control can be dated back to the work of Bartholdi, Tovey and Trick [157].

We study control problems in Chapters 2 and 3. Concretely, we study control problems in multi-peaked elections in Chapter 2 and control problems in elections with bounded single-peaked width in Chapter 3. In particular, we consider r-Approval, Plurality, Condorcet, Copeland$^{\alpha}$ for every $0 \leq \alpha \leq 1$ and Maximin voting correspondences. We defer the definitions of single-peaked elections and multi-peaked elections to Chapter 2 and defer the definition of single-peaked width to Chapter 3.

1.2.2.2 Bribery

The bribery problem is concerned with the question whether a given distinguished candidate can become the unique winner/a winner (unique-winner model of constructive bribery/nonunique-winner model of constructive bribery), or not the unique winner/a winner (unique-winner model of destructive bribery/ nonunique-winner model of destructive bribery) by bribing a limited number of voters. Here, if a voter is bribed, the vote cast by the voter will be replaced with a new vote recast by the voter. The bribery problem and many of its variants were first studied by Faliszewski, Hemaspaandra and Hemaspaandra [107]. Formal definitions are as follows.

Constructive Bribery under φ (CB-φ-UNI/NON)

Input: An election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is not the unique winner/a winner in $\mathcal{E}$, and an integer $0 \leq \mathcal{R} \leq |\Pi_{\mathcal{V}}|$.

Question: Can we replace at most $\mathcal{R}$ votes $\Pi_{\mathcal{T}}$ in $\Pi_{\mathcal{V}}$ with $|\Pi_{\mathcal{T}}|$ many new votes $\Pi_{\mathcal{T}'}$ such that p is the unique winner/a winner in the election $\mathcal{E}' = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}} \uminus \Pi_{\mathcal{T}} \uplus \Pi_{\mathcal{T}'})$ under the voting correspondence φ?

Destructive Bribery under φ (DB-φ-UNI/NON)

Input: An election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is the unique winner/a winner in $\mathcal{E}$, and an integer $0 \leq \mathcal{R} \leq |\Pi_{\mathcal{V}}|$.

Question: Can we replace at most $\mathcal{R}$ votes $\Pi_{\mathcal{T}}$ in $\Pi_{\mathcal{V}}$ with $|\Pi_{\mathcal{T}}|$ many new votes $\Pi_{\mathcal{T}'}$ such that p is not the unique winner/a winner in the election $\mathcal{E}' = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}} \uminus \Pi_{\mathcal{T}} \uplus \Pi_{\mathcal{T}'})$ under the voting correspondence φ?

Bribery problems are studied in Chapter 4. In particular, we study distance restricted bribery problems which differ from the traditional bribery problems as discussed above in that each voter can only be bribed to recast a new vote which is similar to the original one. To this end, we adopt the Hamming distance and the Kendall-Tau distance to measure the similarity between two votes. We defer the formal definitions of the Hamming distance and the Kendall-Tau distance to Chapter 4.

1.2.2.3 Manipulation

In the manipulation problem, we are given an election and a set of voters who have not cast their votes yet. These voters who have not cast their votes form a coalition and attempt to change the result of the election. Due to this reason, they are given the name manipulators. The manipulation problem asks whether the manipulators can cast their votes in some way so that a given distinguished candidate becomes the unique winner/a winner after adding their votes to the election.

Manipulation under φ (CM-φ-UNI/NON)

Input: An election $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, where p is the distinguished candidate who is not the unique winner/a winner in $\mathcal{E}$, and a set of manipulators.

Question: Can the manipulators cast their votes in a way so that p becomes the unique winner/a winner after adding their votes to the election?

In some literature, the above problem is called *Constructive Manipulation*, where "Constructive" takes the same meaning as in the control and bribery problems. The *Destructive Manipulation* is also studied in the literature [68]. In this thesis, we study only the constructive manipulation since the destructive counterpart of the problems studied in this thesis straightforwardly turned out to be polynomial-time solvable.

In this thesis, the manipulation problem is studied in Chapter 6. We will also study weighted manipulation problem in Chapter 6. We defer further details on weighted voting to Chapter 6.

1.2.2.4 Possible Winner(s)

In many practical settings, we might not be able to access or get the full information of an election. The possible winner(s) problem is concerned with the question of which candidate(s) should be the winner(s) in the situation where only partial information of the election is provided. The principle is to extend the given partial election in some way, and examine winner(s) in the extended election. In Chapter 5, we will study possible winner(s) problem in partial tournaments under several prominent tournament solutions. We defer further notations and definitions to Chapter 5.

1.3 Toolkit

In this section, we briefly introduce classical complexity, parameterized complexity, Lenstra's theorem on integer linear programming, and dynamic programming.

1.3.1 Classical Complexity

In this section, we give a brief introduction to the classical complexity. Insightful discussion is not the focus of this thesis. Instead, we refer the interested readers to the textbook of Arora and Barak [4], or the textbook of Garey and Johnson [131] for a comprehensive understanding of computational complexity theory. For readers who want to quickly improve their intuitive ability to assess complexity, we refer to the survey by Tovey [236]. Readers who have been familiar with the concepts of $\mathcal{P}$, $\mathcal{NP}$ and $\mathcal{NP}$-hard can safely skip to the next section.

Generally, computational complexity measures how efficiently problems can be solved, and classifies the problems into complexity classes, such as $\mathcal{P}$ (polynomial-time solvable) and $\mathcal{NP}$-hard (nondeterministic polynomial-time hard) etc., accordingly. We need the concept of Turing machine to define the complexity classes.

A Turing machine works on a tape associated with a head which can read, write, and shift to left or to right. Turing machine can powerfully illustrate in a mathematic way how human beings solve real-world problems. The basic idea is to encode the instances of the problem in hand into strings, which are then written in the tape of a Turing machine. Then, the Turing machine imitates the procedure of how human beings deal with the instance by reading, writing the tape, shifting its head to the left or the right, or remaining its head unmoved. In a formal way, a *Turing machine* is defined by a 7-tuple $M = (Q, \Gamma, b, \sum, \delta, H, F)$, where

- Q is a finite, non-empty set of states.
- Γ is a finite, non-empty set of the tape symbols.
- $b \in \Gamma$ is the blank symbol.
- $\sum \subseteq \Gamma \setminus \{b\}$ is the set of input symbols.
- H consists of an initial state q_0 and a halt state q_1.
- $F \subseteq Q$ is the set of accepting states.
- $\delta : Q \setminus F \times \Gamma \to Q \times \Gamma \times \{L, R, N\}$ is a partial function called the *transition function*, where L is left shift, R is right shift, and N means remaining the head unmoved.

In a formal way, a *problem* is defined as a language $L \subseteq \sum^*$. Given an instance of a problem L, a Turing machine deals with the instance in the following way. First, the instance is written in the tape with the head pointing at the beginning of the instance and the state being set as the initial state q_0. Then, the head reads, writes or shifts according to the partial function δ. If the machine finally stops at an accepting state, we say that the Turing machine *accepts* the instance; otherwise if it stops at a state other than any of the accepting states, we say it *rejects* the instance.

Now we are ready to define the complexity class $\mathcal{P}$. A problem L is *polynomial-time solvable* if there exists a Turing machine such that for every instance $I \in L$ the Turing machine accepts I in a polynomial number of steps in the size of I, and for every instance $I' \notin L$ the Turing machine rejects I' in a polynomial number of steps in the size of I'. Here, each step means either a writing operation, reading operation or shifting operation of the head. The complexity class $\mathcal{P}$ includes all the problems which are solvable in polynomial time. The polynomial-time solvable problems are regarded as tractable by convention.

In real-world applications, people are often confronted with problems which seem not solvable in polynomial time. Many of these problems are related to the complexity class $\mathcal{NP}$, which stands for *nondeterministic polynomial-time solvable*. Here, "nondeterministic" refers to nondeterministic Turing machine which is different from the above defined Turing machine (which we call deterministic Turing machine afterward) in the partial function δ. Concretely, in the definition of nondeterministic Turing machine, the partial function is replaced with a *state relation* $\delta' \subseteq (Q \setminus F \times \Gamma) \times (Q \times \Gamma \times \{L, R, N\})$. Therefore, given a current state of the machine and the symbol the head reads at the moment, the next step is not determined—each $a \in Q \setminus F \times \Gamma$ may be mapped to more than one element in $Q \times \Gamma \times \{L, R, N\}$. A problem L is *nondeterministic polynomial-time solvable*, if there exists a nondeterministic Turing machine NTM such that for every instance $I \in L$, NTM can accept I in a polynomial number of steps in the size of I, and for every instance $I' \notin L$, NTM can reject I' in a polynomial number of steps in the size of I'. According to the above definitions, we have that $\mathcal{P} \subseteq \mathcal{NP}$.

Now we come to introduce the complexity class $\mathcal{NP}$-hard. Basically, every problem in $\mathcal{NP}$-hard is at least as hard as every problem in $\mathcal{NP}$. This is conveyed by the *many-one reduction* [4].

Definition 1.1 ([4]). *A problem Q is many-one reducible to another problem Q', denoted by $Q \leq_m Q'$, if for every instance x of Q there is a polynomial-time algorithm which takes x as input and returns an instance y of Q' as output. Moreover, the instance x is equivalent to the instance y, in the sense that x is a yes-instance of Q if and only if y is a yes-instance of Q'.*

The complexity class $\mathcal{NP}$-hard includes all the problems to which every problem in $\mathcal{NP}$ is many-one reducible. By convention, if a problem is in $\mathcal{NP}$-hard, we simply say that the problem is $\mathcal{NP}$-hard. The $\mathcal{NP}$-hard problems are regarded as *intractable* since they cannot be solved in polynomial time unless $\mathcal{P} = \mathcal{NP}$ which is widely believed not the case [4]. Another important complexity class is $\mathcal{NP}$-complete which is defined as the intersection of $\mathcal{NP}$-hard and $\mathcal{NP}$.

$$\mathcal{NP}\text{-complete} = \mathcal{NP}\text{-hard} \cap \mathcal{NP}$$

We refer to `http://www.nada.kth.se/~viggo/problemlist/compendium.html` maintained by Crescenzi and Kann, and `http://cgi.csc.liv.ac.uk/~ped/teachadmin/COMP202/annotated_np.html` maintained by Dunne for two lists of $\mathcal{NP}$-hard problems that are well-studied in diverse areas.

1.3.2 Parameterized Complexity

In the following, we briefly introduce the parameterized complexity. For a comprehensive understanding of parameterized complexity, we refer to the textbook of Downey and Fellows [88] and the textbook of Niedermeier [203]. Readers who are familiar with parameterized complexity can safely skip to the next section.

As we have seen in the previous section, there exist problems which cannot be solved in polynomial time unless $\mathcal{P} = \mathcal{NP}$. Here, the computational complexity is measured with respect to the whole input size. However, many problems are companied with several parameters which can significantly affect the computational complexity of the problem but are ignored in the classical complexity analysis. Parameterized complexity compensates this negligence by dealing with problems in two dimensions: a main part and a parameter. In essence, how a parameter affects the complexity of the problem is the main concern of parameterized complexity.

Parameterized complexity was firstly systematically studied by Downey and Fellows [87] (see [88] for the second version of this textbook released in 2013). In a formal way, a *parameterized problem* is a language in $\Sigma^* \times \Sigma^*$, where Σ is a finite alphabet. The first component is called the *main part* of the problem while the second component is called the *parameter* which normally is a positive integer. Parameterized problems have the following main hierarchy:

$$\mathcal{FPT} \subseteq \mathcal{W}[1] \subseteq \mathcal{W}[2] \subseteq, ..., \subseteq \mathcal{XP}$$

where $\mathcal{FPT}$ includes all parameterized problems which admit $O(f(\kappa) \cdot |I|^{O(1)})$-time algorithms, while $\mathcal{XP}$ includes all parameterized problems which admit $O(f(\kappa) \cdot |I|^{g(\kappa)})$-time algorithms. Here, I is the main part of the instance, κ is the parameter, and f and g are computable functions depending only on κ. There are also parameterized problems beyond $\mathcal{XP}$. For example, the κ-colorable problem which is to determine whether an undirected graph admits a proper κ-coloring of the vertices has no algorithm of the form $O(f(\kappa) \cdot |I|^{g(\kappa)})$, unless $\mathcal{P} = \mathcal{NP}$ [89]. These problems fall into the class of so-called *para*$\mathcal{NP}$-hard introduced by Flum and Grohe [125]. In a formal way, *para*$\mathcal{NP}$-hard includes all the parameterized problems that are $\mathcal{NP}$-hard for every fixed value of κ above some threshold K. Finally, classes between $\mathcal{FPT}$ and $\mathcal{XP}$ are defined based on $\mathcal{FPT}$-reductions.

Definition 1.2. *Given two parameterized problems Q and Q', an $\mathcal{FPT}$-reduction from Q to Q' is an algorithm that takes as input an instance (I, κ) of Q and outputs an instance (I', κ') of Q' such that*

(1) the algorithm runs in $f(\kappa) \cdot |I|^{O(1)}$ time, where f is a computable function;

(2) $(I, \kappa) \in Q$ if and only if $(I', \kappa') \in Q'$; and

(3) $\kappa' \leq g(\kappa)$, where g is a computable function.

A problem is $\mathcal{W}[i]$-hard if all problems in $\mathcal{W}[i]$ can be $\mathcal{FPT}$-reducible to the problem. From the practical point of view, $\mathcal{W}[1]$ is the basic class of parameterized problems which unlikely admit $\mathcal{FPT}$-algorithms.

Kernelization is a main technique to derive $\mathcal{FPT}$-algorithms. The formal definition of kernelization is as follows.

Definition 1.3. *A kernelization for a parameterized problem Q is a polynomial-time algorithm that reduces a given instance (I, κ) of Q to a new instance (I', κ') of Q such that*

(1) (I, κ) is a yes-instance if and only if (I', κ') is a yes-instance;

(2) $\kappa' \leq \kappa$; and

(3) $|I'| \leq f(\kappa)$, where f is a computable function.

The new instance (I', κ') is called the *problem kernel*, and the function $f(\kappa)$ is the *kernel size*. Moreover, if f is a polynomial function, we call (I', κ') a *polynomial kernel*. Intuitively, a kernelization shrinks the original instance to a new equivalent and size-bounded instance. It is folklore that a parameterized problem is $\mathcal{FPT}$ if and only if it has a kernelization (See Theorem 1.39 in [125] or Proposition 7.2 in [203] for formal proofs). For more background on kernelization, we refer to [29, 120, 142]. We refer to [59] by Cesati, and [76] by Haan and Szeider for compendiums of parameterized problems.

Kernelization has been widely used to solve real-world problems [1, 62, 128, 159]. For the purpose of using kernelization in practice, one desires to have a kernel as small as possible. Unfortunately, many $\mathcal{FPT}$ problems are not likely to admit even a polynomial kernel. *Polynomial parameter reduction* (or polynomial time and parameter transformations), introduced by Bodlaender, Thomassé and Yeo [33], is a commonly used method to show the non-existence of polynomial kernels for $\mathcal{FPT}$ problems.

Definition 1.4. *A parameterized problem Q is polynomial parameter reducible to a parameterized problem Q', if there exists a polynomial-time algorithm with an instance (I, κ) of Q as input, where κ is the parameter, and with an instance (I', κ') of Q' as output such that*

(1) $(I, \kappa) \in Q$ if and only if $(I', \kappa') \in Q'$; and

(2) $\kappa' \leq Poly(\kappa)$, where $Poly(\kappa)$ is a polynomial function in κ.

To use polynomial parameter reduction to show the non-existence of polynomial kernel of an $\mathcal{FPT}$ problems, we need the following lemma.

Lemma 1.1. *([33, 83]) Let Q and Q' be two parameterized problems and $\tilde{Q}$ and $\tilde{Q}'$ be the unparameterized versions of Q and Q', respectively. Suppose that $\tilde{Q}$ is $\mathcal{NP}$-hard and $\tilde{Q}'$ is in $\mathcal{NP}$. Moreover, Q is polynomial parameter reducible to Q'. Then, if Q' has a polynomial kernel, then Q has a polynomial kernel.*

The above lemma will be used in Chapter 5 to show the non-existence of polynomial kernel of a possible winner problem on partial tournaments. We refer to [30, 32, 149, 150, 172] and Chapter 13 of [88] for representative literature on lower bounds for kernelization, where one can find many concrete problems that are showed to have no polynomial kernels using Lemma 1.1, as well as many other approaches to establish lower bounds for $\mathcal{FPT}$ problems.

1.3.3 Lenstra's ILP Theorem

Integer linear programming (ILP for short) is a very powerful technique to tackle major combinatorial optimization problems, see, e.g., [3, 169, 210, 244], due to the remarkable power of modern ILP solvers. However, the ILP problem is $\mathcal{NP}$-hard [131, 133, 162]. On the way exploring this fundamental $\mathcal{NP}$-hard problem, Lenstra [177] derived a polynomial-time algorithm for ILP instances with constant number of variables. The algorithm was later improved by Kannan [161], and then further improved by Frank and Tardos [126]. In fact, from the perspective of parameterized complexity, all their algorithms are $\mathcal{FPT}$-algorithms with respect to the number of variables. Due to this fact, all the parameterized problems which can be $\mathcal{FPT}$-reducible to ILP with respect to the number of variables are $\mathcal{FPT}$ (see e.g., [112, 119, 122] for some examples). Many $\mathcal{FPT}$-reductions from parameterized problems to ILPs had been established long before the parameterized complexity was systematically introduced. Nevertheless, the ILP technique had not been widely used for classifying parameterized problems until the work of Niedermeier [203]. We refer to [186] (Section 2.8) for an interesting discussion on the ILP problem in parameterized complexity. The following theorem is a summary of the work by Lenstra, Kannan and Frank [126, 177, 161].

Theorem 1.1. *ILP can be solved using $O(v^{2.5v+o(v)} \times L)$ arithmetic operations and space polynomial in L. Here L is the number of bits in the input and v the number of variables in ILP.*

Theorem 1.1 will be used in Chapter 3 to show the fixed-parameter tractability of several control problems in elections with bounded single-peaked width.

1.3.4 Dynamic Programming

Dynamic programming as a general algorithm design technique has been widely used to solve combinatorial optimization problems (see, e.g., [2, 25, 55, 206, 224, 256]. The basic idea of dynamic programming is iteratively break down the given instance of the problem in question into a reasonable number of subinstances, in such a way that we can use optimal solutions to smaller subinstances to give us optimal solutions to larger subinstances. In particular, the solutions to smaller subinstances are stored in a dynamic table in order to avoid repeat calculation. We refer to [90, 230] for a comprehensive and vivid introduction to dynamic programming.

In this thesis, we will use the dynamic programming technique to derive a polynomial-time algorithm for several control problems in 2-peaked elections in Chapter 2, and two exponential time algorithms for the Borda manipulation problems in Chapter 6.

1.4 Structure of this Thesis

This thesis is concerned with (parameterized) complexity of strategic voting problems in elections under natural restrictions. The remainder of this thesis is divided into 5 chapters, each is concerned with a specific topic. See Figure 1.3 for an overview.

In Chapters 2 and 3, we study (parameterized) complexity of control problems in generalized single-peaked elections. In particular, Chapter 2 is concerned with control by adding/deleting votes/candidates in $\mathscr{k}$-peaked elections for $\mathscr{r}$-Approval, Condorcet, Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. Our results concerning this topic are summarized in Tables 2.1 and 2.2. Chapter 3 is concerned with control by adding/deleting votes in elections with bounded single-peaked width for Condorcet, Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. Our main results in this chapter are summarized in Table 3.1 and Theorem 3.9. Chapters 2 and 3 are based on the papers [252, 253, 254].

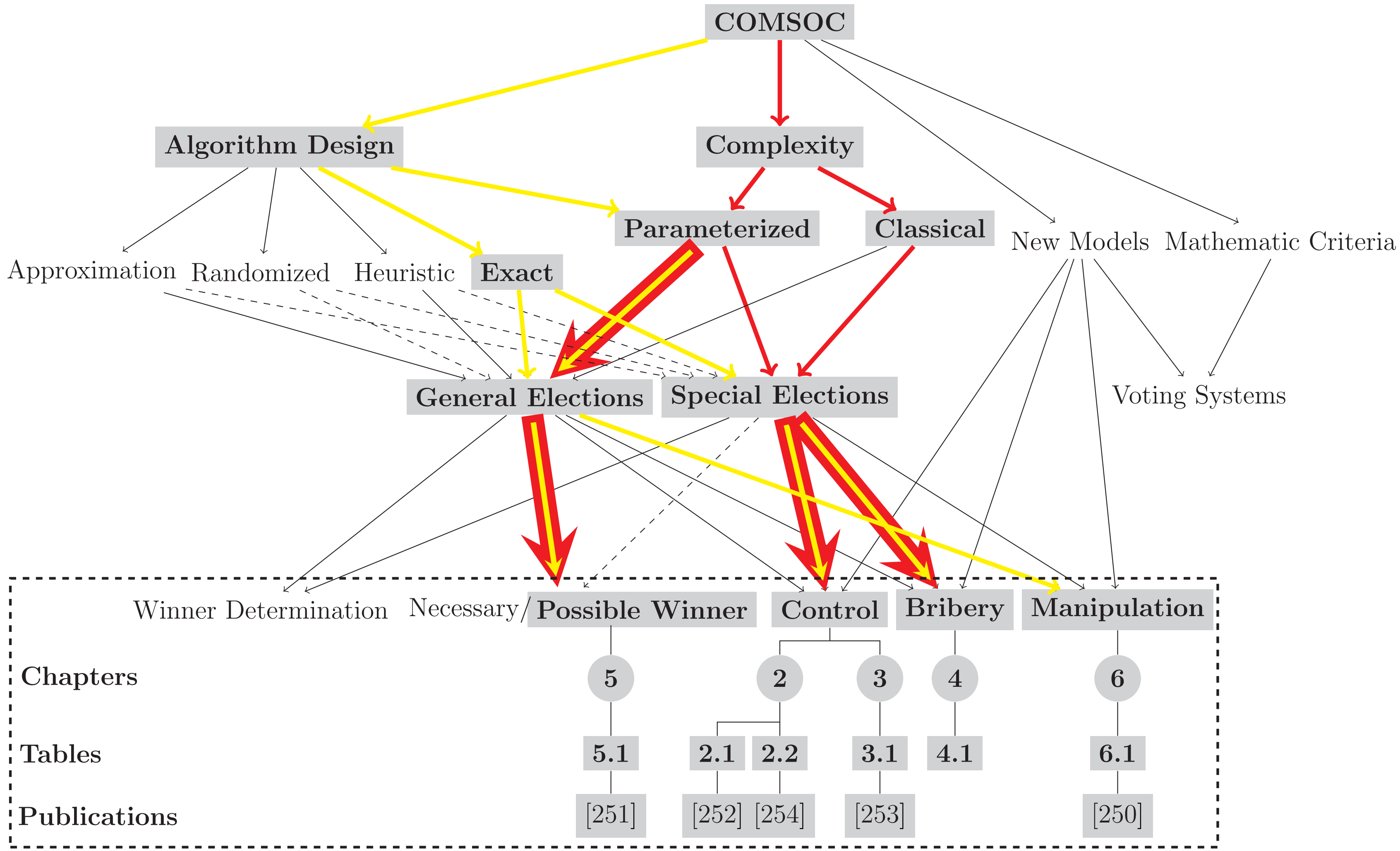


Figure 1.3: Organization of the thesis. Red arcs indicate the complexity work and green arcs indicate the algorithmic work. Arcs with both colors indicate both complexity and algorithmic work.

In Chapter 4, we study the distance restricted bribery problem which differs from the traditional bribery problem in the way that the bribed voter can recast a vote which needs to be as close as to its original vote. We adopt the Hamming distance and the Kendall-Tau distance to measure the similarity of different votes. In particular, we investigated the problem for Borda, Condorcet, Maximin and Copeland$^\alpha$ for all $0 \leq \alpha \leq 1$. Our results of this chapter are summarized in Table 4.1.

In Chapter 5, we study several possible winner(s) problems on partial tournaments related to Uncovered set and Banks set. Our results concerning this topic is summarized in Table 5.1. This chapter is based on the paper [251].

In Chapter 6, we study exact combinatorial algorithms for both weighted and unweighted Borda manipulation problems. Our main results are summarized in Table 6.1. This chapter is based on the paper [250].

In Chapter 7, we summarize our results and discuss several directions for future research.

2

Control in Multi-Peaked Elections

Imagine again the scenario discussed in the previous section where the residents living on the same street are asked to vote for the location of a supermarket among a set of potential candidates. Every voter prefers the candidate which is closest to his residence, and the farther the other candidate located away from his ideal candidate, the less it is preferred. The story in this section differs from the above one in the way that we allow voters to have more than one house on the street, or have relatives who also live on the same street. Therefore, in this story, when we visit the candidates from one side to another side, the preferences of the voters may repeatedly increase and decrease, leading to a multi-peaked voting profile.

2.1 Introduction

In this section, we mainly study control problems in multi-peaked elections. In particular, we are interested in exploring the complexity of control problems in k-peaked elections, where k is a small constant.

2.1.1 Motivation

Voting is a common method for preference aggregation and collective decision-making, and has applications in political elections, multi-agent systems, web spam reduction, pattern recognition etc [92, 93, 160, 187]. Unfortunately, by Arrow's impossibility theorem [5], there is no voting system which satisfies a certain set of desirable criteria (see [5] for the details) when more than two candidates are involved. One possible way to bypass Arrow's impossibility theorem is to restrict the domain of the preferences, for instance, the single-peaked domain introduced by Black [28]. Intuitively, in a single-peaked election, one can order the candidates from left to right such that every voter's preference increases first and then decreases after some point as the candidates are considered from left to right. See Figure 2.1 for an example.

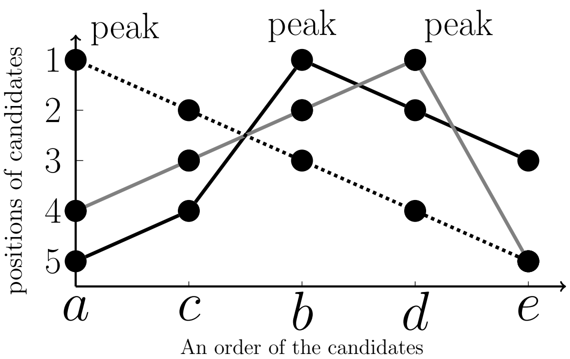


Figure 2.1: A single-peaked election with five candidates a, b, c, d, e and three votes with preferences $b \succ_u d \succ_u e \succ_u c \succ_u a$, $d \succ_v b \succ_v c \succ_v a \succ_v e$ and $a \succ_w c \succ_w b \succ_w d \succ_w e$, respectively. The preferences $\succ_u$, $\succ_v$ and $\succ_w$ are illustrated by the dark line, the gray line, and the dotted line, respectively.

Recently, the complexity of various voting problems in single-peaked elections has been attracting attention of many researchers from both theoretical computer science and social choice communities [44, 103, 113, 129, 237]. It turned out that many voting problems being $\mathcal{NP}$-hard in general become polynomial-time solvable when restricted to single-peaked elections [44, 113]. However, most elections in practice are not purely single-peaked, which motivates researchers to study more general models of elections. We refer readers to [51, 71, 78, 101, 111] for some variants or generalizations of single-peaked elections.

In this section, we consider a natural generalization of single-peaked elections, where more than one peak may occur in each vote. We call this generalization k-peaked elections (or multi-peaked elections if the number of peaks is not specified). This generalization might be relevant for many real-world applications. For example, imagine again the scenario that the residents living on the same street are asked to vote for the location of a supermarket among a set of potential candidates on the same street. Moreover, every resident may owe more than one house, or have relatives who also live on the same street. It is a natural assumption that every voter prefers the candidates which are closest to his residences or his relative's residences, and the farther the other candidates located away from his ideal candidates, the less they are preferred. In this case, when we visit the candidates from one side to another side along the street, the preferences of the voters may repeatedly increase and decrease. Nevertheless, the preference of a voter increases (or decreases) at most the number of times that is equal to the number of houses the voter and his relatives owe. k-peaked elections with k being a small constant may also arise in the scenario where the initial election is single-peaked but some voters are bribed to rank some specific candidates higher in order to get some extra benefits (e.t., money, permission, etc.) from the bribers. In addition, multi-peaked elections also play an important role in politics [69, 93]. We refer to the work of Egan [93] for a detailed discussion of how and when multi-peaked political elections arise in real-world political settings. Very recently, 2-peaked domain of preference was also studied in the context of facility location problem [123].

In this chapter, we mainly study control problems for r-Approval, Condorcet, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Maximin in k-peaked elections. We first study r-Approval in Section 2.2, and then we study the other three voting correspondences in Section 2.3. We put the last three voting correspondences into one section due to the following reasons. First, unlike r-Approval, the other three voting correspondences are pairwise comparison based voting correspondences. Moreover, they are all Condorcet-consistent. Finally, the techniques used for showing hardness of the control problems under these voting correspondences are similar.

2.1.2 Preliminaries

Apart from the definitions in the Section 1, we need the following notations and definitions to investigate the problems in this section.

Single-peaked/k-peaked elections. An election $(\mathcal{C}, \Pi_{\mathcal{V}})$ is *single-peaked* if there is a linear order $\mathcal{L}$ of $\mathcal{C}$ such that for every vote with preference $\succ_v$ in $\Pi_{\mathcal{V}}$ and every three candidates $a, b, c \in \mathcal{C}$ with $a \;\mathcal{L}\; b \;\mathcal{L}\; c$ or $c \;\mathcal{L}\; b \;\mathcal{L}\; a$, $c \succ_v b$ implies $b \succ_v a$, where $a \;\mathcal{L}\; b$ means a is ordered before b in $\mathcal{L}$. The candidate ranked in the first position of $\succ_v$ is the *peak* of $\succ_v$ with respect to $\mathcal{L}$.

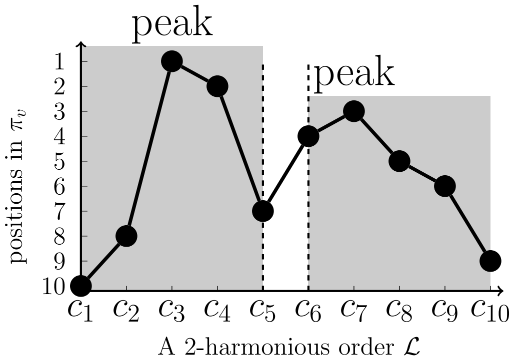


Figure 2.2: This figure shows a 2-peaked vote $\pi_v = (c_3, c_4, c_7, c_6, c_8, c_9, c_5, c_2, c_{10}, c_1)$ with respect to the 2-harmonious order $\mathcal{L} = (c_1, c_2, \ldots, c_{10})$. Here, $\mathcal{L}$ is partitioned into L_1 and L_2 with $L_1 = (c_1, c_2, c_3, c_4, c_5)$ and $L_2 = (c_6, c_7, c_8, c_9, c_{10})$. Clearly, $\pi_v(\mathcal{C}(L_1))$ and $\pi_v(\mathcal{C}(L_2))$ are single-peaked with respect to L_1 with peak c_3 and L_2 with peak c_7, respectively.

For an order $\mathcal{L} = (c_1, c_2, \ldots, c_m)$ of $\mathcal{C}$ and a vote π_v, we say π_v is $\mathcal{k}$*-peaked* with respect to $\mathcal{L}$, if there is a $\mathcal{k}'$-partition $L_1 = (c_1, c_2, ..., c_i)$, $L_2 = (c_{i+1}, c_{i+2}, ..., c_{i+j}), \ldots,$ $L_{\mathcal{k}'} = (c_k, c_{k+1}, ..., c_m)$ of $\mathcal{L}$ such that $\mathcal{k}' \leq \mathcal{k}$ and $\pi_v(\mathcal{C}(L_x))$ is single-peaked with respect to L_x for all $1 \leq x \leq \mathcal{k}'$, where $\mathcal{C}(L_x)$ is the set of candidates appearing in L_x. See Figure 2.2 for an example.

An election is $\mathcal{k}$-peaked if there is an order $\mathcal{L}$ of $\mathcal{C}$ such that every vote in the election is $\mathcal{k}$-peaked with respect to $\mathcal{L}$. Here $\mathcal{L}$ is called a $\mathcal{k}$*-harmonious order*.

Problem declaration. We study constructive/destructive control by adding/deleting votes/candidates. These problems in the general case have been defined in Section 1.2.2. In this section, we study these problems in $\mathcal{k}$-peaked elections. However, these problems in this section differ from that in the general case (see Section 1.2.2) in the way that the input elections are required to be $\mathcal{k}$-peaked elections. In particular, in the control by adding votes, the election with both the registered votes and unregistered votes has to be $\mathcal{k}$-peaked, according to a common $\mathcal{k}$-harmonious order. In the control by adding candidates, the election with candidate set $\mathcal{C} \cup \mathcal{D} \cup \{p\}$ (see Section 1.2.2) has to be $\mathcal{k}$-peaked. Furthermore, we assume that a $\mathcal{k}$-harmonious order is given alone with the given $\mathcal{k}$-peaked election. This assumption is based on the observation that in many real-world applications, the harmonious order is known in advance. This is actually one of the reasons why domain restricted elections arise in practice. For example, in real-world single-peaked political elections, the voters are thought to agree upon that the candidates are ordered on a common known left-right dimension. See [28] for related discussion.

2.2 r-Approval Control

This section considers the complexity of r-Approval control in $\mathcal{k}$-peaked elections. We begin with a short discussion of two famous voting correspondences related to r-Approval. Many of our results apply to these voting correspondences as well.

Approval voting is one of the most famous voting systems and has been extensively studied both in theory and in practice [18, 36, 199]. In an *Approval voting*, we are given a set $\mathcal{C}$ of *candidates* and a set $\mathcal{V}$ of *voters*. Each voter approves or disapproves every candidate $c \in \mathcal{C}$. The system selects a candidate who is approved by the most voters as a winner. A prominent variant of Approval voting is the *sincere-strategy preference-based Approval voting* (SP-AV for short), proposed by Brams and Sanver [39]. In an SP-AV election, each voter provides both a linear order of the candidates and a subset C of candidates such that the candidates are approved according to C, and the "admissible" and "sincere" properties should be fulfilled. In particular, the "admissible" property requires that the candidate ranked in the first position must be approved and the candidate ranked in the last position must be disapproved, and the "sincere" property requires that if a candidate c is approved then all the candidates ranked above c must be approved (see [39, 102] for more details). Many of our results apply to Approval voting and SP-AV.

In the following, we consider only constructive control. Hemaspaandra, Hemaspaandra and Rothe [146] proved that the control problems by adding/deleting votes for Approval voting are $\mathcal{NP}$-hard. The proofs can be adapted to show the $\mathcal{NP}$-hardness of control by adding/deleting votes in SP-AV [102]. Lin [180] proved that control by adding votes in 4-Approval and control by deleting votes in 3-Approval are both $\mathcal{NP}$-hard, while control by adding votes in 3-Approval and control by deleting votes in 2-Approval are polynomial-time solvable. As for the control by modification of candidates, Approval voting turned out to be immune[i] to control by adding candidates and polynomial-time solvable for control by deleting candidates [146]. However, the control problems by adding/deleting candidates are $\mathcal{NP}$-hard for r-Approval, even when degenerated to 1-Approval [157]. The $\mathcal{NP}$-hardness also holds for control by adding/deleting candidates in SP-AV [102]. Recently, control problem in Approval voting and r-Approval voting have also been considered in single-peaked elections. Faliszewski et al. [113] proved that the control problems by adding/deleting votes in Approval are polynomial-time solvable in single-peaked elections[ii]. Moreover, the control problems by adding/deleting candidates for 1-Approval are polynomial-time solvable in single-peaked elections [113].

Motivated by the $\mathcal{NP}$-hardness in the general case and the polynomial-time solvability in the single-peaked case, we study the complexity of control problems for r-Approval voting in k-peaked elections with respect to various values of k, aiming at exploring the complexity border for these control problems. Faliszewski, Hemaspaandra and Hemaspaandra [111] studied a nearly single-peaked model which is called Swoon-SP and can be considered as a special case of 2-peaked elections. They proved that the control problems by adding/deleting candidates for 1-Approval are $\mathcal{NP}$-hard when

[i]A voting system is immune to a control behavior if one cannot make a candidate who is not a winner become a final winner by imposing the strategic behavior on the election.

[ii]In [113], for the approval voting, an election is single-peaked if there is an order of the candidates such that each voter's approved candidates are contiguous within the order.

	κ-peaked elections			
	$\kappa = 1$	$\kappa = 2$	$\kappa \geq 3$	$\kappa = \lceil m/2 \rceil$
CCAV	$\mathcal{P}^{\spadesuit}$	r is not a constant: $\boldsymbol{\mathcal{NP}}$**-h**, Thm. 2.2 r is a constant: $\boldsymbol{\mathcal{P}}$, Thm. 2.1	$r \leq 3 : \mathcal{P}^{\diamondsuit}$ $r \geq 4$: $\boldsymbol{\mathcal{NP}}$**-h**, Thm. 2.4	$r \leq 3 : \mathcal{P}^{\diamondsuit}$ $r \geq 4$: $\mathcal{NP}$-h$^{\diamondsuit}$
CCDV	$\mathcal{P}^{\spadesuit}$	$r \leq 2 : \mathcal{P}^{\diamondsuit}$ $r \geq 3$: $\boldsymbol{\mathcal{NP}}$**-h**, Thm. 2.3	$r \leq 2 : \mathcal{P}^{\diamondsuit}$ $r \geq 3$: $\boldsymbol{\mathcal{NP}}$**-h**, Thm. 2.3	$r \leq 2$: $\mathcal{P}^{\diamondsuit}$ $r \geq 3$: $\mathcal{NP}$-h$^{\diamondsuit}$
CCAC	$\mathcal{P}^{\spadesuit}$	$r = 1$: $\mathcal{NP}$-h$^{\triangle}$	$r = 1$: $\mathcal{NP}$-h$^{\triangle}$	$r = 1 : \mathcal{NP}$-h$^{\clubsuit}$
CCDC	$\mathcal{P}^{\spadesuit}$	$r = 1$: $\mathcal{NP}$-h$^{\triangle}$	$r = 1$: $\boldsymbol{\mathcal{W}}$**[1]-h**, Thm. 2.5	$r = 1 : \mathcal{NP}$-h$^{\clubsuit}$

Table 2.1: A summary of the complexity of r-Approval control problems. Our new results are in bold. In this table, "$\mathcal{NP}$-h" stands for $\mathcal{NP}$-hard and "$\mathcal{P}$" stands for polynomial-time solvable. Moreover, "Thm. #" means that the result follows from Theorem # in this chapter. Note that general elections are κ-peaked elections with $\kappa = \lceil m/2 \rceil$, where m denotes the number of candidates. All results apply to the unique-winner and the nonunique-winner models. Moreover, all our $\mathcal{NP}$-hardness results apply to both Approval voting and SP-AV. However, there are no "r" in both cases. Moreover, the $\mathcal{W}[1]$-hardness result apply to SP-AV as well. Results marked by $\diamondsuit$ are from [180], by $\clubsuit$ from [157], by $\spadesuit$ from [113] and by $\triangle$ from [111].

restricted to Swoon-SP elections, implying the $\mathcal{NP}$-hardness of these problems in 2-peaked elections. We complement their results by studying the adding/deleting votes case. In particular, we show that, control by adding votes in r-Approval with r being a constant is polynomial-time solvable in 2-peaked elections, but $\mathcal{NP}$-hard in k-peaked elections for $k \geq 3$. Meanwhile, if r is not a constant, then control by adding votes in r-Approval in 2-peaked elections becomes $\mathcal{NP}$-hard. Moreover, the deleting votes case turns out to be $\mathcal{NP}$-hard for k-peaked elections with $k \geq 2$, even for r being a constant.

In addition, we present a $\mathcal{W}$-hardness result for r-Approval control in 3-peaked elections. Liu et al. [183] proved that control by adding votes in Approval voting is $\mathcal{W}[1]$-hard and control by deleting votes in Approval voting is $\mathcal{W}[2]$-hard, with the numbers of added and deleted votes as parameters, respectively. In addition, they proved that control by adding candidates in 1-Approval is $\mathcal{W}[2]$-hard, with the number of added candidates as the parameter. Betzler and Uhlmann [27] complemented the results in [183] by proving that control by deleting candidates in 1-Approval is $\mathcal{W}[2]$-hard, with the number of deleted candidates as the parameter. We extend the above results to k-peaked elections by showing that control by deleting candidates in 1-Approval in 3-peaked elections is $\mathcal{W}[1]$-hard with the number of deleted candidates as the parameter. All our findings in this Section are summarized in Table 2.1.

2.2.1 2-Peaked Elections

In this section, we study control problems for r-Approval in 2-peaked elections. We begin with some polynomial-time solvability results.

Theorem 2.1. *Both CCAV-r-Approval-UNI and CCAV-r-Approval-NON in 2-peaked elections are polynomial-time solvable for every constant r.*

Proof. We prove Theorem 2.1 by giving a polynomial-time algorithm based on dynamic programming. We first consider CCAV-r-Approval-UNI.

Let $((\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}), \Pi_{\mathcal{T}}, \mathcal{L}, \mathcal{R})$ be an instance of CCAV-r-Approval-UNI in 2-peaked elections. For a candidate $c \in \mathcal{C}$, let $\overleftarrow{c}(1)$ be the candidate lying immediately before c in $\mathcal{L}$ and $\overleftarrow{c}(i)$ be the candidate lying immediately before $\overleftarrow{c}(i-1)$ in $\mathcal{L}$. Similarly, we use $\overrightarrow{c}(1)$ and $\overrightarrow{c}(i)$ to denote the candidates lying immediately after c and $\overrightarrow{c}(i-1)$, respectively. For example, if $\mathcal{L} = (a, b, c, d, e, f, g, h)$, then $\overrightarrow{d}(1) = e$, $\overrightarrow{d}(4) = h$, $\overleftarrow{d}(1) = c$ and $\overleftarrow{d}(3) = a$.

For a vote π_v, let $1(v)$ denote the set of candidates who get 1 point and $0(v)$ denote the set of candidates who get 0 points, from π_v. For a candidate c, let $SC_{\Pi_{\mathcal{V}}}(c)$

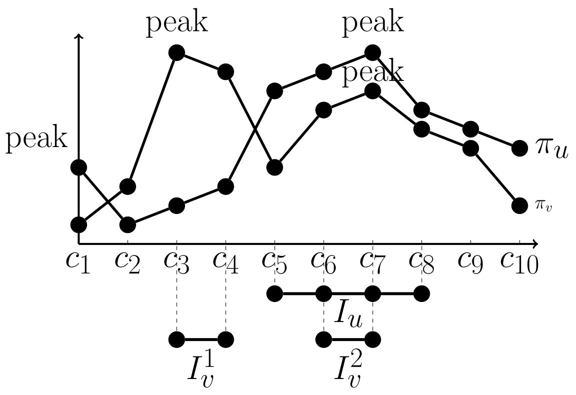


Figure 2.3: This figure shows two votes $\pi_v = (c_3, c_4, c_7, c_6, c_8, c_9, c_5, c_2, c_{10}, c_1)$ and $\pi_u = (c_7, c_6, c_5, c_8, c_9, c_{10}, c_1, c_4, c_3, c_2)$. Each vote gives one point to its top four ranked candidates. $1(v)$ is represented by a 2-discrete interval $\{I_v^1 = (c_3, c_4), I_v^2 = (c_6, c_7)\}$ and $1(u)$ is represented by a 1-discrete interval $\{I_u = (c_5, c_6, c_7, c_8)\}$.

(or simply $SC_{\mathcal{V}}(c)$ if it is clear from the context) be the total score of c from $\Pi_{\mathcal{V}}$, that is, $SC_{\Pi_{\mathcal{V}}}(c) = |\{\pi_v \in_+ \Pi_{\mathcal{V}} \mid c \in 1(v)\}|$.

Given an order $A = (a_1, a_2, \ldots, a_n)$, a *discrete interval* I over A is a consecutive sub-order $(a_i, a_{i+1}, \ldots, a_{i+j})$ of A. We denote the first element a_i (a_i is also referred to as the left endpoint of I) by $l(I)$ and the last element a_{i+j} (a_{i+j} is also referred to as the right endpoint of I) by $r(I)$. We also use $A(l(I), r(I))$ to denote I. Let $\mathcal{S}(I)$ denote the set of elements appearing in I and set $|I| = |\mathcal{S}(I)|$. For example, for a discrete interval $I = A(3, 6)$ over the order $A = (2, 5, 3, 10, 4, 6, 0)$, $\mathcal{S}(I)$ is $\{3, 4, 6, 10\}$. A *$\mathfrak{b}$-discrete interval* over an order A is a collection of $\mathfrak{b}$ disjoint discrete intervals over A, where “disjoint” means no element in A appears in more than one discrete interval. For a $\mathfrak{b}$-discrete interval $\mathcal{I}$, let $\mathcal{S}(\mathcal{I}) = \bigcup_{I \in \mathcal{I}} \mathcal{S}(I)$.

We have the following observation (a similar observation on Approval voting in single-peaked elections is given in [113]).

Observation 2.1. *For each $\mathfrak{k}$-peaked election $(\mathcal{C}, \Pi_{\mathcal{V}})$ associated with a $\mathfrak{k}$-harmonious order $\mathcal{L}$ over $\mathcal{C}$, and each vote $\pi_v \in_+ \Pi_{\mathcal{V}}$, there is a $\mathfrak{b}$-discrete interval $\mathcal{I}$ over $\mathcal{L}$ such that $0 < \mathfrak{b} \leq \mathfrak{k}$ and $1(v) = \mathcal{S}(\mathcal{I})$.*

By Observation 2.1, for every vote π_v in a 2-peaked election associated with $\mathcal{L}$ as a 2-harmonious order, $1(v)$ can be represented by a 2-discrete interval or a 1-discrete interval over $\mathcal{L}$. See Figure 2.3 for an example.

We first derive a polynomial-time algorithm for CCAV-4-Approval-UNI in 2-peaked elections. It is easy to generalize the algorithm to CCAV-$\mathfrak{r}$-Approval-UNI in $\mathfrak{r}$-peaked elections with $\mathfrak{r}$ being a constant. The following observation is trivial.

Observation 2.2. *Every yes-instance of CCAV-$\mathfrak{r}$-Approval has a solution where each added vote approves the distinguished candidate $\mathfrak{p}$.*

Due to Observation 2.2, we can safely assume that for each $\pi_v \in_+ \Pi_{\mathcal{T}}$, $\mathfrak{p} \in 1(v)$. By Observation 2.1, for every vote $\pi_v \in_+ \Pi_{\mathcal{T}}$, $1(v)$ can be represented by a 2-discrete interval $\mathcal{I}_v = \{I_v^{\bar{\mathfrak{p}}}, I_v^{\mathfrak{p}}\}$ or a 1-discrete interval $\mathcal{I}_v = \{I_v^{\mathfrak{p}}\}$ with $\mathfrak{p} \in \mathcal{S}(I_v^{\mathfrak{p}})$. Let Π be the multiset of all votes $\pi_v \in_+ \Pi_{\mathcal{T}}$ where $1(v)$ is represented by a 1-discrete interval over $\mathcal{L}$. We say two votes have the same *type* if they approve exactly the same candidates (For instance, the two votes with preferences $a \succ b \succ c \succ d \succ e \succ f$ and $c \succ d \succ b \succ a \succ e \succ f$ in 4-Approval have the same type, since both votes approve exactly the same candidates a, b, c, d). Since every voter approves exactly four candidates, votes in Π have at most four different types:

(1) votes approving $\overleftarrow{p}(3), \overleftarrow{p}(2), \overleftarrow{p}(1), p$;

(2) votes approving $\overleftarrow{p}(2), \overleftarrow{p}(1), p, \overrightarrow{p}(1)$;

(3) votes approving $\overleftarrow{p}(1), p, \overrightarrow{p}(1), \overrightarrow{p}(2)$; and

(4) votes approving $p, \overrightarrow{p}(1), \overrightarrow{p}(2), \overrightarrow{p}(3)$.

We then can enumerate all possibilities of how many votes in the solution are from each of the four types of votes in Π. This reduces the original instance to at most $\mathcal{R}^4$ subinstances. Thus, in the following, we assume that every vote in $\Pi_{\mathcal{T}}$ is represented by a 2-discrete interval. Let $\overrightarrow{\Pi}_{\mathcal{T}} = (\pi_{v_1}, \pi_{v_2}, ..., \pi_{v_{|\mathcal{T}|}})$ be an order of $\Pi_{\mathcal{T}}$ such that $r(I^{\bar{p}}_{v_i}) = r(I^{\bar{p}}_{v_j})$ or $r(I^{\bar{p}}_{v_i})\ \mathcal{L}\ r(I^{\bar{p}}_{v_j})$ for all $1 \leq i < j \leq |\Pi_{\mathcal{T}}|$.

Our dynamic programming algorithm uses a binary dynamic table

$$DT(i, j, s, s_1, s_2, s_3, s_4, s_5, s_6, s_{i,1}, s_{i,2}, s_{i,3}),$$

where we set $DT(i, j, s, s_1, s_2, s_3, s_4, s_5, s_6, s_{i,1}, s_{i,2}, s_{i,3}) = 1$ if there is a submultiset $\Pi_{\mathcal{T}'} \sqsubseteq \{\pi_{v_1}, \pi_{v_2}, \ldots, \pi_{v_i}\}$ satisfying

(1) $|\Pi_{\mathcal{T}'}| = j$;

(2) $\pi_{v_i} \in_+ \Pi_{\mathcal{T}'}$;

(3) $\max\{SC_{\mathcal{V}\cup\mathcal{T}'}(c) \mid c \in \mathcal{C}\} = s$;

(4) $SC_{\mathcal{V}\cup\mathcal{T}'}(c_t) = s_t$ for all $1 \leq t \leq 6$, where $c_3 = \overleftarrow{p}(1), c_2 = \overleftarrow{p}(2), c_1 = \overleftarrow{p}(3), c_4 = \overrightarrow{p}(1), c_5 = \overrightarrow{p}(2)$ and $c_6 = \overrightarrow{p}(3)$; and

(5) $SC_{\mathcal{V}\cup\mathcal{T}'}(c_{i,t}) = s_{i,t}$ for all $t \in \{1, 2, 3\}$, where $c_{i,1} = r(I^{\bar{p}}_{v_i}), c_{i,2} = \overleftarrow{c_{i,1}}(1)$ and $c_{i,3} = \overleftarrow{c_{i,1}}(2)$. (See Figure 2.4 for an illustration of (4) and (5)).

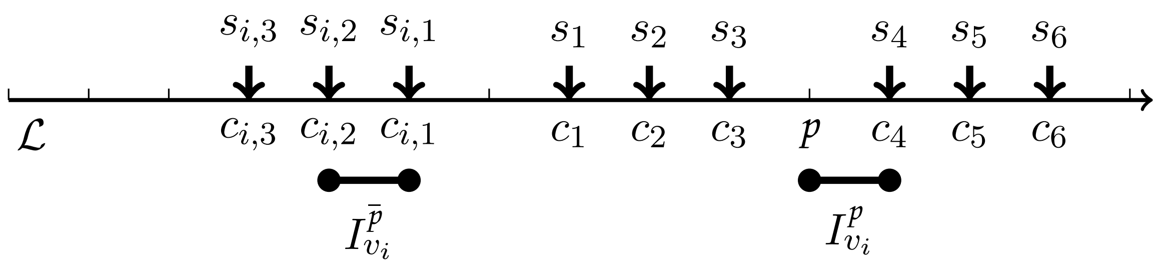


Figure 2.4: Illustration of (4) and (5) in the definition of the dynamic table DT.

It is easy to see that the given instance is a yes-instance if there is a $DT(n, \mathcal{R}', s, s_1, s_2, \ldots, s_6, s_{n,1}, s_{n,2}, s_{n,3}) = 1$ for some $n \leq |\Pi_{\mathcal{T}}|$, $\mathcal{R}' \leq \mathcal{R}$, $s \leq SC_{\mathcal{V}}(p) + \mathcal{R}' - 1$ and $s' \leq s$ for all $s' \in \{s_1, s_2, ..., s_6, s_{n,1}, s_{n,2}, s_{n,3}\}$. Therefore, to solve the problem we need to calculate the values of $DT(i, j, s, s_1, s_2, \ldots, s_6, s_{i,1}, s_{i,2}, s_{i,3})$ for all $1 \leq j \leq \mathcal{R}$, $j \leq i \leq |\Pi_{\mathcal{T}}|$, $1 \leq s \leq SC_{\mathcal{V}}(p) + \mathcal{R} - 1$ and $s' \leq s$ for all $s' \in \{s_1, s_2, ..., s_6, s_{i,1}, s_{i,2}, s_{i,3}\}$. Thus, we have at most $|\Pi_{\mathcal{T}}| \cdot \mathcal{R} \cdot (|\Pi_{\mathcal{V}}| + \mathcal{R})^{10}$ entries to calculate.

We use the following iterative recurrence to update the table. $DT(i, j, s, s_1, s_2, \ldots, s_6, s_{i,1}, s_{i,2}, s_{i,3}) = 1$, if at least one of the following cases applies:

Case 1. $\exists DT(i_1, j-1, s, s'_1, s'_2, \ldots, s'_6, s'_{i_1,1}, s'_{i_1,2}, s'_{i_1,3}) = 1$ such that conditions (1)-(4) hold.

Case 2. $\exists s' \in \{s_1, s_2, ..., s_6, s_{i,1}, s_{i,2}, s_{i,3}\}$ with $s' = s$ and

$$\exists DT(i_1, j-1, s-1, s'_1, s'_2, \ldots, s'_6, s'_{i_1,1}, s'_{i_1,2}, s'_{i_1,3}) = 1$$

such that conditions (1)-(4) hold.

The four conditions are:

(1) $j - 1 \leq i_1 \leq i - 1$;

(2) $s_t = s'_t + SC_{\{v_i\}}(c_t)$ for all $1 \leq t \leq 6$;

(3) $s_{i,t} = s'_{i_1,t_1} + SC_{\{v_i\}}(c_{i,t})$ for all $c_{i,t} = c_{i_1,t_1}$; and

(4) $s_{i,t} = SC_{\mathcal{V} \cup \{v_i\}}(c_{i,t})$ for all $c_{i,t} \in A$, where

$$A = \{r(I^{\bar{p}}_{v_i}), \overleftarrow{r(I^{\bar{p}}_{v_i})}(1), \overleftarrow{r(I^{\bar{p}}_{v_i})}(2)\} \setminus \{r(I^{\bar{p}}_{v_{i_1}}), \overleftarrow{r(I^{\bar{p}}_{v_{i_1}})}(1), \overleftarrow{r(I^{\bar{p}}_{v_{i_1}})}(2)\}.$$

The above algorithm can be adapted to solve the nonunique-winner model CCAV-r-Approval-NON: replacing all appearances of "$SC_{\mathcal{V}}(p)+\mathcal{R}-1$" in the above description with "$SC_{\mathcal{V}}(p) + \mathcal{R}$".

The algorithm can be easily generalized to every $r \geq 4$ by using a bigger but still polynomial-sized dynamic table. In particular, for each fixed r, we need a $3r$-dimension dynamic table $DT(i, j, s, s_1, ..., s_{2(r-1)}, s_{i,1}, ..., s_{i,r-1})$, where i, j, s take the same meanings as in the above algorithm, $s_1, ..., s_{2(r-1)}$ maintain the scores of the $2(r-1)$ candidates around the distinguished candidate p (precisely, we maintain the scores of the $r-1$ candidates immediately lying on the left side of p, and the scores of the $r-1$ candidates immediately lying on the right side of p in the 2-harmonious order. If there are less than $r-1$ candidates lying on one side of p, we reduce the dimension of the dynamic table accordingly), and $s_{i,1}, ..., s_{i,r-1}$ maintain the scores of the candidate $r(I^{\overline{p}}_{v_i})$ and the $r-2$ candidates immediately lying on the left side of $r(I^{\overline{p}}_{v_i})$ in the 2-harmonious order. □

Recall that both CCAV-r-Approval-UNI and CCAV-r-Approval-NON are $\mathcal{NP}$-hard in general for every constant $r \geq 4$ but polynomial-time solvable when restricted to single-peaked elections [113]. Theorem 2.1 shows that the polynomial-time solvability of CCAV-r-Approval-UNI/NON remains when extending from single-peaked elections to 2-peaked elections, for r being a constant. This bound is tight as indicated by the following theorem. More precisely, if r is not a constant, CCAV-r-Approval-UNI/NON becomes $\mathcal{NP}$-hard in 2-peaked elections, in contrast to the polynomial-time solvability in the single-peaked case [113].

Theorem 2.2. *Both CCAV-r-Approval-UNI and CCAV-r-Approval-NON are $\mathcal{NP}$-hard in 2-peaked elections if r is not a constant.*

Proof. We prove Theorem 2.2 by a reduction from a variant of Independent Set which is $\mathcal{NP}$-hard [164].

Let ( ) denote an empty order containing no element. For a linear order $A = (a_1, a_2, \ldots, a_n)$, over the set $\{a_1, a_2, ..., a_n\}$, let $A[a_i, a_j]$ (resp. $A(a_i, a_j]$, $A[a_i, a_j)$ and $A(a_i, a_j)$) with $i \leq j$ be the sub-order $(a_i, a_{i+1}, \ldots, a_j)$ (resp. $(a_{i+1}, a_{i+2}, \ldots, a_j)$ if $i < j$ and ( ) if $i = j$, $(a_i, a_{i+1}, \ldots, a_{j-1})$ if $i < j$ and ( ) if $i = j$, and $(a_{i+1}, a_{i+2}, \ldots, a_{j-1})$ if $i < j-1$ and ( ) if $j \geq i \geq j-1$), and let $A[a_j, a_i]$ (resp. $A[a_j, a_i)$, $A(a_j, a_i]$ and $A(a_j, a_i)$) be the reversed order of $A[a_i, a_j]$ (resp. $A(a_i, a_j]$, $A[a_i, a_j)$ and $A(a_i, a_j)$). For two linear orders $A = (a_1, a_2, ..., a_n)$ and $B = (b_1, b_2, ..., b_m)$ with $A \cap B = \emptyset$, denote by (A, B) the linear order $(a_1, a_2, ..., a_n, b_1, b_2, ..., b_m)$. Let $[n]$ be the set $\{1, 2, ..., n\}$.

> A Variant of Independent Set (Vis)
> *Input:* A multiset $\mathcal{T} = \{T_1, T_2, ..., T_n\}$ where each $T_i \in_+ \mathcal{T}$ is a set of discrete intervals over the linear order $(1, 2, ..., 12n)$. Moreover, $|T_i| \leq 3$ and each discrete interval in T_i is of size 4.
> *Question:* Is there a set $S \subseteq \bigcup_{T \in_+ \mathcal{T}} T$ of discrete intervals such that $|S| = n$, $|S \cap T_i| = 1$ for every $T_i \in_+ \mathcal{T}$ and no two discrete intervals in S intersect?

We first prove the $\mathcal{NP}$-hardness of CCAV-r-Approval-UNI. Given an instance $\mathcal{F} = (\mathcal{T} = \{T_1, T_2, \ldots, T_n\})$ of Vis, we construct an instance $\mathcal{E} = ((\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}), \Pi_{\mathcal{T}}, \mathcal{L}, \mathcal{R} = n)$ for CCAV-r-Approval-UNI in 2-peaked elections with r being not a constant as follows.

Let $\mathcal{I} = \bigcup_{T \in_+ \mathcal{T}} T$. For each discrete interval $I \in \mathcal{I}$, let $l(I)$ be its left endpoint and $r(I)$ be its right endpoint. Let Γ be the set of all elements appearing in some discrete interval of $\mathcal{I}$, i.e., $\Gamma = \{\mathcal{S}(I) \mid I \in \mathcal{I}\}$. Let $\vec{\Gamma} = (x_1, x_2, \ldots, x_{|\Gamma|})$ be an order of Γ where $x_i < x_{i+1}$ for all $i \in [|\Gamma| - 1]$.

Candidates: We create three kinds of candidates C, D and E:

(1) $C = \Gamma$, that is, for each $x_i \in \Gamma$, we create a candidate;

(2) D contains exactly $2n - 1$ candidates $d_1, d_2, \ldots, d_n, \ldots, d_{2n-1}$;

(3) E contains exactly $(n + 3) \cdot (|C| + |D| - 1)$ dummy candidates $x'_1, x'_2, \ldots, x'_{|C|\cdot(n+3)}, d'_1, d'_2, \ldots, d'_{(n+3)\cdot(|D|-1)}$ which will never be winners. The distinguished candidate is d_n, that is, $p = d_n$. Moreover, $r = n + 4$.

2-Harmonious Order: $\mathcal{L} = (\vec{\Gamma}, \vec{D}, \vec{E})$ where $\vec{D} = (d_1, d_2, \ldots, d_{2n-1})$ and $\vec{E} = (x'_1, \ldots, x'_{|C|\cdot(n+3)}, d'_1, \ldots, d'_{(n+3)\cdot(|D|-1)})$

Registered Votes $\Pi_\mathcal{V}$: We create the following registered votes:

(1) for each $x_i \in C$, create $n - 2$ votes defined as

$(x_i, \mathcal{L}[x'_{(n+3)i-n-2}, x'_{i(n+3)}], \mathcal{L}(x_i, x_1], \mathcal{L}(x_i, x'_{(n+3)i-n-2}), \mathcal{L}(x'_{i(n+3)}, d'_{(|D|-1)\cdot(n+3)}]);$

(2) for each $d_i \in D$ where $i \in [n - 1]$, create $n - (i + 1)$ votes defined as

$(d_i, \mathcal{L}[d'_{(n+3)i-n-2}, d'_{i(n+3)}], \mathcal{L}(d_i, x_1], \mathcal{L}(d_i, d'_{(n+3)i-n-2}), \mathcal{L}(d'_{i(n+3)}, d'_{(|D|-1)\cdot(n+3)}]);$

(3) for each $d_i \in D$ where $i \in \{n+1, n+2, \ldots, 2n-1\}$, create $i - (n+1)$ votes which is defined as

$(d_i, \mathcal{L}[d'_{(n+3)i-2n-5}, d'_{(i-1)\cdot(n+3)}], \mathcal{L}(d_i, x_1], \mathcal{L}(d_i, d'_{(n+3)i-2n-5}), \mathcal{L}(d'_{(i-1)\cdot(n+3)}, d'_{(|D|-1)\cdot(n+3)}]).$

Unregistered Votes $\Pi_\mathcal{T}$: For each $I_{ij} \in T_i \in_+ \mathcal{T}$, create a corresponding unregistered vote which is defined as $(\mathcal{L}[l(I_{ij}), r(I_{ij})], \mathcal{L}[d_i, d'_{(|D|-1)\cdot(n+3)}], \mathcal{L}(l(I_{ij}), x_1], \mathcal{L}(r(I_{ij}), d_{i-1}])$. Clearly, this vote approves exactly all four candidates lying between $l(I_{ij})$ and $r(I_{ij})$ (including $l(I_{ij})$ and $r(I_{ij})$) in $\mathcal{L}$ and all candidates lying between d_i and d_{i+n-1} (including d_i and d_{i+n-1}) in $\mathcal{L}$. Thus, every unregistered vote approves d_n.

It is clear that all votes are 2-peaked with respect to $\mathcal{L}$. Due to the construction, it is easy to see that $SC_\mathcal{V}(c) = n - 2$ for all $c \in C$, $SC_\mathcal{V}(d_i) = n - i - 1$ for all $d_i \in D$ with $i \in [n-1]$, $SC_\mathcal{V}(d_i) = i - n - 1$ for all $d_i \in D$ with $i \in \{n+1, n+2, \ldots, 2n-1\}$, and $SC_\mathcal{V}(c) \leq n - 2$ for all $c \in E$ and $SC_\mathcal{V}(d_n) = 0$.

$\Rightarrow$: Suppose that $\mathcal{F}$ is a yes-instance and let S be a solution for $\mathcal{F}$. Let $\vec{S} = (I_1, I_2, \ldots, I_n)$ be an order of S where $I_i = S \cap T_i$ for all $i \in [n]$. Then, we can make d_n the unique winner by adding votes from $\Pi_\mathcal{T}$ according to S. More specifically, for each $I_i \in S$ we select its corresponding vote constructed as above and add it to the registered votes. Clearly, the final score of d_n is n. Due to the construction, no two added votes which correspond to two different intervals I_i and I_j, respectively, approve a common candidate from C. Thus, after adding these votes to the registered votes, no candidate in C has a higher score than that of d_n. To analyze the score of $d_j \in D$ with $j \in [n-1]$, we observe that for any $i > j$ the vote corresponding to I_i does not approve d_j. Since $SC_{\mathcal{V}}(d_j) = n - j - 1$ and $|S \cap T_i| = 1$ for all $i \in [j]$, we know that the final score of d_j is less than n. Similarly, to analyze the score of $d_j \in D$ with $j \in \{n+1, n+2, \ldots, 2n-1\}$, we observe that for any $i \leq j-n$ the vote corresponding to I_i does not approve d_j. Since $SC_{\mathcal{V}}(d_j) = j-n-1$ and $|S \cap T_i| = 1$ for all $i \in \{j-n+1, j-n+2, ..., n\}$, we know that the final score of d_j is less than n. The final score of each $c \in E$ is clearly less than $n-2$ since no unregistered vote approves c. Summarize the above analysis, we conclude that the distinguished candidate d_n becomes the unique winner after adding the selected votes to the registered votes.

$\Leftarrow$: Suppose that $\mathcal{E}$ is a yes-instance and Π_S is a multiset of votes chosen from $\Pi_{\mathcal{T}}$ which makes d_n the unique winner in the election $(C \cup D \cup E, \Pi_{\mathcal{V}} \uplus \Pi_S)$. It is easy to verify that $|\Pi_S| = n$, since otherwise, at least one of C would be a winner; thus, the final score of d_n is n and every $c \in C$ can get at most one point from Π_S. Therefore, no two votes in Π_S approve a common candidate of C, implying that Π_S must be a set. Let $P_1, P_2, \ldots, P_n$ be a partition of $\Pi_{\mathcal{T}}$ where P_i contains all votes corresponding to the intervals of $T_i \in_+ \mathcal{T}$. Clearly, P_i is a set. We claim here that $|\Pi_S \cap P_i| = 1$ for every $i \in [n]$. Suppose this is not true, then there must be a certain P_i with $|\Pi_S \cap P_i| \geq 2$. Let $S_1 = \Pi_S \cap P_i$ (thus, $|S_1| \geq 2$), $S_2 = \{\pi_v \in \Pi_S \cap P_{i'} \mid i' < i\}$ and $S_3 = \{\pi_v \in \Pi_S \cap P_{i'} \mid i' > i\}$. It is clear that $|S_1| + |S_2| + |S_3| = n$. Since all votes in S_1 approve both d_i and d_{i+n-1}, all votes in S_2 approve d_i but do not approve d_{i+n-1}, and all votes in S_3 approve d_{i+n-1} but do not approve d_i, then,

$$
\begin{aligned}
& SC_{\mathcal{V} \uplus S}(d_i) + SC_{\mathcal{V} \uplus S}(d_{i+n-1}) \\
&= SC_{\mathcal{V}}(d_i) + |S_1| + |S_2| + SC_{\mathcal{V}}(d_{i+n-1}) + |S_1| + |S_3| \\
&= n - i - 1 + |S_1| + |S_2| + i - 2 + |S_1| + |S_3| \\
&= 2n - 3 + |S_1| \\
&\geq 2n - 1
\end{aligned}
$$

Thus, at least one of d_i and d_{i+n-1} has final score at least n, contradicting that d_n is the unique winner. The claim is true. It is now easy to see that the set of discrete intervals corresponding to the votes in Π_S forms a solution for $\mathcal{F}$.

The $\mathcal{NP}$-hardness reduction for CCAV-r-Approval-NON is the same as for CCAV-r-Approval-UNI with the difference in the construction of the registered votes. In particular, we need to construct the registered votes so that the score of each candidate in $C \cup D$ is exactly one point greater than that in the above construction. This can be done easily:

(1) for each $x_i \in C$, create $n-1$ votes defined as

$(x_i, \mathcal{L}[x'_{(n+3)i-n-2}, x'_{i(n+3)}], \mathcal{L}(x_i, x_1], \mathcal{L}(x_i, x'_{(n+3)i-n-2}), \mathcal{L}(x'_{i(n+3)}, d'_{(|D|-1)\cdot(n+3)}]);$

(2) for each $d_i \in D$ where $i \in [n-1]$, create $n-i$ votes defined as

$(d_i, \mathcal{L}[d'_{(n+3)i-n-2}, d'_{i(n+3)}], \mathcal{L}(d_i, x_1], \mathcal{L}(d_i, d'_{(n+3)i-n-2}), \mathcal{L}(d'_{i(n+3)}, d'_{(|D|-1)\cdot(n+3)}]);$

(3) for each $d_i \in D$ where $i \in \{n+1, n+2, \ldots, 2n-1\}$, create $i-n$ votes which is defined as

$$(d_i, \mathcal{L}[d'_{(n+3)i-2n-5}, d'_{(i-1)\cdot(n+3)}], \mathcal{L}(d_i, x_1], \mathcal{L}(d_i, d'_{(n+3)i-2n-5}), \mathcal{L}(d'_{(i-1)\cdot(n+3)}, d'_{(|D|-1)\cdot(n+3)}]).$$

□

The control problem by deleting votes for r-Approval is polynomial-time solvable in single-peaked elections for even non-constant r [113]. The following theorem shows that by increasing the number of peaks only by one, this problem becomes $\mathcal{NP}$-hard.

Theorem 2.3. *Both CCDV-r-Approval-UNI and CCDV-r-Approval-NON in 2-peaked elections are $\mathcal{NP}$-hard for every constant $r \geq 3$.*

Proof. We first prove that CCDV-3-Approval-UNI in 2-peaked elections is $\mathcal{NP}$-hard by a reduction from Vertex Cover on bounded degree-3 graphs which is $\mathcal{NP}$-hard [130]. Then, we will show that the proof applies to CCDV-r-Approval-UNI for $r \geq 4$ with a slight modification.

An *undirected graph* is a tuple $G = (V, E)$ where V is the set of vertices and E is the set of edges. We also use $V(G)$ to denote the vertex set of G. For a vertex $u \in V$, $N_G(u)$ denotes the set of its *neighbors* in G, that is, $N_G(u) = \{w \mid (w, u) \in E\}$. The *degree* of a vertex u is the number of its neighbors. A graph is a *bounded degree-3 graph* if it contains at least one degree-3 vertex but no vertex having degree greater than 3. A *vertex cover* for a graph $G = (V, E)$ is a subset $S \subseteq V$ such that every edge in E has at least one of its endpoints in S.

Vertex Cover on Bounded Degree-3 Graphs (VC3)
Input: A bounded degree-3 graph $G = (V, E)$ and a positive integer κ.
Question: Does G have a vertex cover of size at most κ?

To prove the $\mathcal{NP}$-hardness of CCDV-r-Approval-UNI in 2-peaked elections, we first introduce a property for bounded degree-3 graphs. This property may be of independent interest since many graph problems are $\mathcal{NP}$-hard when restricted to graphs with bounded degree 3.

An *interval* over the real line is a closed set $[a, b] = \{x \in \mathbb{R} \mid a \leq x \leq b\}$ where a and b are real numbers. An interval is *trivial* if $a = b$; otherwise, it is called a *non-trivial interval.* For an interval $I = [a, b]$, denote by $l(I)$ and $r(I)$ its left-point a and right-point b, respectively. The *endpoints* of an interval are refereed to its left-point and right-point. A *b-interval* is a set of b intervals over the real line. The *endpoints* of a $\mathscr{b}$-interval are the union of the endpoints of the intervals included in it. A graph $G = (V, E)$ is a *$\mathscr{b}$-interval graph* if there is a set $\mathcal{T}_G$ of $\mathscr{b}$-intervals and a bijection $f : V \to \mathcal{T}_G$ such that for every $u, w \in V$, $(u, w) \in E$ if and only if $f(u)$ and $f(w)$ intersect. Here, $\mathcal{T}_G$ is called a *$\mathscr{b}$-interval representation* of G. For simplicity, we use $\mathcal{I}_u = \{I_u^1, I_u^2, ..., I_u^{\mathscr{b}}\}$ to denote $f(u)$, where each I_u^i is an interval. For two real numbers a and b with $a \leq b$, we define $(a, b) = \{x \in \mathbb{R} \mid a < x < b\}$.

The following lemma states that every bounded degree-3 graph has a 2-interval representation such that (1) every vertex is represented by a 2-interval with one interval is trivial; and (2) every two 2-intervals can only intersect at the endpoints.

Lemma 2.1. *For every bounded degree-3 graph G there is a 2-interval representation for G such that for every $u \in V(G)$, $\mathcal{I}_u = \{I_u^1, I_u^2\}$ satisfies one of the following:*

1. *$I_u^1 = [x_1, x_1]$, $I_u^2 = [x_2, x_3]$, $x_1 < x_2 < x_3$ and $\nexists u' \in V(G) \setminus \{u\}$ such that $r(I(u')) \in (x_2, x_3)$ or $l(I(u')) \in (x_2, x_3)$;*
2. *$I_u^1 = [x_1, x_2]$, $I_u^2 = [x_3, x_3]$, $x_1 < x_2 < x_3$ and $\nexists u' \in V(G) \setminus \{u\}$ such that $r(I(u')) \in (x_1, x_2)$ or $l(I(u')) \in (x_1, x_2)$,*

for each $I(u') \in \{I_{u'}^1, I_{u'}^2\}$. Moreover, such a 2-interval representation can be found in polynomial time. See Figure 2.5 for an example.

We defer the proof of Lemma to [252].

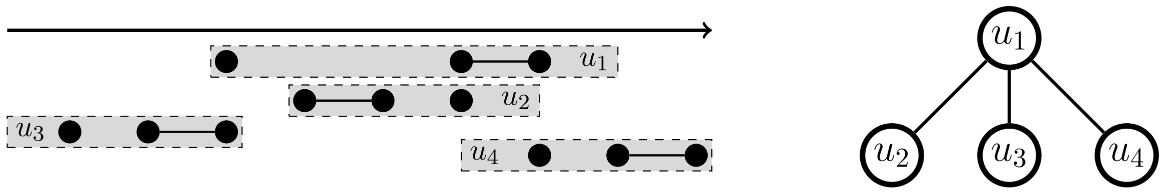


Figure 2.5: The figure on the left-side illustrates a 2-interval representation of the graph on the right-side. Here, the 2-intervals from up to down represent the vertices u_1, u_2, u_3 and u_4, respectively.

We now show the reduction. Let $\mathcal{F} = (G, \kappa)$ be an instance of VC3 and $\mathcal{I}(G)$ be a 2-interval representation of G satisfying all conditions in Lemma 2.1. For every $\mathcal{I}_u = \{I_u^1, I_u^2\}$, let $D(u)$ be the endpoints of I_u^1 and I_u^2 (due to Lemma 2.1, $|D(u)| = 3$ for all $u \in V(G)$), and let $\Gamma = \bigcup_{u \in V(G)} D(u)$. Let $\vec{\Gamma} = (x_1, x_2, \ldots, x_{|\Gamma|})$ be the order of Γ with $x_i < x_{i+1}$ for all $i \in [|\Gamma| - 1]$. We construct an instance $\mathcal{E} = ((\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}), \mathcal{R} = \kappa, \mathcal{L})$ of CCDV-3-Approval-UNI in 2-peaked elections as follows.

Candidates: $\mathcal{C} = \Gamma \cup \{p, c_1, c_2, c_3, c_4\}$ with c_1, c_2, c_3, c_4 being dummy candidates which would never be winners.

2-Harmonious Order: $\mathcal{L} = (\vec{\Gamma}, p, c_1, c_2, c_3, c_4)$.

Votes: There are two types of votes: votes disapproving p and votes approving p. There are $|V(G)|$ votes of the first type each of which corresponds to an $\mathcal{I}_u$ in $\mathcal{I}(G)$ for $u \in V(G)$. More specifically, for every $\mathcal{I}_u$, let $(x_i,\ x_j,\ x_k)$ be the order of $D(u)$ with $x_i < x_j < x_k$, then we create a vote $\pi_u = (x_i, x_j, x_k, \mathcal{L}(x_i, x_1], \mathcal{L}(x_i, x_j), \mathcal{L}(x_j, x_k), \mathcal{L}(x_k, c_4])$. Thus, π_u approves $D(u)$. Due to Lemma 2.1, either x_i or x_k lies consecutively with x_j in $\mathcal{L}$, that is, one of $x_i = \overleftarrow{x_j}(1)$ and $x_k = \overrightarrow{x_j}(1)$ must hold, which implies that all votes of the first type are 2-peaked with respect to $\mathcal{L}$. There are only two votes of the second type: $(p, c_1, c_2, c_3, c_4, \mathcal{L}(p,\ x_1])$ and $(p, c_3, c_4, c_1, c_2, \mathcal{L}(p,\ x_1])$. It is clear that these two votes are 2-peaked with respect to $\mathcal{L}$.

In the following, we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{E}$ is a yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S is a vertex cover of size at most κ of G. Then, we delete all votes in $\{\pi_u \mid u \in S\}$. After deleting these votes, no two votes of the first type approve a common candidate in $\mathcal{C}$, since otherwise, $V(G) \setminus S$ could not be an independent set, contradicting the fact that S is a vertex cover. Thus, after deleting these votes all candidates except for the distinguished candidate p have only one point. Since p has two points, p is the unique winner.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Observe that every yes-instance of CCDV-3-Approval (both in general elections and in the 2-peaked elections) has a solution containing only votes which do not approve p. Let Π_S be such a solution of size at most κ, and $\mathcal{E}'$ be the election obtained from $\mathcal{E}$ by deleting all votes in Π_S. Due to the above discussion, p has two points in $\mathcal{E}'$. Since p is the unique winner in $\mathcal{E}'$, every other candidate can have at most one point in $\mathcal{E}'$, which implies that no two votes of the first type approve a common candidate in $\mathcal{C}$ in $\mathcal{E}'$, further implying that the vertices corresponding to the votes in Π_S form a vertex cover of size at most κ for G.

In order to prove that CCDV-r-Approval-UNI in 2-peaked elections is $\mathcal{NP}$-hard for any constant $r \geq 4$, we need to modify the above reduction slightly. First, we add some dummy candidates. More specifically, there are $r-3$ dummy candidates $X_i = \{x_i^1, x_i^2, ..., x_i^{r-3}\}$ with the order $(x_i^1, x_i^2, ..., x_i^{r-3})$ between $x_i \in \Gamma$ and $x_{i+1} \in \Gamma$ in the 2-harmonious order $\mathcal{L}$, whenever there is a $u \in V(G)$ such that $[x_i, x_{i+1}] \in \mathcal{I}_u$. Besides, we have $2r-6$ dummy candidates $c_5, c_6, ..., c_{2r-2}$ lying after c_4 in $\mathcal{L}$, with the order $(c_5, c_6, ..., c_{2r-2})$. Thus, there are $(r-3) \cdot |V(G)| + 2r - 6$ new dummy candidates in total. We change the first type of votes as follows: for every $u \in V(G)$ with $\mathcal{I}_u = \{[x_i, x_{i+1}], [x_j, x_j]\}$ (resp. $\mathcal{I}_u = \{[x_i, x_i], [x_j, x_{j+1}]\}$), we create a vote defined as $(\mathcal{L}[x_i, x_{i+1}], x_j, \mathcal{L}(x_i, x_1], \mathcal{L}(x_{i+1}, x_j), \mathcal{L}(x_j, c_{2r-2}])$ (resp. $(x_i, \mathcal{L}[x_j, x_{j+1}], \mathcal{L}(x_i, x_1], \mathcal{L}(x_i, x_j), \mathcal{L}(x_{j+1}, c_{2r-2}]))$. As for the second type of votes, we have still two votes defined as $(\mathcal{L}[p, c_{2r-2}], \mathcal{L}(p, x_1])$ and $(p, \mathcal{L}[c_r, c_{2r-2}], \mathcal{L}[c_1, c_r), \mathcal{L}(p, x_1])$, respectively. Then, with the same argument, we can show that CCDV-r-Approval-UNI in 2-peaked elections is $\mathcal{NP}$-hard for any $r \geq 4$.

To prove the $\mathcal{NP}$-hardness of CCDV-r-Approval-NON in 2-peaked elections for every $r \geq 3$, we adapt the above reductions in the following way: we create only the first vote in the second type of votes (so that the score of the distinguished candidate is one in the given election) and remaining all the other parts unchanged. □

2.2.2 3-Peaked Elections

In Section 2.2.1, we proved that control by adding votes in r-Approval is polynomial-time solvable when restricted to 2-peaked elections and r being a constant. In this section, we show that the tractability of the problem does not hold when extended to 3-peaked elections.

Theorem 2.4. *Both CCAV-r-Approval-UNI and CCAV-r-Approval-NON in 3-peaked elections are $\mathcal{NP}$-hard for every constant $r \geq 4$.*

Proof. We first prove the $\mathcal{NP}$-hardness of CCAV-4-Approval-UNI in 3-peaked elections by a reduction from Independent Set on bounded degree-3 graphs which is $\mathcal{NP}$-hard [130]. An *independent set* in a graph $G = (V, E)$ is a subset $S \subseteq V$ such that every edge in E has at most one of its endpoints in S.

> Independent Set on Bounded Degree-3 graphs (IS3)
> *Input:* A bounded degree-3 graph $G = (V, E)$ and a positive integer κ.
> *Question:* Does G have an independent set containing exactly κ vertices?

For an instance $\mathcal{F} = (G, \kappa)$ of IS3, let $\mathcal{I}(G)$ be a 2-interval representation of G which satisfies all conditions in Lemma 2.1. Let $D(u), \Gamma$ and $\vec{\Gamma}$ be defined as in the proof for Theorem 2.3. We construct an instance $\mathcal{E} = ((\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}), \Pi_{\mathcal{T}}, \mathcal{L}, \mathcal{R} = \kappa)$ of CCAV-4-Approval-UNI in 3-peaked elections as follows.

Candidates: $\mathcal{C} = \Gamma \cup \{p, c_1, c_2, c_3\}$.

3-Harmonious Order: $\mathcal{L} = (\vec{\Gamma}, p, c_1, c_2, c_3)$.

Registered Votes $\Pi_{\mathcal{V}}$: The role of registered votes is to guarantee that all candidates of Γ have the same score $\kappa - 2$. To this end, we first create $\kappa - 2$ votes defined as $(\mathcal{L}[x_i, x_{i+3}], \mathcal{L}(x_i, x_1], \mathcal{L}(x_{i+3}, c_3])$ for every $i = 1, 5, \ldots, 4\lfloor|\Gamma|/4\rfloor - 3$. Then, we create some further votes according to $|\Gamma|$.

Case 1. $|\Gamma| \equiv 0 \mod 4$. We create no further vote.

Case 2. $|\Gamma| \equiv 1 \mod 4$. We create additional $\kappa - 2$ votes defined as $(x_{|\Gamma|}, \mathcal{L}[c_1, c_3], \mathcal{L}(x_{|\Gamma|}, x_1], p)$.

Case 3. $|\Gamma| \equiv 2 \mod 4$. We create additional $\kappa - 2$ votes defined as $(x_{|\Gamma|-1}, x_{|\Gamma|}, c_1, c_2, \mathcal{L}(x_{|\Gamma|-1}, x_1], p, c_3)$.

Case 4. $|\Gamma| \equiv 3 \mod 4$. We create additional $\kappa - 2$ votes defined as $(\mathcal{L}[x_{|\Gamma|-2}, x_{|\Gamma|}], c_1, \mathcal{L}(x_{|\Gamma|-2}, x_1], p, c_2, c_3)$.

Unregistered Votes $\Pi_{\mathcal{T}}$: For each $u \in V(G)$, let (x_i, x_j, x_k) be the order of $D(u)$ with $x_i < x_j < x_k$. We create a vote $\pi_u = (x_i, x_j, x_k, p, \mathcal{L}(x_i, x_1], \mathcal{L}(x_i, x_j), \mathcal{L}(x_j, x_k), \mathcal{L}(x_k, p), \mathcal{L}(p, c_3])$. Due to Lemma 2.1, either x_i or x_k lies consecutively with x_j in $\mathcal{L}$; thus, all these unregistered votes have 3 peaks x_α, x_β and p where $\{x_\alpha, x_\beta\} \subseteq \{x_i, x_j, x_k\}$ ($\{x_\alpha, x_\beta\}$ depends on whether x_j lies consecutively with x_i or with x_k), with respect to $\mathcal{L}$.

In the following, we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{E}$ is a yes-instance. It is easy to see that, in the election with registered votes, $SC_{\mathcal{V}}(x) = \kappa - 2$ for all $x \in \mathcal{C} \setminus \{p, c_1, c_2, c_3\}$, $SC_{\mathcal{V}}(p) = 0$ and $SC_{\mathcal{V}}(c) \leq \kappa - 2$ for all $c \in \{c_1, c_2, c_3\}$.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S is an independent set of size κ. Then we add all votes corresponding to S, that is, all votes in $\{\pi_u \mid u \in S\}$, to the registered votes. Since S is an independent set, no two added votes approve a common candidate except p; thus, each candidate except p has a final score at most $\kappa - 1$. Since each added vote approves p, it follows that p has a final score of κ points, implying that p becomes the unique winner after adding all votes corresponding to S to the registered votes.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and Π_S is a solution. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all the unregistered votes in Π_S to the registered votes. Clearly, p has a score of κ points in $\mathcal{E}'$. Since p is the unique winner in $\mathcal{E}'$, for every $c \in \mathcal{C} \setminus \{p\}$, there is at most one vote in Π_S approving c. Thus, no two votes in Π_S approve a common candidate except p. Due to the construction, the vertices corresponding to the votes in Π_S must form an independent set.

The proof applies to CCAV-r-Approval-UNI in 3-peaked elections for any constant $r \geq 5$ by a similar modification as discussed in the proof of Theorem 2.3.

To prove CCAV-r-Approval-NON in 3-peaked elections for every $r \geq 4$, we adapt the above reductions. In particular, we adapt the above reductions so that every candidate in Γ has the same score $\kappa - 1$ other than $\kappa - 2$. This can be done easily: replace all appearances of "$\kappa - 2$" with "$\kappa - 1$", and all appearances of "$\kappa - 1$" with "κ" in the above reductions. □

Now we discuss control by modifying candidates. Faliszewski, Hemaspaandra and Hemaspaandra [111] proved that control by deleting candidates in 1-Approval is $\mathcal{NP}$-hard when restricted to Swoon-SP elections, for both the unique-winner model and the nonunique-winner model. Since Swoop-SP elections are a subset of 2-peaked elections, CCDC-1-Approval-UNI and CCDC-1-Approval-NON in k-peaked elections with $k \geq 2$ are $\mathcal{NP}$-hard. We strengthen this result by proving that both CCDC-1-Approval-UNI and CCDC-1-Approval-NON in 3-peaked elections are $\mathcal{W}[1]$-hard with the number of deleted candidates as the parameter.

Theorem 2.5. *Both CCDC-1-Approval-UNI and CCDC-1-Approval-NON are* $\mathcal{W}[1]$-*hard in 3-peaked elections, with respect to the number of deleted candidates.*

Proof. We prove the theorem by $\mathcal{FPT}$-reductions from Independent Set which is $\mathcal{W}[1]$-hard [87]. For a linear order $\vec{A} = (a_1, a_2, ..., a_n)$ over $A = \{a_1, a_2, ..., a_n\}$ and a subset $B \subseteq A$, denote by $\vec{A} \setminus B$ the linear order of $A \setminus B$ obtained from $\vec{A}$ by deleting all elements in B. We first consider CCDC-1-Approval-UNI. For an instance $\mathcal{F} = (G = (V, E), \kappa)$ of Independent Set we construct an instance $\mathcal{E}$ of CCDC-1-Approval-UNI as follows.

Candidates: $V \cup \{p, a, a_1, a_2, ..., a_\kappa, b, b_1, b_2, ..., b_\kappa\}$.

3-Harmonious Order: Let $\overrightarrow{V} = (c_1, c_2, ..., c_n)$ be an (arbitrary but fixed) order of V. Then, the 3-harmonious order $\mathcal{L}$ is given by $(b_\kappa, b_{\kappa-1}, ..., b_1, b, p, a, a_1, a_2, ..., a_\kappa, c_1, c_2, ..., c_n)$.

Votes: There are seven types of votes.

(1) $2|E| - 1$ votes defined as $(\mathcal{L}[a, c_n], \mathcal{L}[p, b_\kappa])$;

(2) $2|E|$ votes defined as $(\mathcal{L}[p, c_n], \mathcal{L}[b, b_\kappa])$;

(3) $2|E| + \kappa - 1$ votes defined as $(\mathcal{L}[b, b_\kappa], \mathcal{L}[p, c_n])$;

(4) for each edge $\{c_i, c_j\} \in E(G)$ with $i < j$, create one vote defined as $(c_i, c_j, \mathcal{L}[a, a_\kappa], \mathcal{L}[p, b_\kappa], \overrightarrow{V} \setminus \{c_i, c_j\})$;

(5) for each vertex c_i, create one vote defined as $(c_i, \mathcal{L}[p, a_\kappa], \mathcal{L}[b, b_\kappa], \overrightarrow{V} \setminus \{c_i\})$ and one vote defined as $(c_i, \mathcal{L}[a, a_\kappa], \mathcal{L}[p, b_\kappa], \overrightarrow{V} \setminus \{c_i\})$;

(6) $\kappa + 1$ votes defined as $(\mathcal{L}[a_1, c_n], \mathcal{L}[a, b_\kappa])$; and

(7) one vote defined as $(\mathcal{L}[b_1, b_\kappa], \mathcal{L}[b, c_n])$.

It is easy to verify that all constructed votes are 3-peaked with respect to $\mathcal{L}$.

Number of Added Candidates: $\mathcal{R} = \kappa$.

($\Leftarrow$:) It is easy to verify that $\mathcal{F}$ is a yes-instance implies $\mathcal{E}$ is a yes-instance: for every independent set S of size κ, deleting the candidates S from the election clearly make the distinguished candidate p become the unique winner.

($\Rightarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and S' is a solution with $|S'| \leq \kappa$. We first observe that $b \notin S'$. This observation is true, since otherwise, all candidates in $\{b_1, b_2, ..., b_\kappa\}$ must be deleted, contradicting that $|S'| \leq \kappa$. The same argument applies to the candidate a. However, in order to make p have a strictly higher score than that of b, exactly κ candidates from V must be deleted so that p can get extra κ points from the constructed votes of type (5). Since $|S'| \leq \kappa$, S' must be a subset of V. Moreover, no two candidates $c_1, c_2 \in S'$ are adjacent to each other in the graph G, since otherwise, the candidate a would get at least one extra point from the constructed votes of type (4), and p cannot be the unique winner. Thus, S' forms an independent set of size κ of G.

CCDC-1-Approval-NON in 3-peaked elections can be proved $\mathcal{W}[1]$-hard by an $\mathcal{FPT}$-reduction obtained from the above reduction by creating one less vote of the second type. That is, we create one less vote defined as $(\mathcal{L}[p, c_n], \mathcal{L}[b, b_\kappa])$. □

2.3 Condorcet, Copeland and Maximin Control

In this section, we study the parameterized complexity of the control problems in 3,4-peaked elections under Condorcet, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Maximin voting correspondences. Recall that in the general case both the constructive control and the destructive control by adding/deleting votes for Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ are $\mathcal{NP}$-hard [110, 112]. Concerning Condorcet, the constructive control by adding/deleting votes is $\mathcal{NP}$-hard while the destructive control by adding/deleing votes is polynomial-time solvable [157]. From the parameterized complexity point of view, Liu and Zhu [184] proved that both the constructive control and the destructive control by adding/deleting votes for Maximin are $\mathcal{W}[1]$-hard in the general case, with respect to the number of added/deleted votes. Moreover, Liu et al. [183] proved that the constructive control by adding/deleting votes for Condorcet is $\mathcal{W}[1]$-hard in the general case, with respect to the number of added/deleted votes. However, their reductions do not apply to 3,4-peaked elections. In this section, we complement their results by proving a cluster of $\mathcal{W}[1]$-hardness results for the control problems under Condorcet, Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ in 3,4-peaked elections. Our strategies to show these $\mathcal{W}[1]$-hardness results in 3,4-peaked elections are technically completely different from the ones used in [183, 184]. Our main results are summarized in Table 2.2.

2.3.1 3-Peaked Elections

This section is devoted to the parameterized complexity of control problems in 3-peaked elections under Maximin, Condorcet and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. We first examine the Maximin voting. It has been proved that both constructive and destructive control by adding votes are $\mathcal{NP}$-hard for Maximin in general [110]. The following theorem shows that both $\mathcal{NP}$-hardness hold even in 3-peaked elections. In fact, from the parameterized complexity point of view, we prove that both problems are $\mathcal{W}[1]$-hard with respect to the number of added votes.

	number of peaks k									Evidence for	
	$k = 1$	$k = 3$				$k \geq 4$				$k = 3$	$k = 4$
	for all	CC		DC		CC		DC			
		AV	DV	AV	DV	AV	DV	AV	DV		
Condorcet	$\mathcal{P}$	**$\mathcal{W}$[1]-h**		$\mathcal{P}$		**$\mathcal{W}$[1]-h**		$\mathcal{P}$		Thm 2.8	Thm 2.11
Maximin		**$\mathcal{W}$[1]-h**		**$\mathcal{W}$[1]-h**		**$\mathcal{W}$[1]-h**				Thm 2.6	Thm 2.9
Copeland$^\alpha$		**$\mathcal{W}$[1]-h**		**$\mathcal{W}$[1]-h**		**$\mathcal{W}$[1]-h**				Thm 2.7	Thm 2.10

Table 2.2: A summary of (parameterized) complexity of control problems under Condorcet, Maximin and Copeland$^\alpha$ in k-peaked elections. Here, "$\mathcal{W}$[1]-h" stands for $\mathcal{W}$[1]-hard and "$\mathcal{P}$" stands for polynomial-time solvable. Our results are in bold. Moreover, the results for Copeland$^\alpha$ apply to all $0 \leq \alpha \leq 1$. The $\mathcal{W}$[1]-hardness results are with respect to the number of added votes for the control by adding votes, and with respect to the number of deleted votes for the control by deleting votes. Note that when $k = m/2 + 1$, k-peaked elections are general elections, where m is the number of candidates. The polynomial-time solvability results in single-peaked elections (1-peaked elections) are from [44]. The polynomial-time solvability of the destructive control by adding/deleting votes for Condorcet is from [157].

Theorem 2.6. *DCAV-Maximin-UNI, DCAV-Maximin-NON, CCAV-Maximin-UNI and CCAV-Maximin-NON in 3-peaked elections are all $\mathcal{W}[1]$-hard with respect to the number of added votes.*

Proof. We first prove the $\mathcal{W}[1]$-hardness for DCAV-Maximin-UNI in 3-peaked elections by an $\mathcal{FPT}$-reduction from INDEPENDENT SET on 2-interval graphs which is $\mathcal{W}[1]$-hard [117]. For a given instance $\mathcal{F} = (\mathcal{I} = (I_1, I_2, ..., I_n), \kappa)$ of the INDEPENDENT SET problem on 2-interval graphs, we construct an instance $\mathcal{E}$ for DCAV-Maximin-UNI in 3-peaked elections as follows. We denote by I_i^1 and I_i^2 the two intervals of I_i. Let $D(I_i)$ be the endpoints of I_i, and let $\Gamma = \cup_{i\in[n]} D(I_i)$. Moreover, let $\vec{\Gamma} = (x_1, x_2, ..., x_{|\Gamma|})$ be the order of Γ with $x_i < x_{i+1}$ for all $i \in [|\Gamma| - 1]$.

Candidates: $\mathcal{C} = \Gamma \cup \{p, q\}$ where q is the distinguished candidate. Concretely, for each $x \in \Gamma$, we create a candidate. For ease of exposition, we still use x to denote the candidate corresponding to the endpoint x.

3-Harmonious Order: $\mathcal{L} = (q, \vec{\Gamma}, p)$.

Registered Votes: We create $3\kappa - 1$ registered votes in total. Concretely, we create $2\kappa - 1$ registered votes defined as $\mathcal{L}[q, p]$, and κ registered votes defined as $(p, \mathcal{L}[q, x_{|\Gamma|}])$. The comparisons between every two candidates, based on the registered votes, are summarized in Table 2.3.

	p	q	$x_j (i < j)$	$x_j (i > j)$
p	-	κ	κ	
q	$2\kappa - 1$	-	$3\kappa - 1$	
x_i	$2\kappa - 1$	0	$3\kappa - 1$	0

Table 2.3: Comparisons between every two candidates in the $\mathcal{W}[1]$-hardness reduction for DCAV-Maximin-UNI in Theorem 2.6. Each entry with row indicated by candidate c and column indicated by candidate c' is $N(c, c')$, the number of registered votes ranking c above c'. Here, the value of $N(\cdot)$ is based on the registered votes.

Unregistered Votes: The unregistered votes are created according to the intervals in $\mathcal{F}$. Precisely, for every 2-interval $I_i = \{I_i^1, I_i^2\}$, we create an unregistered vote. Let x_α and x_β with $x_\alpha \le x_\beta$ denote the left endpoint and the right endpoint of I_i^1 respectively, and x_γ and x_δ with $x_\gamma \le x_\delta$ denote the left endpoint and the right endpoint of I_i^2 respectively. Without loss of generality, assume that I_i^2 is on the right side of I_i^1, that is $x_\beta < x_\gamma$. The unregistered vote π_{I_i} corresponding to I_i is defined as $(\mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta], p, \mathcal{L}[x_{\alpha-1}, x_1], \mathcal{L}[x_{\beta+1}, x_{\gamma-1}], \mathcal{L}[x_{\delta+1}, x_{|\Gamma|}], q)$. See Figure 2.6 for an illustration.

Number of Added Votes: $\mathcal{R} = \kappa$.

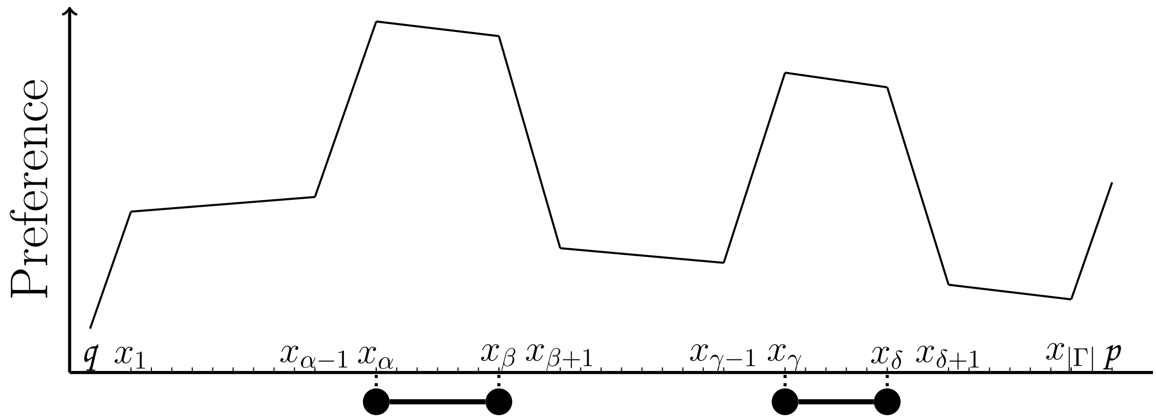


Figure 2.6: An illustration of an unregistered vote corresponding to a 2-interval in the $\mathcal{NP}$-hardness reduction for DCAV-Maximin-UNI in 3-peaked elections in Theorem 2.6.

Now we come to show the correctness of the reduction. First observe that q is the current winner with Maximin score $2\kappa - 1$. Moreover, the Maximin score of q cannot increase by adding unregistered votes to registered votes, since q is ranked below every other candidate in every unregistered vote; and thus, q will have a Maximin score $2\kappa - 1$ in the final election. Furthermore, every $x_i \in \Gamma$ cannot have a no less Maximin score than that of q by adding at most κ votes. This is because $N(x_i, q) = 0$ with respect to the registered votes; and thus, the maximum Maximin score of every x_i is at most κ in the final election. Therefore, the only candidate which has chance to have a no less score than q in the final election is the candidate p.

($\Rightarrow$:) Suppose that $\mathcal{F}$ has an independent set S of size κ. We claim that q is no longer the unique winner after adding all unregistered votes corresponding to S to the registered votes. Let $\Pi_S = \{\pi_I \mid I \in S\}$ be the set of the unregistered votes corresponding to S. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all the votes in Π_S to the registered votes. Due to the construction of the unregistered votes and the fact that S is an independent set, we have that for every $x_i \in \Gamma$ there is at most one vote in Π_S which ranks x_i above p. This implies that $N_{\mathcal{E}'}(p, x_i) \geq 2\kappa - 1$. Moreover, since p is ranked above q in every unregistered vote, $N_{\mathcal{E}'}(p, q) = 2\kappa$. It is now easy to see q is no longer the unique winner.

($\Leftarrow$:) Suppose that q is not the unique winner after adding at most κ unregistered votes to the registered votes. Let Π_S be the unregistered votes added to the registered votes. Due to the above discussion, we know that p has a no less Maximin score than that of q in the final election. Since q has a Maximin score $2\kappa - 1$ in the final election, for every candidate x_i there has to be at least $\kappa - 1$ votes in Π_S which rank p above x_i. Due to the construction of the unregistered votes, this happens only if there is an independent set of size κ in $\mathcal{F}$.

The proof for DCAV-Maximin-NON is similar to the proof for DCAV-Maximin-UNI with the difference that in the construction we have one less registered vote defined as $\mathcal{L}[q, p]$.

To prove the CCAV-Maximin-NON, we adopt exactly the same construction as for DCAV-Maximin-UNI but with p being the distinguished candidate. The reduction applies to CCAV-Maximin-NON since under the construction, p is the only candidate who has chance to replace q as the winner by adding at most κ unregistered votes, as discussed above.

To prove the CCAV-Maximin-UNI, we need to adopt the construction for DCAV-Maximin-NON and set p as the distinguished winner. The reason why the reduction works here is the same as for CCAV-Maximin-NON. □

Now we examine Copeland control in 3-peaked elections. Both constructive and destructive control by adding votes are $\mathcal{NP}$-hard for Copeland$^\alpha$ for every $0 \le \alpha \le 1$ in general [112]. Same as the complexity we have proved for Maximin, we show that both problems are indeed $\mathcal{W}[1]$-hard even in 3-peaked elections, with respect to the number of added votes.

Theorem 2.7. *DCAV-Copeland$^\alpha$-UNI, DCAV-Copeland$^\alpha$-NON, CCAV-Copeland$^\alpha$-UNI and CCAV-Copeland$^\alpha$-NON in 3-peaked elections are all $\mathcal{W}[1]$-hard for every $0 \le \alpha \le 1$, with respect to the number of added votes.*

Proof. We first show the proof for DCAV-Copeland^0-NON in 3-peaked elections from an $\mathcal{FPT}$-reduction from the Independent Set problem on 2-interval graphs which is $\mathcal{W}[1]$-hard [117]. Given an instance $\mathcal{F} = (\mathcal{I} = (I_1, I_2, ..., I_n), \kappa)$ of the Independent Set problem on 2-interval graphs, we construct an instance $\mathcal{E}$ for DCAV-Copeland^0-NON in 3-peaked elections as follows. The notations $I_i^1, I_i^2, D(I_i), \Gamma$ and $\vec{\Gamma}$ hereinafter are defined in the same way as in the proof of Theorem 2.6.

Candidates: $\mathcal{C} = \Gamma \cup \{p, q, y\}$ where q is the distinguished candidate.

3-Harmonious Order: $\mathcal{L} = (q, \vec{\Gamma}, p, y)$.

Registered Votes: We create $3\kappa - 3$ registered votes in total. Concretely, we create $2\kappa - 3$ registered votes defined as $(q, y, \mathcal{L}[x_1, p])$, and κ registered votes defined as $(p, q, y, \mathcal{L}[x_1, x_{|\Gamma|}])$. It is easy to verify that q is a Copeland^0 winner (precisely, q is the current unique winner). The comparisons between every two candidates, base on the registered votes, are summarized in Table 2.4.

Unregistered Votes: The unregistered votes are created according to the intervals in $\mathcal{F}$. Precisely, for every 2-interval $I_i = \{I_i^1, I_i^2\}$, we create an unregistered vote. Let x_α and x_β with $x_\alpha \le x_\beta$ denote the left endpoint and the right endpoint of I_i^1 respectively, and x_γ and x_δ with $x_\gamma \le x_\delta$ denote the left endpoint and the right endpoint of I_i^2 respectively. Without loss of generality, assume that I_i^2 is on the right side of I_i^1, that is $x_\beta < x_\gamma$. The unregistered vote π_{I_i} corresponding to I_i is defined as $(\mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta], p, y, \mathcal{L}[x_{\alpha-1}, x_1], \mathcal{L}[x_{\beta+1}, x_{\gamma-1}], \mathcal{L}[x_{\delta+1}, x_{|\Gamma|}], q)$.

	p	q	$x_j(i<j)$	$x_j(i>j)$	y
p	-	κ	κ		
q	$2\kappa-3$	-	$3\kappa-3$		
x_i	$2\kappa-3$	0	$3\kappa-3$	0	
y	$2\kappa-3$	0	$3\kappa-3$		-

Table 2.4: Comparisons between every two candidates in the $\mathcal{W}[1]$-hardness reduction for DCAV-Copeland^0-NON in Theorem 2.7. Each entry with row indicated by candidate c and column indicated by candidate c' is $N(c,c')$, the number of registered votes ranking c above c'. Here, the value of $N(c,c')$ is based on the registered votes.

Number of Added Votes: $\mathcal{R} = \kappa$.

Now we prove the correctness.

($\Rightarrow$:) Suppose that $\mathcal{F}$ has an independent set S of size κ. Consider the election $\mathcal{E}'$ obtained from $\mathcal{E}$ by adding all the unregistered votes in $\Pi_S = \{\pi_I \mid I \in S\}$ corresponding to S to the registered votes. We have in total $4\kappa - 3$ votes in $\mathcal{E}'$. Since p is ranked above q and y in every unregistered vote, p beats both q and y in $\mathcal{E}'$. Due to the construction of the unregistered votes and the fact that S is an independent set, for every $x_i \in \Gamma$ there is at most one unregistered vote in Π_S which ranks x_i above p. Therefore, $N_{\mathcal{E}'}(p, x_i) \geq 2\kappa - 1$, implying that p beats every candidate $x_i \in \Gamma$. Summary all above, p beats every other candidate in $\mathcal{E}'$, implying that q is no longer a Copeland^0 winner.

($\Leftarrow$:) Suppose that Π_S is the multiset of unregistered votes added to the registered votes which makes q no longer a Copeland^0 winner. Here, the index S is the set of 2-intervals corresponding to Π_S in $\mathcal{F}$. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all votes in Π_S to the registered votes. We first illustrate that the candidate p is the only one which has chance to have a no less Copeland^0 score than that of q in $\mathcal{E}'$. Observe first that the comparisons between the candidates y and q, and between every candidate x_i and q cannot be changed by adding at most κ unregistered votes. Therefore, the candidate q beats y and every x_i in $\mathcal{E}'$. In this case, in order to prevent q from being a Copeland^0 winner, q has to be beaten by p in $\mathcal{E}'$, since otherwise, q would beat every other candidate and thus remains the winner (actually remains as the Condorcet winner). However, once q is beaten by p in $\mathcal{E}'$, y is also beaten by p in $\mathcal{E}'$. Hence, y has no chance to have a strictly greater Copeland^0 score than that of q. Analogously, every $x_i \in \Gamma$ cannot have a strictly greater Copeland^0 score than that of q since every x_i is beaten by y and q in $\mathcal{E}'$. Therefore, the only candidate which has chance to have a no less score than that of q is the candidate p. Since q beats every candidate except p, in order to make p have a no less score than that of q, p has to beat every other candidate. This happens only if Π_S contains κ unregistered votes,

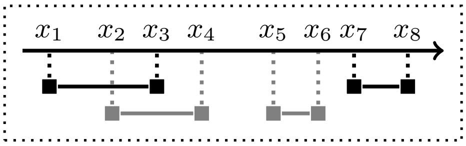


Figure 2.7: An illustration of the restriction in the $\mathcal{W}[1]$-hardness reduction for DCAV-Copeland^0-UNI in Theorem 2.7. Once two 2-intervals intersect, they intersect at more than one point.

and moreover, for every candidate x_i there is at most one vote in Π_S ranking x_i above p. The latter condition directly implies that S, the set of 2-intervals corresponding to the votes in Π_S, is an independent set, and the former condition implies that $|S| = \kappa$. The proof for DCAV-Copeland^0-NON is finished.

The above reduction does not apply to DCAV-Copeland^0-UNI directly, since, in this case, q could also become not a unique-winner when there is no independent set of size κ for $\mathcal{F}$. To check this, consider the situation where p is beaten by some $x \in \Gamma$ (but p beats every other candidate in Γ) in the final election. This can happen when we add two unregistered votes corresponding to two 2-intervals which intersect only at x to the election. In this situation, p beats every other candidate except x and q beats every other candidate except p, implying that q is no longer a unique winner. In order to prove the hardness of DCAV-Copeland^0-UNI, we need to restrict the 2-intervals in $\mathcal{F}$ in such a way that once two 2-intervals intersect, they do not intersect at only one point, but in a non-trivial interval. See Figure 2.7 for an illustration. This restriction does not change the $\mathcal{W}[1]$-hardness of the Independent Set problem on 2-interval graphs [117]. Under this restriction, once two unregistered votes corresponding to two 2-intervals that intersect are added to the registered votes, p will be beaten by at least two candidates in Γ, implying that p cannot prevent q from being the unique Copeland^0 winner if there is no independent set of size κ for $\mathcal{F}$. Remaining other parts of the proof the same as for DCAV-Copeland^0-NON, the $\mathcal{W}[1]$-hardness for DCAV-Copeland^0-UNI follows.

Now we move to the hardness of CCAV-Copeland^0-UNI. The reduction is exactly the same as for DCAV-Copeland^0-NON with only the difference that we set p as the distinguished candidate. We have argued in the proof for DCAV-Copeland^0-NON that if $\mathcal{F}$ is a yes-instance, p can be made the Copeland^0 unique-winner in the final election. Our argument for the other direction is also the same as for DCAV-Copeland^0-NON. Actually, the argument can be simpler here since we do not need to argue that y and every $x \in \Gamma$ have no chance to be the final winner, since we have set p as the distinguished candidate.

The proof for CCAV-Copeland^0-NON is exactly the same as for DCAV-Copeland^0-UNI with only the difference that p is the distinguished candidate. On the one hand, if there is an independent set of size κ, we can make p a winner by adding the unregistered votes corresponding to the independent set. On the other hand, if there is no independent set of size κ, there must be at least two candidates in Γ, which are ranked above p simultaneously in two unregistered votes that are added to the registered votes, implying that p cannot be a winner in the final election. To check the latter argument, one should first observe that we cannot make p a winner by adding at most $\kappa - 1$ unregistered votes: if this is the case, there must be no less than two candidates from Γ which beat or tie p; however, q beats every other candidate except p (if adding less than $\kappa - 2$ unregistered votes, q will beat every other candidate).

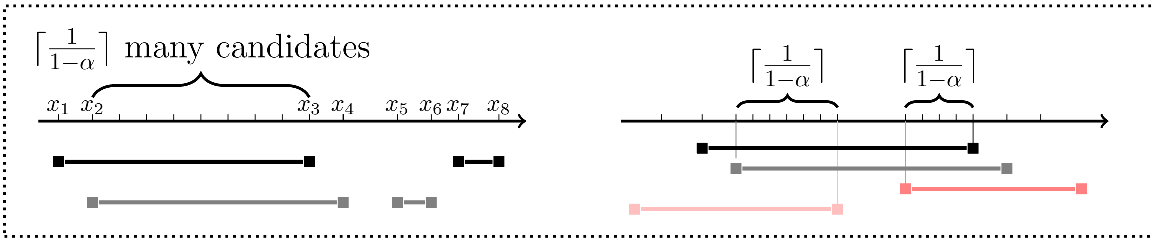


Figure 2.8: An illustration of the dummy candidates in the $\mathcal{W}[1]$-hardness reduction for DCAV-Copeland$^\alpha$-UNI in Theorem 2.7. The figure on the left side shows two 2-intervals (the black one above and the gray one below) which intersect in a non-trivial interval $[x_2, x_3]$. Moreover, there is no other 2-interval whose endpoint is in $[x_2, x_3]$. We create exactly $\lceil \frac{1}{1-\alpha} \rceil$ dummy candidates between x_2 and x_3 (including x_2 and x_3). The figure on the right side shows four 2-intervals, with only one interval of each 2-intervals is showed. There are $\lceil \frac{1}{1-\alpha} \rceil$ candidates corresponding to the intersection (this is a minimal intersection since no other 2-interval has any of its endpoints in this intersection) of the second 2-interval and the fourth 2-interval, and $\lceil \frac{1}{1-\alpha} \rceil$ candidates corresponding to the intersection (a minimal intersection) of the first 2-interval and the third 2-interval.

Now we come to Copeland$^\alpha$ control for every $0 < \alpha < 1$. In the following, let α be a fixed real number with $0 < \alpha < 1$. We first consider DCAV-Copeland$^\alpha$-UNI. The $\mathcal{W}[1]$-hardness reduction is adapted from that for DCAV-Copeland^0-UNI by creating polynomially many dummy candidates. The role of these dummy candidates is to enlarge the score gap between p and q to a certain extent, when two unregistered votes corresponding to two intersected 2-intervals are added to the registered votes; hence guarantees that q would be still the unique winner when a multiset of at most κ unregistered votes corresponding to a non independent set are added. To this end, we also adopt the same restriction on 2-intervals here as for DCAV-Copeland^0-UNI: every two 2-intervals either do not intersect or they intersect in a non-trivial interval. Precisely, we create these dummy candidates in a way so that any intersection of two 2-intervals corresponds to no less than $\lceil \frac{1}{1-\alpha} \rceil$ such dummy candidates. To this end, we do the following. We call an intersection $[x_1, x_2]$ with $x_1 < x_2$ of two 2-intervals a *minimal intersection* if there is no other 2-interval which has at least one of its endpoints in $[x_1, x_2]$. Clearly, given a 2-interval representation of an instance of Independent Set on 2-interval graphs, all minimal intersections can be found in polynomial time. Apart from creating all the candidates as in the reduction for DCAV-Copeland^0-UNI, we create, for each minimal intersection $[x_1, x_2]$, a set of $\lceil \frac{1}{1-\alpha} \rceil$ dummy candidates which lie in distinguished places in $[x_1, x_2]$. See Figure 2.8 for an illustration. This construction ensures that p can prevent q from being the unique winner only if p beats every other candidate in the final election. The observation is that if p ties or be beaten by some candidate $x \in \Gamma$, then p also ties or be beaten by no less than $\lceil \frac{1}{1-\alpha} - 1 \rceil$ other candidates. The amount $\lceil \frac{1}{1-\alpha} \rceil$ is enough to make p have a strictly less score than that of q; and hence cannot prevent q from being the unique winner. However, p beats every other candidate if and only if there is an independent set of size κ for $\mathcal{F}$, implying the correctness of the reduction. The proofs for other three problems DCAV-Copeland$^\alpha$-NON, CCAV-Copeland$^\alpha$-UNI and CCAV-Copeland$^\alpha$-NON are adapted from DCAV-Copeland^0-NON, CCAV-Copeland^0-UNI and CCAV-Copeland^0-NON, respectively. The constructions are analogous to the above reduction.

Finally we consider Copeland^1. The reductions for the four problems DCAV-Copeland^1-UNI, DCAV-Copeland^1-NON, CCAV-Copeland^1-UNI, CCAV-Copeland^1-NON are adapted from DCAV-Copeland^0-UNI, DCAV-Copeland^0-NON, CCAV-Copeland^0-UNI, CCAV-Copeland^0-NON, respectively. Precisely, each reduction is different from the corresponding one in the way that we set $\mathcal{R} = \kappa - 1$. Moreover, instead of searching for an independent set of size κ, we search for an independent set of size $\kappa - 1$. □

Now we come to Condorcet. The following theorem summarizes our findings for constructive control by adding votes for Condorcet in 3-peaked elections. Recall that in general, the constructive control by adding votes for Condorcet is $\mathcal{NP}$-hard, while the destructive control by adding votes is polynomial-time solvable [157].

Theorem 2.8. *CCAV-Condorcet-UNI and CCAV-Condorcet-NON in 3-peaked elections are $\mathcal{W}[1]$-hard with respect to the number of added votes.*

Proof. The proof for CCAV-Condorcet-UNI is exactly the same as for CCAV-Copeland^0-UNI. The proof for CCAV-Condorcet-NON is similar to the one for CCAV-Condorcet-UNI with the difference that we create one more registered vote defined as $(q, y, \mathcal{L}[x_1, p])$. □

2.3.2 4-Peaked Elections

In the previous sections, we have discussed control by adding votes in 3-peaked elections. In this section, we consider control by deleting votes in 4-peaked elections for Condorcet, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Maximin. We first examine the Maximin voting. It is known that both constructive and destructive control by deleting votes are $\mathcal{NP}$-hard for Maximin in general [110]. The following theorem shows both problems are $\mathcal{W}[1]$-hard even in 4-peaked elections, with respect to the number of added votes.

Theorem 2.9. *CCDV-Maximin-UNI, CCDV-Maximin-NON, DCDV-Maximin-UNI and DCDV-Maximin-NON are $\mathcal{W}[1]$-hard in 4-peaked elections, with respect to the number of deleted votes.*

Proof. Our $\mathcal{FPT}$-reductions are again from INDEPENDENT SET on 2-interval graphs which is $\mathcal{W}[1]$-hard [117]. Moreover, we adopt another restriction on the 2-intervals (different from the one in the proof of Theorem 2.7). For two 2-intervals I and J, we say I covers J if $J \subseteq I$. See Figure 2.9 for an illustration. We restrict the given instance of INDEPENDENT SET on 2-interval graphs in a way so that there is no 2-interval which is covered by another 2-interval. This does not change the $\mathcal{W}[1]$-hardness of the problem [117].

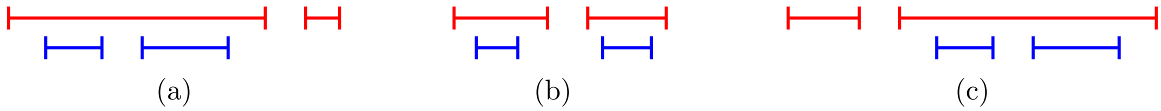


Figure 2.9: This figure shows three different ways of how a red 2-interval covers a blue 2-interval. The two 2-intervals are draw on different levels for the sake of clarity. However, they are actually both defined on the real line.

Let $\mathcal{F} = (\mathcal{I} = (I_1, I_2, ..., I_n), \kappa)$ be a given instance of INDEPENDENT SET on 2-interval graphs. The following construction applies to both CCDV-Maximin-UNI and DCDV-Maximin-NON. We will discuss the construction for the other two problems later. Hereby, $I_i^1, I_i^2, D(I_i), \Gamma = \cup_i D(I_i)$ and $\vec{\Gamma} = (x_1, x_2, ..., x_{|\Gamma|})$ are defined in the same way as in the proof of Theorem 2.6.

Candidates: $\mathcal{C} = \Gamma \cup \{p, q, x_0\}$, where p is the distinguished candidate in CCDV-Maximin-UNI, while q is the distinguished candidate in DCDV-Maximin-NON.

4-Harmonious Order: $\mathcal{L} = (q, x_0, \vec{\Gamma}, p)$.

Votes: We create $4n - \kappa + 2$ votes in total. Concretely, we first create the following $2n - \kappa + 2$ votes (number of votes: votes represented by linear orders).

$n : (\mathcal{L}[x_0, p], q)$

$n - \kappa : (\mathcal{L}[p, x_0], q)$

$2 : (p, q, \mathcal{L}[x_0, x_{|\Gamma|}])$

Then, for every 2-interval $I_i = \{I_i^1, I_i^2\}$ of $\mathcal{F}$, we create two votes $\pi_{I_i^1}$ and $\pi_{I_i^2}$ as follows. Let x_α and x_β with $x_\alpha \leq x_\beta$ denote the left endpoint and the right endpoint of I_i^1, respectively, and x_γ and x_δ with $x_\gamma \leq x_\delta$ denote the left endpoint and the right endpoint of I_i^2, respectively. Without loss of generality, assume that I_i^2 is on the right side of I_i^1, that is $x_\beta < x_\gamma$. The two votes corresponding to I_i are defined as follows.

$\pi_{I_i^1} = (\mathcal{L}[q, x_{\alpha-1}], \mathcal{L}(x_\beta, x_\gamma), \mathcal{L}(x_\delta, p], \mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta]);$

$\pi_{I_i^2} = (q, \mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta], p, \mathcal{L}[x_0, x_{\alpha-1}], \mathcal{L}(x_\beta, x_\gamma), \mathcal{L}(x_\delta, x_{|\Gamma|}])$.

It is easy to check that $\pi_{I_i^1}$ has three peaks q, x_β, x_δ, and $\pi_{I_i^2}$ has four peaks q, x_α, x_γ, p. In the following, let $\Pi_1 = \{\pi_{I_i^1} \mid i = 1, 2, ..., n\}$ and $\Pi_2 = \{\pi_{I_i^2} \mid i = 1, 2, ..., n\}$.

Number of Deleted Votes: $\mathcal{R} = \kappa$.

We denote by $\mathcal{E}$ the above constructed election instance. The comparisons between every two candidates are shown in Table 2.5.

	p	q	$x_j(j > i)$	$x_j(j < i)$
p	-	$2n - \kappa + 2$	$2n - \kappa + 2$	
q	$2n$	-	$2n + 2$	
x_i	$2n$	$2n - \kappa$	$\times$	$\times$

Table 2.5: Comparisons between every two candidates for the $\mathcal{W}[1]$-hardness reductions for CCDV-Maximin-UNI and DCDV-Maximin-NON in Theorem 2.9. The comparisons between x_i and x_j are marked with $\times$ since they cannot be exactly determined. However, they do not play any role in the correctness argument. What is important for x_i is the comparison between x_i and q, which implies that the final Maximin score of every x_i can be at most $2n - \kappa$.

It is clear that q has the maximum Maximin score $2n$ and is thus the unique Maximin winner. We now show the correctness for the CCDV-Maximin-UNI.

($\Rightarrow$:) Suppose that $\mathcal{F}$ has an independent set S of size κ. We claim that we can make p the unique Maximin winner by deleting votes corresponding to S in Π_1. Let $\Pi_S = \{\pi_{I^1} \mid I \in S\}$ be the set of the votes corresponding to S in Π_1, and let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in Π_S. Since S is an independent set, we have that for every $x_i \in \Gamma$ there is at most one vote in Π_S which ranks p above x_i. This implies that $N_{\mathcal{E}'}(p, x_i) \geq 2n - \kappa + 1$ for every $x_i \in \Gamma$. Moreover, since q is ranked above p in every vote in Π_S, $N_{\mathcal{E}'}(p, q) = 2n - \kappa + 2$ and $N_{\mathcal{E}'}(q, p) = 2n - \kappa$. Therefore, the Maximin score of p is at least $2n - \kappa + 1$ while the Maximin score of q is at most $2n - \kappa$. Finally, since q is ranked above every $x_i \in \Gamma$ in every vote in Π_S, we have that $N_{\mathcal{E}'}(x_i, q) = 2n - \kappa$, implying that every x_i has a Maximin score at most $2n - \kappa$. Summary all above, we know that p is the unique winner with Maximin score at least $2n - \kappa + 1$ in the final election.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Let Π_S be a solution of $\mathcal{E}$, and $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in Π_S. Observe first that Π_S contains no vote which ranks p above q, since otherwise, $N_{\mathcal{E}'}(p, q) \leq 2n - \kappa + 1$ and $N_{\mathcal{E}'}(q, p) \geq 2n - \kappa + 1$, contradicting with the fact that p is the unique winner in $\mathcal{E}'$. Since we can delete at most κ votes and the Maximin score of q is $2n$ in the original election $\mathcal{E}$, the final Maximin score of q is at least $2n-\kappa$. Since p is the unique winner in the final election $\mathcal{E}'$, the Maximin score of p is at least $2n-\kappa+1$ in $\mathcal{E}'$. Therefore, for every $x_i \in \Gamma$ there is at most one vote in Π_S which ranks p above x_i. Due to the fact, we have the following claim.

Claim. Π_S contains no vote in Π_2.

(Proof of the Claim.) We prove the claim by contradiction. Suppose that $\pi_{I_i^2} \in \Pi_S \cap \Pi_2$ is a vote corresponding to a 2-interval I_i. Let A be the set of candidates which lie in the 2-interval I_i. Due to the construction, all the candidates in $\Gamma \setminus A$ are ranked below p. Let I_j be another 2-interval which corresponds to another vote $\pi_{I_j^u} \neq \pi_{I_i^2}$ (observe that Π_S contains at least two votes, since otherwise q would have a too large Maximin score). Let B be the set of candidates which lie in the 2-interval I_j. Due to the restriction of the instance, we know that $B \setminus A \neq \emptyset$. Therefore, $u \neq 1$, since otherwise, both $\pi_{I_j^u}$ and $\pi_{I_i^2}$ rank p above every candidate in $B \setminus A$. However, it also cannot be the case that $u=2$, since otherwise, both $\pi_{I_j^u}$ and $\pi_{I_i^2}$ rank p above the candidate x_0[iii]. See Figure 2.10 for an illustration.

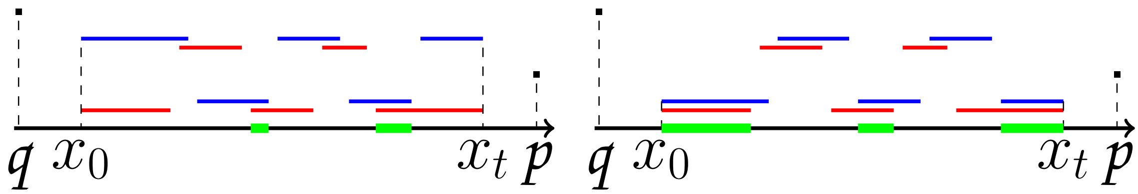


Figure 2.10: An illustration of the Claim in the proof of Theorem 2.9. Here, $t = |\Gamma|$ In both the left-hand figure and the right-hand figure. Most comparisons among the candidates in Γ are not explicitly showed. Moreover, the figure on the left side shows the case that $u=1$, and the figure on the right side shows the case that $u=2$. In either case, the candidates lie in the green interval are ranked below p in the two votes corresponding to the red 2-interval and the blue 2-interval.

Due to the above claim, we know that $\Pi_S \subseteq \Pi_1$. Let S be the set of 2-intervals corresponding to Π_S. Since for every $x_i \in \Gamma$ there is at most one vote in Π_S which ranks p above x_i, there is no two 2-intervals in S which intersect, implying that $\mathcal{F}$ has an independent set of size κ.

To check that the same reduction applies to DCDV-Maximin-NON, observe first that no $x_i \in \Gamma$ can have a higher Maximin score than that of q in the final election: since $N_{\mathcal{E}'}(x_i, q) = 2n-\kappa, N_{\mathcal{E}'}(q,p) = 2n$ and we can delete at most κ votes, every x_i would have a Maximin score at most $2n-\kappa$ and q would have a Maximin score at least $2n-\kappa$ in the final election. Due to the above analysis, p is the only candidate which can prevent q from being a winner. This turns the problem into exactly CCDV-Maximin-UNI. The above argument for CCDV-Maximin-UNI then works.

[iii] The dummy candidate x_0 can be deleted from the construction without destroying the correctness. However, the introducing of x_0 simplifies the exposition of the proof.

	p	q	$x_j(j > i)$	$x_j(j < i)$
p	-	$2n - \kappa + 1$	$2n - \kappa + 1$	
q	$2n$	-	$2n + 1$	
x_i	$2n$	$2n - \kappa$	$\times$	$\times$

Table 2.6: Comparisons between every two candidates in the $\mathcal{W}[1]$-hardness reductions for CCDV-Maximin-NON and DCDV-Maximin-UNI in Theorem 2.9. The comparisons between x_i and x_j are marked with $\times$ since they cannot be exactly determined. However, they do not play any role in the correctness argument. What is important for x_i is the comparison between x_i and q, which implies that the final Maximin score of every x_i can be at most $2n - \kappa$.

Now we discuss the reductions for CCDV-Maximin-NON and DCDV-Maximin-UNI. Analogously, we adopt the same reduction as discussed above for CCDV-Maximin-UNI, with only the difference that we create only one vote defined as $(p, q, \mathcal{L}[x_0, x_{|\Gamma|}])$, other than two. Moreover, in CCDV-Maximin-NON we set p as the distinguished candidate, while in DCDV-Maximin-UNI we set q as the distinguished candidate. The comparisons between every two candidates are shown in Table 2.6. The correctness argument for CCDV-Maximin-NON and DCDV-Maximin-UNI is similar to that for CCDV-Maximin-UNI and DCDV-Maximin-NON, respectively. The difference is that for CCDV-Maximin-NON and DCDV-Maximin-UNI, we argue that p can have a no less Maximin score than that of q by deleting at most κ votes if and only if there is an independent set of size κ, other than requiring p to have a strictly higher Maximin score than that of q in the final election as for CCDV-Maximin-UNI and DCDV-Maximin-NON. This difference is accurately reflected in the one less creation of the vote defined as $(p, q, \mathcal{L}[x_0, x_{|\Gamma|}])$. □

Now we study Copeland$^\alpha$ control by deleting votes in 4-peaked elections. Recall that in general, both the constructive control and the destructive control by deleting votes for Copeland$^\alpha$ are $\mathcal{NP}$-hard, for every $0 \leq \alpha \leq 1$ [112]. Our results concerning the same problems in 4-peaked elections are summarized in the following theorem.

Theorem 2.10. *CCDV-Copeland$^\alpha$-UNI, CCDV-Copeland$^\alpha$-NON and DCDV-Copeland$^\alpha$-UNI and DCDV-Copeland$^\alpha$-NON for every $0 \leq \alpha \leq 1$ are $\mathcal{W}[1]$-hard in 4-peaked elections, with respect to the number of deleted votes.*

Proof. Our reductions are again from the Independent Set problem on 2-interval graphs. Moreover, we adopt the restriction on 2-intervals as in the proof of Theorem 2.7 that every two 2-intervals either do not intersect or intersect at more than one point. This does not change the $\mathcal{W}[1]$-hardness of the problem [117]. Given instance $\mathcal{F} = (\mathcal{I} = (I_1, I_2, ..., I_n), \kappa)$ of the Independent Set problem on 2-interval graphs, we construct instances $\mathcal{E}$ for the problems stated in Theorem 2.10 as follows. We first consider CCDV-Copeland$^\alpha$-UNI, CCDV-Copeland$^\alpha$-NON and DCDV-Copeland$^\alpha$-NON. Hereby, $I_i^1, I_i^2, D(I_i), \Gamma = \cup_{i\in[n]} I_i$ and $\vec{\Gamma} = (x_1, x_2, ..., x_{|\Gamma|})$ are defined in the same way as in the proof of Theorem 2.6.

Candidates: $\mathcal{C} = \Gamma \cup \{p, q\}$.

4-Harmonious Order: $\mathcal{L} = (q, \vec{\Gamma}, p)$.

Votes: We create $4n - \kappa + 1$ votes in total. Precisely, we first create $2n - \kappa + 1$ votes as follows (number of votes: votes represented by linear orders).

$n - 1 : (\mathcal{L}[x_1, p], q)$

$n - \kappa + 2 : (p, \mathcal{L}[q, x_{|\Gamma|}])$

Then, for every 2-interval $I_i = \{I_i^1 = [x_\alpha, x_\beta], I_i^2 = [x_\gamma, x_\delta]\}$ of $\mathcal{F}$ we create two votes. Without loss of generality, assume that I_i^2 is on the right side of I_i^1, that is $x_\beta < x_\gamma$. The two votes corresponding to I_i are defined as follows.

$\pi_{I_i^1} = (\mathcal{L}[q, x_{\alpha-1}], \mathcal{L}(x_\beta, x_\gamma), \mathcal{L}(x_\delta, p], \mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta]);$

$\pi_{I_i^2} = (q, \mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta], p, \mathcal{L}[x_1, x_{\alpha-1}], \mathcal{L}(x_\beta, x_\gamma), \mathcal{L}(x_\delta, x_{|\Gamma|}]).$

In the following, let $\Pi_1 = \{\pi_{I_i^1} \mid i = 1, 2, ..., n\}$ and $\Pi_2 = \{\pi_{I_i^2} \mid i = 1, 2, ..., n\}$. It is easy to verify that all votes have at most 4 peaks with respect to the 4-harmonious order $\mathcal{L}$.

Number of Deleted Votes: $\mathcal{R} = \kappa$.

The comparisons between every two candidates are shown in Table 2.7.

	p	q	$x_j (j > i)$	$x_j (j < i)$
p	-	$2n - \kappa + 1$	$2n - \kappa + 2$	
q	$2n$	-	$3n - \kappa + 2$	
x_i	$2n - 1$	$n - 1$	$\times$	$\times$

Table 2.7: Comparisons between every two candidates in the $\mathcal{W}[1]$-hardness reductions for CCDV-Copeland$^\alpha$-UNI, CCDV-Copeland$^\alpha$-NON and DCDV-Copeland$^\alpha$-NON in Theorem 2.10. The comparisons between x_i and x_j cannot be exactly determined. However, their comparisons do not paly any role in the correctness argument.

Now we prove the correctness for CCDV-Copeland$^\alpha$-UNI.

($\Rightarrow$:) Suppose that $\mathcal{F}$ has an independent set S of size κ. We claim that we can make p the unique Copeland$^\alpha$ winner by deleting votes corresponding to S in Π_1. Let $\Pi_S = \{\pi_{I^1} \mid I \in S\}$ be the set of the votes corresponding to S in Π_1, and let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in Π_S. Since S is an independent set, we have that for every $x_i \in \Gamma$ there is at most one vote in Π_S which ranks p

above x_i. This implies that $N_{\mathcal{E}'}(p, x_i) \geq 2n - \kappa + 1$ for every $x_i \in \Gamma$, and hence, p beats every $x_i \in \Gamma$ in $\mathcal{E}'$. Moreover, since q is ranked above p in every vote in Π_S, $N_{\mathcal{E}'}(p, q) = 2n - \kappa + 1$. Therefore, p beats q in $\mathcal{E}'$. Summary all above, we know that p beats every other candidate in $\mathcal{E}'$; and thus, p is the unique Copeland$^\alpha$ winner (more precisely, p is the Condorcet winner in $\mathcal{E}'$).

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Let Π_S be a solution of $\mathcal{E}$, and $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in Π_S. Clearly, $\mathcal{E}'$ contains at least $4n - 2\kappa + 1$ votes. Since $N_{\mathcal{E}'}(q, x_i) \geq N_{\mathcal{E}}(q, x_i) - \kappa = 3n - 2\kappa + 2$ and $\kappa \leq n$, we know that q beats every candidate $x_i \in \Gamma$ in the final election. Since p is the unique winner in $\mathcal{E}'$, we know that Π_S contains no vote which ranks p above q (otherwise, q would also beat p, contradicting with the fact that p is the unique winner in $\mathcal{E}'$). Moreover, we know that p beats every candidate $x_i \in \Gamma$ in $\mathcal{E}'$. Since the final election contains at least $4n - 2\kappa + 1$ votes and $N_{\mathcal{E}}(p, x_i) = 2n - \kappa + 2$, p beats every $x_i \in \Gamma$ in the final election if there is at most one vote in Π_S which ranks p above x_i. Due to the fact, we have the following claim.

Claim. Π_S contains no vote in Π_2.

The correctness of the above claim follows from the proof of the Claim in the proof of Theorem 2.9. Due to the above claim, we know that $\Pi_S \subseteq \Pi_1$. Let S be the set of 2-intervals corresponding to Π_S. Since for every $x_i \in \Gamma$ there is at most one vote in Π_S which ranks p above x_i, there is no two 2-intervals in S which intersect, implying that $\mathcal{F}$ has an independent set of size κ. This finishes the proof for CCDV-Copeland$^\alpha$-UNI.

Now we argue why the same reduction applies to CCDV-Copeland$^\alpha$-NON. We have showed above that if there is an independent set of size κ, we can make p a (unique) winner. It remains to show the other direction. We begin with two observations. First, observe that we have to delete exactly κ votes to make p a winner, since otherwise, q would beat every other candidate. Second, observe that q beats every candidate $x_i \in \Gamma$ in the final election no matter which κ votes are deleted (this observation has been discussed above). Then, recall that every two 2-interval either do not intersect or they intersect at more than one point. Therefore, if we delete two votes which rank p above some candidate x_i, there must be another candidate $x_j \neq x_i$ which are ranked below p in both of the two votes. This implies that p beats every candidate in $x_i \in \Gamma$ in the final election (otherwise, p would be beaten by at least two candidates in Γ, contradicting with the fact that p is a winner in $\mathcal{E}'$). However, p beats every candidate in Γ only if there is an independent set of size κ for $\mathcal{F}$ as discussed above. This finishes the proof for CCDV-Copeland$^\alpha$-NON.

To check that the same reduction applies to DCDV-Copeland$^\alpha$-NON, observe first that no $x_i \in \Gamma$ can have a higher Copeland$^\alpha$ score than that of q in the final election—every x_i is beaten by q in the final election. Due to this, p is the only candidate which can prevent q from being a winner. This turns the problem into exactly CCDV-Copeland$^\alpha$-UNI. The argument for CCDV-Copeland$^\alpha$-UNI then works.

	p	q	q'	$x_j(j>i)$	$x_j(j<i)$
p	-	$2n-\kappa+1$	$2n-\kappa+1$	$2n-\kappa+2$	
q	$2n$	-	$4n-\kappa+1$	$3n-\kappa+2$	
q'	$2n$	0	-	$3n-\kappa+2$	
x_i	$2n-1$	$n-1$	$n-1$	$\times$	$\times$

Table 2.8: Comparisons between every two candidates in the $\mathcal{W}[1]$-hardness reduction for DCDV-Copeland$^\alpha$-UNI in Theorem 2.10. The comparisons between x_i and x_j cannot be exactly determined. However, their comparisons do not paly any role in the correctness argument.

Now we consider DCDV-Copeland$^\alpha$-UNI. The reduction is similar to the above one with the difference that we create one more dummy candidate q' which lies immediately on the right side of q in the 4-harmonious order. That is, the candidate set is $\Gamma \cup \{p, q, q'\}$ with q being the distinguished candidate, and the 4-harmonious order is $(q, q', \overrightarrow{\Gamma}, p)$. The role of the dummy candidate q' is to guarantee that, in the final election, every candidate in Γ is beaten by both q and q'; and thus, exclude the possibility that some x_i would have a higher score than that of q in the final election. To achieve this goal, we rank q' immediately after q in every vote and remains the order of other candidates unchanged. Precisely, we create the following votes.

$n-1 : (\mathcal{L}[x_1, p], q, q')$

$n-\kappa+2 : (p, \mathcal{L}[q, x_{|\Gamma|}])$

Besides, for every 2-interval $I_i = \{I_i^1 = [x_\alpha x_{,\beta}], I_i^2 = [x_\gamma, x_\delta]\}$ with $x_\beta < x_\gamma$, we create two votes as follows.

$(\mathcal{L}[q, q', x_{\alpha-1}], \mathcal{L}(x_\beta, x_\gamma), \mathcal{L}(x_\delta, p], \mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta]);$

$(q, q', \mathcal{L}[x_\alpha, x_\beta], \mathcal{L}[x_\gamma, x_\delta], p, \mathcal{L}[x_1, x_{\alpha-1}], \mathcal{L}(x_\beta, x_\gamma), \mathcal{L}(x_\delta, x_{|\Gamma|}]).$

The comparisons between every two candidates are shown in Table 2.8.

We have discussed that if there is an independent set of size κ, the candidate p can prevent q from being the unique winner by deleting κ votes. For the other direction, observe first that no candidate $x_i \in \Gamma$ can have a higher score than that of q since every x_i is beaten by both q and q' in the final election. Clearly, q' also cannot prevent q from being the unique winner since every vote ranks q above q'. Therefore, the only candidate which can prevent q from being the unique winner is p, and moreover, this happens only if p beats every candidate in Γ. The remaining argument is the same as for CCDV-Copeland$^\alpha$-UNI. $\square$

The last problems we consider are constructive control by deleting votes for Condorcet in 4-peaked elections. Recall that the constructive control by deleting votes for Condorcet is $\mathcal{NP}$-hard in general, while destructive control by deleting votes is polynomial-time solvable [157].

Theorem 2.11. *CCDV-Condorcet-UNI and CCDV-Condorcet-NON are $\mathcal{W}[1]$-hard in 4-peaked elections with respect to the number of deleted votes.*

Proof. The proof for CCDV-Condorcet-UNI is exactly the same as for CCDV-Copeland$^\alpha$-UNI, and the proof for CCDV-Condorcet-NON is exactly the same as for CCDV-Copeland$^\alpha$-UNI in Theorem 2.10. □

2.4 Conclusion

In this section, we have studied k-peaked elections which generalize the single-peaked elections by allowing at most k-peaks in each vote. We derived a dichotomy of the complexity of control problems for r-Approval voting in k-peaked elections with respect to k. Several of our results apply to approval voting and SP-AV as well. Furthermore, we have studied control problems for Maximin, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Condorcet from the parameterized complexity point of view. We proved that, except the destructive control by adding/deleting votes for Condorcet which is polynomial-time solvable in general, the constructive/destructive control by adding/deleting votes for all these three voting systems are $\mathcal{W}[1]$-hard in k-peaked elections with $k = 3, 4$, with respect to the number of added/deleted votes. In particular, control by adding votes turned out to be $\mathcal{W}[1]$-hard in 3-peaked elections, while control by deleting votes turned out to be $\mathcal{W}[1]$-hard in 4-peaked elections. All our results apply to both the unique-winner model and the nonunique-winner model. Our results are summarized in Tables 2.1 and 2.2.

Several challenging and intriguing questions remain open. Among them are the complexity of control by adding votes in 2-peaked elections and control by deleting votes in 2,3-peaked elections. See Table 2.2 for further details. It is well-known that determining whether an election is single-peaked is polynomial-time solvable [81, 103, 158]. However, we do not know whether the polynomial-time solvability holds in checking whether an election is 2-peaked elections. More generally, we do not know the complexity of checking whether an election is k-peaked with k being a constant.

3

Control in Elections with Bounded Single-Peaked Width

The concept of single-peaked width provides another prominent approach to generalize single-peaked elections. It can arise in the settings where the candidates are divided into groups, with each including the candidates which are similar each other. The similarity of the candidates in the same group leads to the fact that every voter ranks them together. Therefore, in these settings, preferences of voters over the candidates can be determined in two steps. First, voters present their preferences over the groups. Then, the voters present their preferences over all the candidates in every group. Elections with single-peaked with k are the elections where each group contains at most k candidates, and moreover, the preferences over the groups are single-peaked with respect to a harmonious order over the groups. In this chapter, we study control problems in elections with bounded single-peaked width.

3.1 Introduction

In this chapter, we mainly study control problems in elections with bounded single-peaked width. Intuitively, in an election with single-peaked width $\mathscr{k}$, the candidates can be grouped together, where the size of each group is bounded by $\mathscr{k}$, and for each group, every voter has the same preferences over all candidates in this group compared to candidates not in the group. Moreover, if considering each group as a candidate, the election is single-peaked. Clearly, single-peaked elections have a width equal to one. Cornaz, Galand and Spanjaard [70] first introduced single-peaked width into the complexity study of voting problems[i]. In particular, they considered a multi-winner determination problem (the proportional representation problem) and proved that this problem is $\mathcal{FPT}$ with respect to single-peaked width. Later, Cornaz, Galand and Spanjaard [71] showed that the Kemeny winner determination is $\mathcal{FPT}$ with respect to single-peaked width.

In this chapter, we study three concrete voting correspondences, namely, (weak) Condorcet, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$, and Maximin. Recall that in the general case, the following problems are all $\mathcal{NP}$-hard: the constructive control by adding/deleting votes for Condorcet [157], the constructive/destructive control by adding/deleting votes for Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ [112], and the constructive/destrutive control by adding/deleting votes for Maximin [110]. Our results are summarized as follows. Concerning the constructive control problems, we achieved $\mathcal{NP}$-hardness for Copeland$^\alpha$ with $0 \leq \alpha < 1$ even with single-peaked width $\mathscr{k} = 2$, while for Copeland^1 and Maximin, we show polynomial-time solvability with $\mathscr{k} = 2$ but $\mathcal{NP}$-hardness with $\mathscr{k} = 3$. In contrast, the constructive control problems for (weak) Condorcet turn out to be polynomial-time solvable for every fixed $\mathscr{k}$. More precisely, we prove that for (weak) Condorcet, the constructive control problems are $\mathcal{FPT}$ with respect to single-peaked width. In the destructive control case, both Copeland$^\alpha$ for all $0 \leq \alpha \leq 1$ and Maximin behave in the same way, that is, for both correspondences, the destructive control problems are $\mathcal{FPT}$ with respect to single-peaked width, implying polynomial-time solvability with every fixed $\mathscr{k}$. Note that the destructive control problems for (weak) Condorcet are polynomial-time solvable, even in general (i.e., with unbounded $\mathscr{k}$) [157]. Our results concerning the above problems are summarized in Table 3.1.

In addition to these concrete voting correspondences, we provide a general characterization for a broad class of voting correspondences to identify the ones for which the control problems are $\mathcal{FPT}$ with respect to single-peaked width. The considered class contains all correspondences passing the Smith-IIA criterion. The *Smith set* in

[i]Cornaz, Galand and Spanjaard defined in [70] the single-peaked width as $\mathscr{k} - 1$, the size of the maximum group minors one.

	Single-Peaked Width κ									Evidence
	$\kappa = 1$	$\kappa = 2$		$\kappa = 3$		κ: parameter		$\kappa = m$		
	for all	CC	DC	CC	DC	CC	DC	CC	DC	
		AV/DV	AV/DV	AV/DV	AV/DV	AV/DV	AV/DV	AV/DV	AV/DV	
Condorcet	$\mathcal{P}$					$\mathcal{FPT}$	$\mathcal{FPT}$ ($\mathcal{P}$)	$\mathcal{NP}$-h	$\mathcal{P}$	Theorem 3.1
Maximin				$\mathcal{NP}$-h		$\mathcal{NP}$-h	$\mathcal{FPT}$	$\mathcal{NP}$-h		Theorems 3.6, 3.7, 3.8
Copeland^1				$\mathcal{NP}$-h		$\mathcal{NP}$-h	$\mathcal{FPT}$	$\mathcal{NP}$-h		Theorems 3.3, 3.4, 3.5
Copeland$^\alpha$ $0 \le \alpha < 1$		$\mathcal{NP}$-h		$\mathcal{NP}$-h		$\mathcal{NP}$-h	$\mathcal{FPT}$	$\mathcal{NP}$-h		Theorems 3.2, 3.5

Table 3.1: A summary of results of control problems under Condorcet, Maximin and Copeland$^\alpha$ in elections with single-peaked width κ. Here, "$\mathcal{NP}$-h" stands for $\mathcal{NP}$-hard and "$\mathcal{P}$" stands for polynomial-time solvable. All results shown here for $\kappa = 2, 3$ and for κ being a parameter are our results. Moreover, all the results corresponding to the area in gray are polynomial-time solvability results. Note that $\kappa = 1$ is the single-peaked case and $\kappa = m$ is the general case, where m is the number of candidates. The polynomial-time solvability results in single-peaked elections (1-peaked elections) are from [44]. The polynomial-time solvability of the destructive control by adding/deleting votes for Condorcet is from [157]. The $\mathcal{NP}$-hardness results for $\kappa = m$ are from [110, 112, 157]. We remark that the definition of single-peaked width in [70] is equal to $\kappa - 1$.

an election is a subset S of candidates with minimum size, such that every candidate in S is preferred by more voters than every candidate outside S. Clearly, every election has a unique Smith set. A voting correspondence *passes the Smith-IIA criterion* ("IIA" stands for "independence of irrelevant alternatives"), if deleting any candidate outside the Smith set does not change the winners. Several voting correspondences have been found passing the Smith-IIA criterion, for instance, Ranked pairs, Schulze's, and Kemeny. The characterization considers elections with odd number of votes and states that, if a control problem for a correspondence in the above class is $\mathcal{FPT}$ with the number of candidates as parameter, then the same holds for the single-peaked width being the parameter. This characterization applies to both constructive and destructive cases. We remark that all our results in this chapter apply to both the unique-winner and the nonunique-winner models.

The following definitions and notations are essential for presenting this chapter.

A voting correspondence is said to be *weakCondorcet-consistent*, if on every input that has at least one weak Condorcet winner, the winners, according to the voting correspondence, are exactly the set of weak Condorcet winners [44].

Single-Peaked Width. A subset $C \subseteq \mathcal{C}$ is called an *interval*[ii] if all candidates in C are ranked contiguously in every vote. For example, for the election with candidates $\{a, b, c, d, e\}$ and votes with preferences $\{a \succ_1 b \succ_1 c \succ_1 d \succ_1 e,\ d \succ_2 c \succ_2 b \succ_2 e \succ_2 a,\ a \succ_3 e \succ_3 b \succ_3 d \succ_3 c\}$, $\{b, c, d\}$ is an interval. *Contracting* an interval C is the operation that first adds a new candidate c' to the election such that $C \cup \{c'\}$ forms a new interval and the preference between any two candidates of $\mathcal{C}$ in each vote preserves the same as before, and then deletes all candidates in C. For example, after contracting the interval $\{b, c, d\}$ in the above example, we get the new election with candidates a, c', e and votes with preferences $\{a \succ_1 c' \succ_1 e,\ c' \succ_2 e \succ_2 a,\ a \succ_3 e \succ_3 c'\}$, where c' is the newly introduced candidate. Intuitively, contracting is to assign a new candidate to an interval which can represent the interval properly in the sense that the information of the preference between every candidate in the interval and every candidate outside the interval is preserved.

Let $P = (C_1, C_2, ..., C_\omega)$ be an ordered partition of $\mathcal{C}$ with each C_i being an interval. We say P is a *single-peaked partition* if contracting all intervals in P results in a single-peaked election with the harmonious order $(c_1, c_2, ..., c_\omega)$, where each c_i is the new candidate introduced for the interval C_i. We say a vote *has its peak at C_i with respect to P* if the interval C_i is ranked above every other interval in the vote.

[ii]The term "interval" used in this chapter is different from that of previous chapter. The term "interval" used in this chapter follows from the latest paper concerning single-peaked width by Cornaz, Galand and Spanjaard [71] who first introduced the concept of single-peaked width in the context of computational social choice. In an earlier paper by the same authors [70], they also implicitly used the term "cluster". The term "interval" here is also equal to "clone set" studied by Tideman [235]. Besides, it is also related to the notion of component on profile studied by Laffond [173].

The *width* of P is defined as $\max_{1 \le i \le \omega}\{|C_i|\}$. The *single-peaked width* of an election is the minimum width among all its single-peaked partitions.

Median Group: Let $P = (C_1, C_2, ..., C_\omega)$ be a single-peaked partition of the election $(\mathcal{C}, \Pi_\mathcal{V})$, and let $(\pi_1, \pi_2, ..., \pi_n)$ be an order of $\Pi_\mathcal{V}$ such that for $i < j$ the peak of π_i does not lie on the right-side of the peak of π_j in P. The set of all intervals lying between the peak C_i of $\pi_{\lceil n/2 \rceil}$ and the peak C_j of $\pi_{\lfloor n/2+1 \rfloor}$, together with C_i and C_j, denoted by $\mathcal{G}[C_i, C_j]$, is called the *median group*. Furthermore, C_i is called the *left boundary* of the median group and C_j is the *right boundary* of the median group. If there is only one interval in the median group, we call it a *median interval*. See Figure 3.1 for an example.

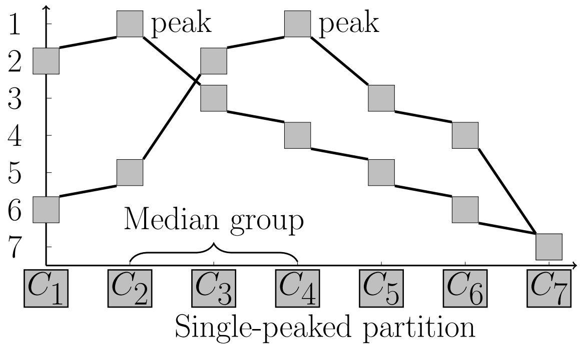


Figure 3.1: An illustration of median group. There are two votes, where the first vote has preference $C_2 \succ C_1 \succ C_3, ..., \succ C_7$ over the intervals, and the second vote has the preference $C_4 \succ C_3 \succ C_5 \succ C_6 \succ C_2 \succ C_1 \succ C_7$. The peak C_2 of the first vote is on the left side of the peak C_4 of the second vote.

This chapter studies control by adding/deleting votes for Condorcet, Maximin and Copeland$^\alpha$ for every $0 \le \alpha \le 1$ as in Chapter 2, but with the input elections having bounded single-peaked width. Moreover, we assume that optimal single-peaked partitions is given alone with the input elections. This assumption is sound since searching for an optimal single-peaked partition can be done in polynomial time [71]. In addition, we do not create new votes throughout handling the problems (we only add/delete votes/canddiates which are given in advance). We remark that in the (constructive/destructive) control by adding votes, the single-peaked partition is based on the registered votes union the unregistered votes. Therefore, both the registered votes and the unregistered votes have single-peaked width at most k with respect to the given single-peaked partition.

All $\mathcal{NP}$-hardness reductions in this paper are from the following $\mathcal{NP}$-hard problem [131].

Exact 3 Set Cover (X3C)
Input: A universal set $U = \{c_1, c_2, ..., c_{3\kappa}\}$ and a collection S of 3-subsets of U.
Question: Is there an $S' \subseteq S$ such that $|S'| = \kappa$ and each $c_i \in U$ appears in exactly one set of S'?

3.2 Condorcet and Weak Condorcet Control

The constructive control by adding/deleting votes for (weak) Condorcet is $\mathcal{NP}$-hard in the general case [157] but turned out to be polynomial-time solvable when restricted to single-peaked elections [44]. On the other hand, the destructive control by adding/deleting votes is polynomial-time solvable even in the general case [157]. In this section, we study constructive control by adding/deleting votes in Condorcet and weak Condorcet, restricted to elections with bounded single-peaked width. We prove that both problems are polynomial-time solvable if the single-peaked width is a constant. From the perspective of the parameterized complexity, our results indeed show that these problems are $\mathcal{FPT}$. The following observations are useful.

Observation 3.1. *Every two candidates from different intervals in the median group are tied.*

Proof. Let $(C_1, C_2, ..., C_\omega)$ be the single-peaked partition and $\mathcal{G}[C_l, C_r]$ be the median group. Let C_i and C_j be two arbitrary intervals in $\mathcal{G}[C_l, C_r]$ with $i < j$, and $c \in C_i, c' \in C_j$ be two candidates. Due to the definition of median group, all votes with peaks at C_l or on the left-side of C_l (let $\Pi^l_{\mathcal{V}}$ denote the multiset of these votes) prefer c to c', and all votes with peaks at C_r or on the right-side of C_r (let $\Pi^r_{\mathcal{V}}$ denote the multiset of these votes) prefer c' to c. Moreover, the size of $\Pi^l_{\mathcal{V}}$ is equal to the size of $\Pi^r_{\mathcal{V}}$. Therefore, c ties c'. □

Observation 3.2. *Every weak Condorcet winner is from the median group.*

Proof. This observation is correct since every candidate which is not in the median group is beaten by at least one candidate in the median group. More precisely, suppose that c is a candidate contained in an interval lying on the right-side (resp. left-side) of the median group, then every candidate in C_r (resp. C_l) beats c, where C_l and C_r are the left boundary and the right boundary of the median group, respectively. □

Observation 3.3. *If an election $\mathcal{E}$ has a Condorcet winner, then the median group contains exactly one interval.*

Proof. Suppose that the median group $\mathcal{G}$ contains more than one interval. Due to Observation 3.1, every candidate in the median group ties at least one candidate in a different interval in the median group, and thus, the Condorcet winner cannot exist. □

In the following, "modifiable" votes refer to the registered votes in the case of control by deleting votes, and refer to the unregistered votes in the case of control by adding votes. For two subsets of candidates C and C' with $C \subseteq C'$, we say two votes with preferences $\succ_1$ *and* $\succ_2$*, respectively, are consistent with respect to* C and C' if they have the same preference over all candidates in C, and for every two candidates $c \in C$ and $c' \in C' \setminus C$, $c \succ_1 c'$ if and only if $c \succ_2 c'$.

Theorem 3.1. *CCAV-Condorcet-UNI, CCAV-Condorcet-NON, CCDV-Condorcet-UNI and CCDV-NON are $\mathcal{FPT}$ with respect to single-peaked width.*

Proof. We first consider CCAV-Condorcet-UNI. Let $\Pi_{\mathcal{V}_1}$ be the multiset of registered votes and $\Pi_{\mathcal{V}_2}$ be the multiset of the unregistered votes. Let C_p be the interval containing the distinguished candidate p. Let $\mathcal{k}$ be the single-peaked width of the given election. Due to Observation 3.3, to make p the Condorcet winner we need to make the interval C_p the median interval and to make p beat all the other candidates in C_p. To this end, we first divide the modifiable votes (in this case the modifiable votes are unregistered votes) $\Pi_{\mathcal{V}_2}$ into three multisets: X containing the votes with peaks on the left-side of C_p with respect to the single-peaked partition, Y the votes with peaks on the right-side of C_p, and Z the votes with peaks at C_p. Then, we further divide each of these three multisets into at most $2^{\mathcal{k}-1}$ submultisets, each containing the votes which are pairwise consistent with respect to $\{p\}$ and C_p. By assigning to each subset a variable (indicating how many votes from this subset are in the solution), the election instance is reduced to an ILP instance which can be solved in $\mathcal{FPT}$ time based on Lenstra's theorem [177]. See Section 1.3.3 for a detailed discussion of Lenstra's theorem [177].

Let $\bar{x}, \bar{y}$ and $\bar{z}$ be the numbers of votes in $\Pi_{\mathcal{V}_1}$ with peaks on the left-side of C_p with respect to the single-peaked partition, with peaks on the right-side of C_p, and with peaks at C_p, respectively. We will use x_β, y_β and z_β to denote the variables assigned to the subsets of X, Y and Z, respectively, where β is a subset of $C_p \setminus \{p\}$. Here, for each β, x_β (y_β, z_β) is assigned to the submultiset of X (Y, Z), which contains votes ranking every candidate of β above p and ranking every candidate not in β below p. Firstly, the ILP instance has the following constraints:

$$(1)\ \bar{x} + \sum_\beta x_\beta < \bar{y} + \bar{z} + \sum_\beta y_\beta + \sum_\beta z_\beta$$

$$(2)\ \bar{y} + \sum_\beta y_\beta < \bar{x} + \bar{z} + \sum_\beta x_\beta + \sum_\beta z_\beta$$

$$(3)\ \sum_\beta (x_\beta + y_\beta + z_\beta) \le \mathcal{R}$$

Here, (1) and (2) together are to ensure that C_p is the unique interval in the median group. In particular, (1) implies that there are less than half votes having their peaks on the left side of C_p in the final election, and (2) implies that there are less than half votes having their peaks on the right side of C_p in the final election. Moreover, (3) states that at most $\mathcal{R}$ votes are added. Then, for every $c \in C_p \setminus \{p\}$, there is a constraint:

$$N(p,c) + \sum_{c \notin \beta}(x_\beta + y_\beta + z_\beta) - N(c,p) - \sum_{c \in \beta}(x_\beta + y_\beta + z_\beta) > 0$$

where $N(.)$ is based on the registered votes $\Pi_{\mathcal{V}_1}$.

These inequalities ensure that p beats every candidate in $C_p \setminus \{p\}$. Since we formulate the control problems as decision problems, there is no optimization function in the ILP.

Now we consider CCAV-Condorcet-NON, that is, the problem to determine whether we can make the distinguished candidate a weak Condorcet winner by adding limited votes. Due to Observations 3.1 and 3.2, to make the distinguished candidate p a weak Condorcet winner, we have to make the interval C_p be included in the median group and to make p the weak Condorcet winner among the candidates in C_p. Therefore, we can use similar ILP technique as for CCAV-Condorcet-UNI to solve this problem. Precisely, the constraints for CCAV-Condorcet-NON are the same as that for CCAV-Condorcet-UNI with only the difference that the last constraint is replaced by the following one.

$$N(p,c) + \sum_{c \notin \beta}(x_\beta + y_\beta + z_\beta) - N(c,p) - \sum_{c \in \beta}(x_\beta + y_\beta + z_\beta) \geq 0$$

Now we consider the control by deleting votes. We first consider CCDV-Condorcet-UNI. Let $\mathcal{E}$ be the given election. The modifiable votes are divided and assigned with variables in the same way as discussed in the case of CCAV-Condorcet-UNI. Let $\bar{x}, \bar{y}$ and $\bar{z}$ be the numbers of votes in $\mathcal{E}$ with peaks on the left-side of C_p with respect to the single-peaked partition, with peaks on the right-side of C_p, and with peaks at C_p, respectively. The constraints are as follows.

(1) $\bar{x} + \sum_\beta x_\beta < \bar{y} + \bar{z} + \sum_\beta y_\beta + \sum_\beta z_\beta$

(2) $\bar{y} + \sum_\beta y_\beta < \bar{x} + \bar{z} + \sum_\beta x_\beta + \sum_\beta z_\beta$

(3) $\sum_\beta (x_\beta + y_\beta + z_\beta) \leq \mathcal{R}$

(4) For every $c \in C_p \setminus \{p\}$,

$$N_{\mathcal{E}}(p,c) - \sum_{c \notin \beta}(x_\beta + y_\beta + z_\beta) - N_{\mathcal{E}}(c,p) + \sum_{c \in \beta}(x_\beta + y_\beta + z_\beta) > 0$$

Finally, we come to CCDV-Condorcet-NON. We use the similar method here to solve the problem. Precisely, the constraints for CCDV-Condorcet-NON are the same as that for CCDV-Condorcet-UNI with only the difference that the last constraint is replaced by the following one.

$$N_{\mathcal{E}}(p, c) - \sum_{c \notin \beta}(x_\beta + y_\beta + z_\beta) - N_{\mathcal{E}}(c, p) + \sum_{c \in \beta}(x_\beta + y_\beta + z_\beta) \geq 0$$

$\square$

Due to Theorem 3.1, we can directly get the following result for the Young winner determination problem which is $\mathcal{P}_{||}^{\mathcal{NP}}$-complete in general [221]. In an Young election, each candidate c has a *Young score* defined as the minimum number of votes to be deleted to make c the Condorcet winner. A *Young winner* is a candidate with the least Young score. The Young winner determination problem can be reduced to the problem of deciding whether a distinguished candidate can be made a Condorcet winner by deleting $\mathcal{R}$ votes, equivalent to the problem control by deleting votes for Condorcet.

Corollary 3.1. *Young winner determination is $\mathcal{FPT}$ with respect to single-peaked width.*

3.3 Copeland Control

In this section, we study control problems for Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. Our results are summarized in Table 3.2. In particular, we prove that both the constructive control by adding votes and the constructive control by deleting votes are $\mathcal{NP}$-hard for Copeland$^\alpha$ for every $0 \leq \alpha < 1$ but polynomial-time solvable for Copeland^1, when restricted to elections with single-peaked width 2. Moreover, we prove that the same problems become $\mathcal{NP}$-hard for Copeland^1 when restricted to elections with single-peaked width 3. In the contrast, the destructive control by adding/deleting votes for Copeland$^\alpha$ for all $0 \leq \alpha \leq 1$ turns out to be $\mathcal{FPT}$. Recall that the problems constructive/destructive control by adding/deleting votes are all $\mathcal{NP}$-hard for Copeland$^\alpha$ for all $0 \leq \alpha \leq 1$ [112].

Theorem 3.2. *CCAV-Copeland$^\alpha$-UNI, CCAV-Copeland$^\alpha$-NON, CCDV-Copeland$^\alpha$-UNI, CCDV-Copeland$^\alpha$-NON are $\mathcal{NP}$-hard in elections with single-peaked with 2, for every $0 \leq \alpha \leq 1$.*

Proof. We first consider CCAV-Copeland$^\alpha$-UNI. Let $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S)$ be an instance of X3C. We construct an instance $\mathcal{E}$ for CCAV-Copeland$^\alpha$-UNI in elections with single-peaked width 2 as follows.

<table>
<tr><td rowspan="2"></td><td colspan="3">Single-peaked width $\mathcal{k}$</td></tr>
<tr><td>$\mathcal{k} = 2$</td><td>$\mathcal{k} = 3$</td><td>$\mathcal{k}$: parameter</td></tr>
<tr><td>CCAV</td><td rowspan="2">$\mathcal{NP}$-hard: $0 \leq \alpha < 1$
$\mathcal{P}$: $\alpha = 1$</td><td colspan="2" rowspan="2">$\mathcal{NP}$-hard: $0 \leq \alpha \leq 1$</td></tr>
<tr><td>CCDV</td></tr>
<tr><td>DCAV</td><td colspan="2" rowspan="2">$\mathcal{P}$</td><td rowspan="2">$\mathcal{FPT}$</td></tr>
<tr><td>DCDV</td></tr>
</table>

Table 3.2: Complexity of constructive/destructive control by adding/deleting votes in Copeland$^\alpha$. Here, "$\mathcal{P}$" stands for polynomial-time solvable. All results apply to both the unique-winner and the nonunique-winner models.

Candidates: There are in total $6\kappa + 2$ candidates. More specifically, for each $c_x \in U$ we create two corresponding candidates c'_x and c''_x which form an interval denoted by $I(c_x)$ in the election. In addition, we have two candidates p and p' which form an interval $I(p)$. The distinguished candidate is p.

Single-Peaked Partition: $(I(p), I(c_1), I(c_2), ..., I(c_{3\kappa}))$.

Registered Votes: There are $\kappa - 1$ registered votes defined as $c'_{3\kappa} \succ c''_{3\kappa} \succ c'_{3\kappa-1} \succ c''_{3\kappa-1} \succ, ..., \succ p' \succ p$. In addition, there is one vote defined as $c''_{3\kappa} \succ c'_{3\kappa} \succ c''_{3\kappa-1} \succ c'_{3\kappa-1} \succ, ..., \succ p \succ p'$. Clearly, with the registered votes, p has Copeland$^\alpha$ score 0, p' has Copeland$^\alpha$ score 1, each c'_x has Copeland$^\alpha$ score $2x + 1$, and each c''_x has Copeland$^\alpha$ score $2x$.

Unregistered Votes: The unregistered votes are created according to S. More precisely, for each $s = \{c_i, c_j, c_k\} \in S$, we create a vote π_s with preference $\succ_s$ as follows. The peak of the vote π_v is at the interval $I(p)$ and $p \succ_s p'$. For every two candidates $a \in I(c_x)$ and $b \in I(c_y)$ with $x < y$, we have $a \succ_s b$. Finally, in each interval $I(c_x)$, we set $c'_x \succ_s c''_x$ if $x \in \{i, j, k\}$ and $c''_x \succ_s c'_x$ otherwise.

Number of Added Votes: $\mathcal{R} = \kappa$.

In the following, we show that $\mathcal{F}$ has an exact 3-set cover if and only if we can add at most $\mathcal{R} = \kappa$ unregistered votes to make p the unique winner.

($\Rightarrow$:) Let S' be an exact 3-set cover of $\mathcal{F}$. We claim that adding all votes corresponding to S', that is, the votes $\Pi_{\mathcal{V}'} = \{\pi_s \mid s \in S'\}$ (π_v has preference $\succ_s$), will make p the unique winner. Let $\mathcal{E}'$ be the election obtained from $\mathcal{E}$ by adding all the votes in $\Pi_{\mathcal{V}'}$ to the registered votes. It is clear that p beats p' in $\mathcal{E}'$. Since there are exactly κ votes with peaks at $I(p)$ and exactly κ votes with peaks at $I(c_{3\kappa})$ in $\mathcal{E}'$, every two candidates which are in different intervals are tied. Therefore, p has Copeland$^\alpha$ score $6\alpha \cdot \kappa + 1$ and p' has Copeland$^\alpha$ score $6\alpha \cdot \kappa$ in $\mathcal{E}'$. We now analyze the Copeland$^\alpha$ score of other candidates in $\mathcal{E}'$. Let c'_x and c''_x be the two candidates in an interval

$I(c_x)$ with $1 \le x \le 3\kappa$. Since S' is an exact 3-set cover, due to the construction, there is exactly one vote in $\Pi_{\mathcal{V}'}$ which prefers c'_x to c''_x; thus there are exactly $\kappa - 1$ votes in $\Pi_{\mathcal{V}'}$ which prefer c''_x to c'_x. Together with the registered votes, c'_x ties c''_x in the final election $\mathcal{E}'$. Since each of c'_x and c''_x ties all other candidates as stated above, the final Copeland$^\alpha$ score of c'_x and c''_x are both $\alpha \cdot (6\kappa + 1)$. Since $\alpha < 1$, p is the unique winner in $\mathcal{E}'$.

($\Leftarrow$:) Let $\Pi_{\mathcal{V}'}$ be a solution of $\mathcal{E}$ and S' be the subset of S corresponding to $\Pi_{\mathcal{V}'}$, that is, $S' = \{s \mid \pi_s \in \Pi_{\mathcal{V}'}\}$. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all votes in $\Pi_{\mathcal{V}'}$ to the registered votes. It is easy to see that $\Pi_{\mathcal{V}'}$ contains exactly κ votes, since otherwise, one of $c'_{3\kappa}$ and $c''_{3\kappa}$ would beat all the other candidates and thus be a winner in $\mathcal{E}'$. Moreover, since all unregistered votes have their peaks at $I(p)$, every two candidates from different intervals are tied in the final election $\mathcal{E}'$. Since all unregistered votes prefer p to p', the Copeland$^\alpha$ score of p is $6\alpha \cdot \kappa + 1$ in $\mathcal{E}'$. Since p is the unique winner in $\mathcal{E}'$, c'_x ties c''_x for all $1 \le x \le 3\kappa$ (otherwise, at least one of c'_x and c''_x would have a Copeland$^\alpha$ score $6\alpha \cdot \kappa + 1$, contradicting that p is the unique winner in $\mathcal{E}'$). Then, according to the construction, for each c_x there is exactly one vote in $\Pi_{\mathcal{V}'}$ preferring c'_x to c''_x. This implies that S' contains exactly one subset containing c_x; thus, S' forms an exact 3-set cover of $\mathcal{F}$.

The $\mathcal{NP}$-hardness proof for CCAV-Copeland$^\alpha$-NON can be derived from the above reduction by deleting the candidate p' in $\mathcal{E}$.

Now, we consider CCDV-Copeland$^\alpha$-UNI. Let $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S)$ be an instance of X3C. For each $c \in U$, let $o(c)$ be the number of sets in S which contain c, and let $\bar{o}(c)$ be the number of sets in S which do not contain c. We assume that $\bar{o}(c) \ge \kappa - 1$ for all $c \in U$. This assumption does not change the complexity of X3C, since any instance which does not satisfy the requirement must be a no-instance. We construct an instance $\mathcal{E}$ for CCDV-Copeland$^\alpha$-UNI restricted to elections with single-peaked width 2 as follows. The candidate set and the single-peaked partition are the same as for CCAV-Copeland$^\alpha$-UNI.

Votes: There are in total $2|S| - \kappa$ votes with $|S| - \kappa$ votes having peaks at $I(p)$ and all the other $|S|$ votes (corresponding to S) having peaks at $I(c_{3\kappa})$. The central idea is to construct the votes such that all deleted votes are from the ones with peaks at $I(c_{3\kappa})$ whenever $\mathcal{E}$ is a yes-instance. Furthermore, after deleting these votes, every two candidates c'_x and c''_x in $I(c_x)$ are tied ($x = 1, 2, ..., 3\kappa$). Note that deleting one vote with peaks at $I(p)$ will make one of $c'_{3\kappa}$ and $c''_{3\kappa}$ a winner. Recall that, if more than half of the votes have peaks at $I(c_{3\kappa})$, then each of $c'_{3\kappa}$ and $c''_{3\kappa}$ beats all other candidates. Thus, we delete only votes with peaks at $I(c_{3\kappa})$.

We first create votes corresponding to S. For each $s = (c_i, c_j, c_k) \in S$, we create a vote π_s with preference $\succ_s$. The peak of the vote π_s is at $I(c_{3\kappa})$ and the preference of the vote between p and p' is $p \succ_s p'$. Thus, for every two candidates $a \in I(c_x)$ and $b \in I(c_y)$ with $x < y$, we have that $b \succ_s a$. Moreover, for each $I(c_x)$ with $1 \le x \le 3\kappa$, we set $c'_x \succ_s c''_x$ if $x \in \{i, j, k\}$ and $c''_x \succ_s c'_x$ otherwise. Thus, there are $|o(c_x)|$ votes with preferences $c'_x \succ c''_x$ and $|\bar{o}(c_x)|$ votes with preferences $c''_x \succ c'_x$ now.

We now construct the votes with peaks at $I(p)$. There are in total $|S| - \kappa$ such votes each preferring p to p'. Our goal then is to create the votes such that for each $I(c_x)$ with $1 \leq x \leq 3\kappa$, there are $|o(c_x)| - 1$ votes with preferences $c''_x \succ c'_x$ and $\bar{o}(c_x) - \kappa + 1$ votes with preferences $c'_x \succ c''_x$. To this end, we do the following: for each c_x with $1 \leq x \leq 3\kappa$, we set $c'_x \succ c''_x$ in arbitrary $|o(c_x)| - 1$ votes, and in all others (in total $|S| - \kappa - |o(c_x)| + 1 = |\bar{o}(c_x)| - \kappa + 1$) we set $c''_x \succ c'_x$. Since all these votes have their peaks at $I(p)$, the preference between every two candidates $a \in I(c_x)$ and $b \in I(c_y)$ with $x < y$ is $a \succ b$.

Number of Deleted Votes: $\mathcal{R} = \kappa$.

In the following, we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{E}$ is a yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. We claim that deleting all votes corresponding to S', that is, all votes in $\Pi_{\mathcal{V}'} = \{\pi_s \mid s \in S'\}$ makes p the unique winner. Since after deleting all votes in $\Pi_{\mathcal{V}'}$, the number of votes with peaks at $I(p)$ is equal to that of votes with peaks at $I(c_{3\kappa})$, and there is no vote with peak between $I(p)$ and $I(c_{3\kappa})$, every two candidates from different intervals are tied. Moreover, for each c'_x with $1 \leq x \leq 3\kappa$, there are exactly $|\bar{o}(c_x)| - \kappa + 1 + |o(c_x)| - 1 = |S| - \kappa$ votes (exactly half of the remaining votes) preferring c'_x to c''_x after deleting all votes in $\Pi_{\mathcal{V}'}$. Thus, c'_x ties c''_x for all $1 \leq x \leq 3\kappa$. Thus, every candidate except p has a Copeland$^\alpha$ score $\alpha \cdot (6\kappa + 1)$. Since p is preferred to p' by all votes, the Copeland$^\alpha$ score of p is $6\alpha \cdot \kappa + 1$. Since $\alpha < 1$, p becomes the unique winner.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and $\Pi_{\mathcal{V}'}$ is a solution. Let $S' = \{s \mid \pi_s \in \Pi_{\mathcal{V}'}\}$. We claim that S' is an exact 3-set cover for $\mathcal{F}$. Clearly, $\Pi_{\mathcal{V}'}$ contains exactly κ votes and all have peaks at $I(c_{3\kappa})$, since otherwise, there will be more votes with peaks at $I(c_{3\kappa})$ than these votes with peaks at $I(p)$, resulting in one of $c'_{3\kappa}$ and $c''_{3\kappa}$ being a winner. Therefore, after deleting all votes in $\Pi_{\mathcal{V}'}$, every two candidates from two different intervals are tied. Moreover, since p becomes the unique winner in the final election, c'_x must tie c''_x for all $1 \leq x \leq 3\kappa$. Therefore, for each c_x with $1 \leq x \leq 3\kappa$, there is exactly one vote in $\Pi_{\mathcal{V}'}$ having the preference $c'_x \succ c''_x$. This vote corresponds to an $s \in S'$ containing c_x, implying that S' must be an exact 3-set cover of $\mathcal{F}$.

The proof for CCDV-Copeland$^\alpha$-NON can be derived from the above reduction by deleting the candidate p'. □

In the following, we study the control problems for Copeland^1. We first consider elections with single-peaked width 2. Observe that every election with single-peaked width 2 contains at least one weak Condorcet winner. More precisely, each interval in the median group contains at least one weak Condorcet winner. Note that every candidate in the median group beats or ties every candidate not in the median group. Furthermore, since Copeland^1 is weakCondorcet-consistent and the problems constructive control by adding/deleting votes are polynomial-time solvable for (weak) Condorcet when restricted to elections with single-peaked width 2, as implied by Theorem 3.1, the problems constructive control by adding/deleting votes for Copeland^1 are polynomial-time solvable in elections with single-peaked width 2. This result is summarized in Theorem 3.3. We remark that Copeland$^\alpha$ for every $0 \leq \alpha < 1$ is not weakCondorcet-consistent even when restricted to single-peaked elections [44], and thus, the following theorem does not apply to $0 \leq \alpha < 1$.

Theorem 3.3. *CCAV-Copeland^1-UNI, CCAV-Copeland^1-NON, CCDV-Copeland^1-UNI and CCDV-Copeland^1-NON are polynomial-time solvable in elections with single-peaked width 2.*

Now we consider the problems restricted to elections with single-peaked width 3. In contrast to the polynomial-time solvability as stated in Theorem 3.3, we show that the constructive control problems become $\mathcal{NP}$-hard in elections with single-peaked width 3. We remark that, even though Copeland^1 is weakCondorcet-consistent, the argument for Theorem 3.3 does not hold in this case since there may not be a weak Condorcet winner in elections with single-peaked width 3.

Theorem 3.4. *CCAV-Copeland^1-UNI, CCAV-Copeland^1-NON, CCDV-Copeland^1-UNI and CCDV-Copeland^1-NON are $\mathcal{NP}$-hard in elections with single-peaked width 3.*

Proof. We prove the theorem by reductions from X3C. We start with the reduction for CCAV-Copeland^1-UNI.

Let $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S)$ be an instance of X3C. We assume that $\kappa \equiv 0 \bmod 6$. This assumption does not change the hardness of X3C: if $\kappa \not\equiv 0 \bmod 6$, we can add some dummy elements to U, and add some 3-subsets to S which form an exact 3-set cover of the dummy elements. We construct an instance $\mathcal{E}$ for CCAV-Copeland^1-UNI restricted to elections with single-peaked width 3 as follows.

Candidates: There are $9\kappa + 3$ candidates in total. More specifically, for each $c_x \in U$ we create three candidates c_x^1, c_x^2 and c_x^3 which form an interval denoted by $I(c_x)$. In addition, we have three candidates p, p' and p'' which form an interval $I(p)$. The distinguished candidate is p.

Single-Peaked Partition: $(I(p), I(c_1), I(c_2), ..., I(c_{3\kappa}))$.

Registered Votes: There are $\frac{7}{3}\kappa$ registered votes. In particular, we have

(1) $\frac{5}{6}\kappa$ votes with preference

$$c_{3\kappa}^1 \succ c_{3\kappa}^2 \succ c_{3\kappa}^3 \succ c_{3\kappa-1}^1 \succ c_{3\kappa-1}^2 \succ c_{3\kappa-1}^3 \succ, ..., \succ p \succ p' \succ p''$$

(2) $\frac{5}{6}\kappa$ votes with preference

$$c_{3\kappa}^2 \succ c_{3\kappa}^3 \succ c_{3\kappa}^1 \succ c_{3\kappa-1}^2 \succ c_{3\kappa-1}^3 \succ c_{3\kappa-1}^1 \succ, ..., \succ p \succ p' \succ p''$$

(3) $\frac{2}{3}\kappa$ votes with preference

$$p \succ p' \succ p'' \succ c_1^3 \succ c_1^1 \succ c_1^2 \succ, ..., \succ c_{3\kappa}^3 \succ c_{3\kappa}^1 \succ c_{3\kappa}^2$$

Clearly, with the registered votes, p has Copeland^1 score 2, p' has Copeland^1 score 1, p'' has Copeland^1 score 0, and each c_x^γ with $1 \leq x \leq 3\kappa$ and $\gamma = 1, 2, 3$ has Copeland^1 score $3x + 1$.

Unregistered Votes: We create the unregistered votes according to S. Precisely, for each $s = \{c_i, c_j, c_k\} \in S$, we create a vote π_s with preference $\succ_s$ as follows: the peak of the vote is at $I(p)$ and $p \succ_s p' \succ_s p''$. For every two candidates $a \in I(c_x)$ and $b \in I(c_y)$ with $x < y$, we have $a \succ_s b$. Finally, in each interval $I(c_x)$, we set $c_x^2 \succ_s c_x^3 \succ_s c_x^1$ if $x \in \{i, j, k\}$ and set $c_x^3 \succ_s c_x^1 \succ_s c_x^2$ otherwise.

Number of Added Votes: $\mathcal{R} = \kappa$.

In the following, we show that $\mathcal{F}$ has an exact 3-set cover if and only if we can add at most κ unregistered votes to make p the unique winner.

($\Rightarrow$:) Let S' be an exact 3-set cover of $\mathcal{F}$. We claim that adding all unregistered votes corresponding to S', that is, the votes in $\Pi_{\mathcal{V}'} = \{\pi_s \mid s \in S'\}$, will make p the unique winner. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all votes in $\Pi_{\mathcal{V}'}$ to the registered votes. It is clear that p beats p' and p'' in $\mathcal{E}'$. Since there are exactly $\frac{5}{3}\kappa$ votes with peaks at $I(p)$ and exactly $\frac{5}{3}\kappa$ votes with peaks at $I(c_{3\kappa})$ in $\mathcal{E}'$, every two candidates from different intervals are tied. Thus, p has Copeland^1 score $9\alpha \cdot \kappa + 2$, p' has Copeland^1 score $9\alpha \cdot \kappa + 1$ and p'' has Copeland^1 score $9\alpha \cdot \kappa$ in $\mathcal{E}'$. We now analyze the Copeland^1 scores of other candidates in $\mathcal{E}'$. Due to the construction and the fact that S' is an exact 3-set cover, for each c_x there is exactly one vote in $\Pi_{\mathcal{V}'}$ with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and exactly $\kappa - 1$ votes in $\Pi_{\mathcal{V}'}$ with preference $c_x^3 \succ c_x^1 \succ c_x^2$. Together with the registered votes, where there are $\frac{5}{6}\kappa$ votes with preference $c_x^1 \succ c_x^2 \succ c_x^3$, $\frac{5}{6}\kappa$ votes with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and $\frac{2}{3}\kappa$ votes with preference $c_x^3 \succ c_3^1 \succ c_x^2$, we know that for each c_x, c_x^1 beats c_x^2, c_x^2 beats c_x^3 and c_x^3 beats c_x^1 in $\mathcal{E}'$. As discussed above, each of c_x^1, c_x^2 and c_x^3 ties any other candidate in $\mathcal{E}'$, thus, the Copeland^1 scores of c_x^1, c_x^2 and c_x^3 are all $9\alpha \cdot \kappa + 1$, for every $1 \leq x \leq 3\kappa$. It is clear now that p becomes the unique winner in $\mathcal{E}'$.

($\Leftarrow$:) Let $\Pi_{\mathcal{V}'}$ be a solution of $\mathcal{E}$ and S' be the subset of S corresponding to $\Pi_{\mathcal{V}'}$, that is, $S' = \{s \mid \pi_s \in \Pi_{\mathcal{V}'}\}$. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all votes in $\Pi_{\mathcal{V}'}$ to the registered votes. It is easy to see that $\Pi_{\mathcal{V}'}$ contains exactly κ votes, since otherwise, each candidate in $I(c_{3\kappa})$ would beat every other candidate not in $I(c_{3\kappa})$, and thus at least one of them is a winner in $\mathcal{E}'$. Moreover, since all unregistered votes have their peaks at $I(p)$, every two candidates from different intervals are tied in the final election $\mathcal{E}'$. Therefore, p has Copeland^1 score $9\alpha \cdot \kappa + 2$. Since p is the unique winner in the final election $\mathcal{E}'$, for every c_x with $1 \leq x \leq 3\kappa$, every candidate in $I(c_x)$ must be beaten by at least one other candidate which is also in $I(c_x)$. Therefore, for every $c_x \in U$ there is at least one vote π_s in $\Pi_{\mathcal{V}'}$ with preference $c_x^2 \succ_s c_x^3 \succ_s c_x^1$ (Assume this is not true. Due to the construction, there will be in total $\frac{5}{6}\kappa$ votes with preference $c_x^1 \succ c_x^2 \succ c_x^3$, $\frac{5}{6}\kappa$ votes with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and $\frac{5}{3}\kappa$ votes with preference $c_x^3 \succ c_x^1 \succ c_x^2$, implying c_x^3 beats c_x^1 and ties c_x^2, contradicting with the fact that there is at least one candidate in $I(c_x)$ which beats c_x^3.). The vote π_s corresponds to an $s \in S'$ with $c_x \in s$. Since U contains exactly 3κ elements and due to the construction every unregistered vote gives exactly three different c_x with preference $c_x^2 \succ c_x^3 \succ c_x^1$, the union of such subsets s (that is S') forms an exact 3-set cover of $\mathcal{F}$.

The $\mathcal{NP}$-hardness reduction for CCAV-Copeland^1-NON can be modified from the above construction by deleting the candidate p'' from the election.

In the following, we show the $\mathcal{NP}$-hardness for CCDV-Copeland^1-UNI from X3C. For each $c \in U$, let $o(c)$ be the number of sets in S which contain c, and let $\bar{o}(c)$ be the number of sets in S which do not contain c. We assume that $o(c) \geq 3$ and $\bar{o}(c) \geq \kappa - 1$ for all $c \in U$ and $|S| \geq \kappa + 2$. For a given instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S)$ of X3C, we construct an instance $\mathcal{E}$ for CCDV-Copeland^1-UNI restricted to elections with single-peaked width 3 as follows. The candidate set and the single-peaked partition are the same as for CCAV-Copeland^1-UNI.

Votes: There are $2|S| - \kappa$ votes in total. Precisely, we create $|S| - \kappa$ votes with peaks at $I(p)$, and $|S|$ votes (corresponding to S) with peaks at $I(c_{3\kappa})$. The central idea is to construct the votes in such way that all deleted votes have peaks at $I(c_{3\kappa})$ whenever $\mathcal{E}$ is a yes-instance. Furthermore, after deleting these votes, each candidate except p, is beaten by at least another candidate which is from the same interval.

We first create the votes corresponding to S. For each $s = \{c_i, c_j, c_k\} \in S$, we create a vote π_s with preference $\succ_s$. The peak of the vote π_s is at $I(c_{3\kappa})$ and $p \succ_s p' \succ_s p''$. For every two candidates $a \in I(c_x)$ and $b \in I(c_y)$ with $x < y$, we have that $b \succ_s a$. With regard to the preference in each $I(c_x)$ with $1 \leq x \leq 3\kappa$, we set $c_x^2 \succ_s c_x^3 \succ_s c_x^1$ if $x \in \{i, j, k\}$ and set $c_x^3 \succ_s c_x^1 \succ_s c_x^2$ otherwise. Thus, there are in total $o(c_x)$ votes with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and $\bar{o}(c_x)$ votes with preference $c_x^3 \succ c_x^1 \succ c_x^2$.

We now construct the votes with peaks at $I(p)$. There are $|S| - \kappa$ such votes in total, all of which prefer p to p' to p''. Since all these votes have their peaks at $I(p)$, the preference between every two candidates $a \in I(c_x)$ and $b \in (c_y)$ with $x < y$ is $a \succ b$. Concerning the preference in each interval $I(c_x)$, we set $c_x^1 \succ c_x^2 \succ c_x^3$ in $\frac{1}{2} \cdot o(c_x)$ arbitrary votes. In the remaining votes, we set $c_x^2 \succ c_x^3 \succ c_x^1$ in $\bar{o}(c_x) - \kappa$ many of them and set $c_x^3 \succ c_x^1 \succ c_x^2$ in the rest. Clearly, $|S| - \kappa - \frac{1}{2} \cdot o(c_x) - \bar{o}(c_x) + \kappa = \frac{1}{2} \cdot o(c_x)$.

In summary, for each c_x, there are in total $\frac{1}{2} \cdot o(c_x)$ votes with preference $c_x^1 \succ c_x^2 \succ c_x^3$, $|S| - \kappa$ votes with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and $\frac{1}{2} \cdot o(c_x) + \bar{o}(c_x)$ votes with preference $c_x^3 \succ c_x^1 \succ c_x^2$. Moreover, all votes prefer p to p' to p''.

Number of Deleted Votes: $\mathcal{R} = \kappa$

Now we come to the correctness.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. We claim that deleting all votes corresponding to S', that is, all votes in $\Pi_{\mathcal{V}'} = \{\pi_s \mid s \in S'\}$, makes p the unique winner. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in $\Pi_{\mathcal{V}'}$. Since in the final election $\mathcal{E}'$ the number of votes with peaks at $I(p)$ is equal to the number of votes with peaks at $I(c_{3\kappa})$, and there is no vote with peak between $I(p)$ and $I(c_{3\kappa})$, any two candidates from different intervals are tied. Moreover, for each c_x with $1 \leq x \leq 3\kappa$, c_x^1 beats c_x^2, c_x^2 beats c_x^3 and c_x^3 beats c_x^1 in $\mathcal{E}'$ (to see this, observe that $N_{\mathcal{E}'}(c_x^1, c_x^2) - N_{\mathcal{E}'}(c_x^2, c_x^1) = 2$, $N_{\mathcal{E}'}(c_x^2, c_x^3) - N_{\mathcal{E}'}(c_x^3, c_x^2) = o(c_x) - 2 > 0$ and $N_{\mathcal{E}'}(c_x^3, c_x^1) - N_{\mathcal{E}'}(c_x^1, c_x^3) = |S| + \bar{o}(c_x) - 2\kappa > 0$). Thus, each candidate except p has Copeland^1 score at most $9\alpha \cdot \kappa + 1$ in $\mathcal{E}'$ (p'' has Copeland^1 score $9\alpha \cdot \kappa$ in $\mathcal{E}'$). Since p is preferred to p' and p'' by all votes, p has Copeland^1 score $9\alpha \cdot \kappa + 2$ in $\mathcal{E}'$, implying that p is the unique winner.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and $\Pi_{\mathcal{V}'}$ is a solution. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in $\Pi_{\mathcal{V}'}$. Observe first that all votes in $\Pi_{\mathcal{V}'}$ must have their peak at $I(c_{3\kappa})$, since otherwise, at least one of $I(c_{3\kappa})$ would be a winner. Let $S' = \{s \mid \pi_s \in \Pi_{\mathcal{V}'}\}$. We claim that S' is an exact 3-set cover of $\mathcal{F}$. Clearly, $\Pi_{\mathcal{V}'}$ contains exactly κ votes with peaks at $I(c_{3\kappa})$, since otherwise, there would be more votes with peaks at $I(c_{3\kappa})$ than votes with peaks at $I(p)$ in $\mathcal{E}'$, resulting in at least one of $c_{3\kappa}^1, c_{3\kappa}^2$ and $c_{3\kappa}^3$ being a winner in $\mathcal{E}'$. Therefore, after deleting all votes in $\Pi_{\mathcal{V}'}$, every two candidates from two different intervals must be tied. Moreover, since p is the unique winner in the final election $\mathcal{E}'$, every candidate in an interval $I(c_x)$ must be beaten by at least one candidate from the interval $I(c_x)$ in $\mathcal{E}'$. Therefore, for every $c_x \in U$, there is at least one vote, corresponding to an $s \in S'$ containing c_x, in $\Pi_{\mathcal{V}'}$ having the preference $c_x^2 \succ c_x^3 \succ c_x^1$ (since otherwise, there would be in total $\frac{1}{2}|o(c_x)|$ votes with preference $c_x^1 \succ c_x^2 \succ c_x^3$, $|S| - \kappa$ votes with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and $\frac{1}{2}|o(c_x)| + |\bar{o}(c_x)| - \kappa$ votes with preference $c_x^3 \succ c_x^1 \succ c_x^2$ in $\mathcal{E}'$, implying that c_x^2 beats c_x^3 and ties c_x^1). Since U contains exactly 3κ elements and each vote in $\Pi_{\mathcal{V}'}$ gives exactly three different c_x with preference $c_x^2 \succ c_x^3 \succ c_x^1$, S' must be an exact 3-set cover of $\mathcal{F}$.

The proof for CCDV-Copeland^1-NON can be modified from the construction for CCDV-Copeland^1-UNI by deleting the candidate p'. □

Now, we discuss destructive control by adding/deleting votes for Copeland$^\alpha$. In contrast to the $\mathcal{NP}$-hardness of constructive control by adding/deleting votes in Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ when restricted to elections with single-peaked width 3, we show that the destructive counterparts can be solved in polynomial time, if the single-peaked width is bounded by a constant. More precisely, from the parameterized complexity perspective, we prove that destructive control by adding/deleting votes for Copeland$^\alpha$ with $0 \leq \alpha \leq 1$ are $\mathcal{FPT}$ with respect to single-peaked width. Recall that all these problems are $\mathcal{NP}$-hard in the general case [112].

To present the $\mathcal{FPT}$-algorithm, we first introduce the following lemmas. Intuitively, the first lemma states that the closer a candidate outside the median group to the boundary of the median group is, the greater the Copeland$^\alpha$ score it has.

Lemma 3.1. *Let $\mathcal{G}[C_l, C_r]$ be the median group of an election with respect to the single-peaked partition $(C_1, C_2, ..., C_\omega)$. Let $a_1 \in C_{z_1}, a_2 \in C_{z_2}, b_1 \in C_{x_1}, b_2 \in C_{x_2}$ be four candidates with $z_2 < z_1 \leq l \leq r \leq x_1 < x_2$. Then, for every $0 \leq \alpha \leq 1$, the Copeland$^\alpha$ score of b_1 is strictly greater than that of b_2, and the Copeland$^\alpha$ score of a_1 is strictly greater than that of a_2.*

Proof. Due to symmetry, we need only to prove the claim for b_1 and b_2. Recall that in a Copeland$^\alpha$ election, every candidate c is compared with every other candidate. In each comparison, the candidate c gets 1 point if it beats its rival, and gets α point if it ties with its rival, otherwise, it gets 0 points. Let $\mathcal{C}_1$ be the set of candidates contained in the intervals on the right-side of C_{x_1}. Clearly, $b_2 \in \mathcal{C}_1$. Moreover, b_1 beats every candidate b' in $\mathcal{C}_1$, since all votes with peaks at C_r or on the left-side of C_r, which amount to more than half of the votes, prefer b_1 to b'. Thus, even b_2 also beats every candidate in $\mathcal{C}_1 \setminus \{b_2\}$, the candidates in $\mathcal{C}_1$ contribute one more point to b_1 than to b_2. Now consider the candidates in $\mathcal{C}_2 = \mathcal{C} \setminus (\mathcal{C}_1 \cup C_{x_1})$. These candidates are in intervals on the left-side of C_{x_1}. Due to the definitions of single-peaked election and single-peaked partition, for every candidate $c \in \mathcal{C}_2$, every vote which prefers b_2 to c also prefers b_1 to c. Thus, if b_2 beats (resp. ties) a candidate $c \in \mathcal{C}_2$, so does b_1 (resp. b_1 beats c or ties c). Thus, the candidates in $\mathcal{C}_2$ contribute to b_1 at least as the same points as to b_2. Since every candidate in C_{x_1} beats b_2, the lemma follows. □

Due to Lemma 3.1, we know that for every candidate c which is not in the median group, there exists at least one candidate who has a strictly greater Copeland$^\alpha$ score than that of c. This implies the following lemma.

Lemma 3.2. *All Copeland$^\alpha$ winners, for all $0 \leq \alpha \leq 1$, are in the median group.*

The correctness of the following lemma follows from the fact that the candidates in every interval are ranked together by every vote.

Lemma 3.3. *For every two intervals C_x and C_y. If a candidate in C_x beats a candidate in C_y, then every candidate in C_x beats every candidate in C_y.*

The following lemma is also useful.

Lemma 3.4. *Let $\mathcal{E} = \{\mathcal{C}, \Pi_{\mathcal{V}}\}$ be an election with single-peaked partition $P = (C_1, C_2, ..., C_\omega)$. Let $\mathcal{G}[C_l, C_r]$ be the median group. Let $C_{z_1}, C_{z_2}, C_{x_1}, C_{x_2}$ be four intervals with $z_2 < z_1 < l \leq r < x_1 < x_2$. Let C_y be an arbitrary interval in the median group. If every candidate in C_y beats every candidates in C_{z_1}, then every candidate in C_y beats every candidate in C_{z_2}. Symmetrically, if every candidate in C_y beats every candidates in C_{x_1}, then every candidate in C_y beats every candidate in C_{x_2}.*

Proof. Due to symmetry, we only need to prove the lemma for C_{z_1}, C_{z_2}. Let c be a candidate in C_y and c' be a candidate in C_{z_1}. Since all of the half votes with peaks on the right-side of C_l prefer c to c', and all votes with peaks at C_{z_1} or on the left-side of C_{z_1} prefer c' to c, c beats c' if and only if there is at least one vote, with peak between C_r and C_{z_1}, which prefers c to c'. Due to the definition of single-peaked partition, such a vote must also prefer c to every candidate in C_{z_2}. Therefore, if c beats c', c must also beats every candidate in C_{z_2}. Due to Lemma 3.3, the lemma is proved. □

Recall that every candidate in the median group beats or ties with every candidate not in the median group. Lemma 3.3 and Lemma 3.4 together imply that for every candidate c in some interval in the median group, there are two integers $1 \leq z \leq l$ and $r \leq x \leq \omega$ such that c beats all candidates in $\cup_{i \in [1,z] \cup [x,\omega]} C_i$ and ties with all the remaining candidates that are not in the median group.

Theorem 3.5. *DCAV-Copeland$^\alpha$-UNI, DCAV-Copeland$^\alpha$-NON, DCDV-Copeland$^\alpha$-UNI and DCDV-Copeland$^\alpha$-NON are $\mathcal{FPT}$ with respect to single-peaked width, for every* $0 \leq \alpha \leq 1$

Proof. Let $P = (C_1, C_2, ..., C_\omega)$ be the single-peaked partition, and k be the single-peaked width. We derive $\mathcal{FPT}$-algorithms for the problems stated in the theorem. Recall that a Copeland$^\alpha$ winner must be included in the median group according to Lemma 3.2. Thus, to make p not a winner, there are two possibilities:

(1) make p outside the median group; or

(2) make p inside the median group but simultaneously make another candidate in the median group have a Copeland$^\alpha$ score higher (nonunique-winner model) or no less (unique-winner model) than that of p.

Our algorithms firstly consider the former case. Thus, for all the problems stated in the theorem, we first calculate the minimum number β of votes to be added or deleted to make p outside the median group. Clearly, this can be done in polynomial-time. Moreover, if $\beta \leq \mathcal{R}$, we are done. However, if it turns out that $\beta > \mathcal{R}$, then we cannot make p outside the median group by modifying at most $\mathcal{R}$ votes. In this case, we consider the latter case of making p in the median group. Precisely, we do the following.

First, we enumerate all possible candidates p' which can prevent p from being the winner by modifying at most $\mathcal{R}$ votes. Then, for each p', we enumerate all possible median groups $\mathcal{G}[C_l, C_r]$ which can appear by modifying at most $\mathcal{R}$ votes; clearly $\mathcal{G}[C_l, C_r]$ must contain the interval C_i containing p and the interval C_j containing p'. Next, we enumerate all possible combinations of four integers $\chi_l, \chi_r, \phi_l, \phi_r$ with $\chi_l, \phi_l < l$ and $\chi_r, \phi_r > r$. Here χ_l (resp. χ_r) indicates the right-most interval on the left-side (resp. the left-most interval on the right-side) of the median group in which all candidates are beaten by p. Due to Lemma 3.4, p also beats every candidate in the interval on the left-side (resp. right-side) of C_{χ_l} (resp. C_{χ_r}). The two integers ϕ_l and ϕ_r indicate the similar meaning but with respect to p'. Finally, we consider all possible partitions $(C_i^1, C_i^\alpha, C_i^0)$ of $C_i \setminus \{p\}$, and all possible partitions $(C_j^1, C_j^\alpha, C_j^0)$ of $C_j \setminus \{p'\}$ such that $|C_i^1| + \alpha(|C_i^\alpha| + |C_j|) + \text{score}(p) \leq |C_j^1| + \alpha(|C_j^\alpha| + |C_i|) + \text{score}(p')$ (for the unique-winner case) or $|C_i^1| + \alpha(|C_i^\alpha| + |C_j|) + \text{score}(p) < |C_j^1| + \alpha(|C_j^\alpha| + |C_i|) + \text{score}(p')$ (for the nonunique-winner case). Here, score(p) and score(p') are the Copeland$^\alpha$ scores of p and p' contributed by the candidates outside the median group. These scores can be calculated in polynomial time for fixed values of $\chi_l, \chi_r, \phi_l, \phi_r$, due to Lemma 3.4. Moreover, C_i^1, C_i^α, C_i^0 (resp. C_j^1, C_j^α, C_j^0) are the sets of prospective candidates in C_i (resp. C_j) that p (resp. p') beats, ties and be beaten in the final election, respectively. Clearly, if both p and p' are in the median group, p' has a higher score (resp. a no less score) than that of p if and only if $|C_i^1| + \alpha(|C_i^\alpha| + |C_j|) + \text{score}(p) < |C_j^1| + \alpha(|C_j^\alpha| + |C_i|) + \text{score}(p')$ (resp. $|C_i^1| + \alpha(|C_i^\alpha| + |C_j|) + \text{score}(p) \leq |C_j^1| + \alpha(|C_j^\alpha| + |C_i|) + \text{score}(p')$. This is due to the fact that every two candidates from different intervals in the median group are tied.

The above enumeration results in at most $m^5 \times \binom{\omega}{2} \times 3^{2\mathcal{k}-2}$ subinstances, where m is the number of candidates, ω is the number of intervals, and $\mathcal{k}$ is the single-peaked width. Concretely, in each subinstance, we have, in addition to the original input, also a candidate p', two intervals C_l and C_r with $l \leq i, j \leq r$ (note: $p \in C_i, p' \in C_j$), four integers $\chi_l, \chi_r, \phi_l, \phi_r$, a partition $(C_i^1, C_i^\alpha, C_i^0)$ of $C_i \setminus \{p\}$ and a partition $(C_j^1, C_j^\alpha, C_j^0)$ of $C_j \setminus \{p'\}$. We are asked to add/delete at most $\mathcal{R}$ votes such that

(1) the median group is $\mathcal{G}[C_l, C_r]$;

(2) p beats all candidates in $C_i^1 \cup C_{\chi_l} \cup C_{\chi_r}$, ties all candidates in C_i^α and be beaten by all candidates in C_i^0; and

(3) p' beats all candidates in $C_j^1 \cup C_{\phi_l} \cup C_{\phi_r}$, ties all candidates in C_j^α and be beaten by all candidates in C_j^0.

Clearly, the first condition can be easily checked in polynomial time. Thus, all such subinstances which we cannot make $\mathcal{G}[C_l, C_r]$ the median group by adding/deleting at most $\mathcal{R}$ votes are discarded immediately. We focus on the remaining subinstances. To solve them, we reduce each subinstance to an ILP instance with bounded number of variables (bounded by a function of the parameter $\mathcal{k}$). For this purpose, we divide the modifiable votes into several parts. Recall that modifiable votes refer to unregistered votes in the case of control by adding votes, but refer to registered votes in the case of control by deleting votes. In control by adding votes, we divide the modifiable votes into two parts: votes with peaks on the left-side of C_{l+1}; and votes with peaks on the right-side of C_{r-1}. Moreover, we distinguish between two cases: $l \neq r$ and $l = r$. For control by deleting votes, however, we divide the modifiable votes into three parts: votes with peaks on the left-side of C_i, votes with peaks at C_i and votes with peaks on the right-side of C_i. Then, for both the former case and the latter case, each part is further divided in to many subparts each containing all votes which are consistent with respect to $\{p, p'\}$ and $C_i \cup C_j \cup \{a_1, a_2, b_1, b_2\}$, where a_1, a_2, b_1, b_2 are any arbitrary candidates from $C_{\chi_l}, C_{\chi_r}, C_{\phi_l}, C_{\phi_r}$, respectively. According to Lemma 3.3, if p (resp. p') beats a_1 and a_2 (resp. b_1 and b_2), then p (resp. p') beats every candidate in $C_{\chi_l} \cup C_{\chi_r}$ (resp. $C_{\phi_l} \cup C_{\phi_r}$).

There are at most $2 \times 2 \times 3^{2k+2} = 36 \times 9^{k}$ subparts for control by adding votes and at most 54×9^{k} subparts for control by deleting votes. By assigning each subpart a variable, we reduce the problems to ILP's with bounded number of variables, which can be solved in $\mathcal{FPT}$-time with respect to k. Here, each variable specifies how many votes from the corresponding subpart are in the solution. The constrictions should serve the three conditions stated above. □

3.4 Maximin Control

In this section, we focus on control problems for Maximin. It is known that constructive/destructive control by adding/deleting votes are for Maximin $\mathcal{NP}$-hard in the general case [110]. Moreover, all these problems are $\mathcal{W}[1]$-hard with respect to the number of added/deleted votes as the parameter in the general case [184]. Our main results of this section are summarized in Table 3.3. Even though Maximin and Copeland^1 are two different voting correspondences, our results show that the complexity of the control problems studied in this section for Maximin behave in the same way as Copeland^1.

The next theorem follows from the facts that

(1) Maximin is weakCondorcet-consistent [44];

(2) there is at least one weak Condorcet winner in every election with single-peaked width 2; and

(3) the constructive control by adding/deleting votes for (weak) Condorcet are polynomial-time solvable in elections with single-peaked width 2 (implied by Theorem 3.2).

	Single-peaked width k		
	$k = 2$	$k = 3$	k: parameter
CCAV	$\mathcal{P}$	$\mathcal{NP}$-hard	
CCDV			
DCAV	$\mathcal{P}$		$\mathcal{FPT}$
DCDV			

Table 3.3: Complexity of the contructive/destructive control by adding/deleting votes for Maximin. All results shown in the table apply to both the unique-winner model and the nonunique-winner model. Here, "$\mathcal{P}$" stands for polynomial-time solvable.

Theorem 3.6. *CCAV-Maximin-UNI, CCAV-Maximin-NON, CCDV-Maximin-UNI and CCDV-Maximin-NON are polynomial-time solvable in elections with single-peaked width 2.*

Then, we consider elections with single-peaked width 3. The following theorem shows that the polynomial-time solvability for these constructive control problems do not hold any more in elections with single-peaked width 3.

Theorem 3.7. *CCAV-Maximin-UNI, CCAV-Maximin-NON, CCDV-Maximin-UNI and CCDV-Maximin-NON are $\mathcal{NP}$-hard in elections with single-peaked width 3.*

Proof. We first consider CCAV-Maximin-UNI. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S)$ of X3C, we construct an instance $\mathcal{E}$ of CCAV-Maximin-UNI as follows.

Candidates: For each $c_x \in U$, we create three candidates c_x^1, c_x^2, c_x^3 which form an interval denoted by $I(c_x)$. In addition, we have three candidates p, p' and p'' which form an interval denoted by $I(p)$. The distinguished candidate is p.

Single-Peaked Partition: $(I(p), I(c_1), I(c_2), ..., I(c_{3\kappa}))$.

Registered Votes: Let η be an integer with $\eta \geq 3\kappa$ and $\eta \equiv 0 \mod 3$. We create $2\eta + 1$ registered votes. Precisely, we have

(1) $\frac{2}{3} \cdot \eta - \kappa + 1$ votes defined as

$$p \succ p' \succ p'' \succ c_1^1 \succ c_1^2 \succ c_1^3 \succ, ..., \succ c_{3\kappa}^1 \succ c_{3\kappa}^2 \succ c_{3\kappa}^3$$

(2) κ votes defined as

$$p' \succ p \succ p'' \succ c_1^1 \succ c_1^2 \succ c_1^3 \succ, ..., \succ c_{3\kappa}^1 \succ c_{3\kappa}^2 \succ c_{3\kappa}^3$$

(3) $\frac{1}{3}\eta$ votes defined as

$$p' \succ p'' \succ p \succ c_1^2 \succ c_1^3 \succ c_1^1 \succ, ..., \succ c_{3\kappa}^2 \succ c_{3\kappa}^3 \succ c_{3\kappa}^1$$

(4) $\frac{1}{3}\eta$ votes defined as

$$c_{3\kappa}^2 \succ c_{3\kappa}^3 \succ c_{3\kappa}^1 \succ, ..., c_1^2 \succ c_1^3 \succ c_1^1 \succ p' \succ p'' \succ p$$

(5) $\frac{2}{3}\eta$ votes defined as

$$c_{3\kappa}^3 \succ c_{3\kappa}^1 \succ c_{3\kappa}^2 \succ, ..., c_1^3 \succ c_1^1 \succ c_1^2 \succ p'' \succ p \succ p'$$

It is easy to verify that p' is the current unique winner.

Unregistered Votes: For each $s = \{c_i, c_j, c_k\} \in S$, we create a vote π_s with preference $\succ_s$ and with peak at $I(p)$. Moreover, in the interval $I(p)$, we set $p \succ_s p' \succ_s p''$. For every $I(c_x)$, we set $c_x^2 \succ_s c_x^3 \succ_s c_x^1$ if $x \in \{i, j, k\}$, and set $c_x^1 \succ_s c_x^2 \succ_s c_x^3$ otherwise.

Number of Added Votes: $\mathcal{R} = \kappa$.

In the following, we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{E}$ is a yes-instance.

($\Rightarrow$:) Let S' be a solution of $\mathcal{F}$. We claim that the set of unregistered votes corresponding to S', that is, $\Pi_{\mathcal{V}'} = \{\pi_s \mid s \in S'\}$ form a solution for $\mathcal{E}$. Since S' is an exact 3-set cover, for each $I(c_x)$, there is exactly one vote in $\Pi_{\mathcal{V}'}$ with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and exactly $\kappa - 1$ votes with preference $c_x^1 \succ c_x^2 \succ c_x^3$. Then, it is easy to calculate that, after adding all votes in $\Pi_{\mathcal{V}'}$ to the registered votes, c_x^1 has the highest Maximin score $\frac{2}{3}\eta + \kappa$ among all candidates in $I(c_x)$ for all $0 \leq x \leq 3\kappa$. Moreover, since all unregistered votes prefer p to p' to p'', p has the highest Maximin score $\frac{2}{3}\eta + \kappa + 1$ among all candidates in $I(p)$, which is also the highest Maximin score among all candidates in the final election. Hence, p becomes the unique winner.

($\Leftarrow$:) Let $\Pi_{\mathcal{V}'}$ be a solution of $\mathcal{E}$, and let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by adding all votes in $\Pi_{\mathcal{V}'}$ to the registered votes. We claim that the subset S' corresponding to $\Pi_{\mathcal{V}'}$, that is $S' = \{s \mid \pi_s \in_+ \Pi_{\mathcal{V}'}\}$, is an exact 3-set cover of $\mathcal{F}$. We first observe that $\Pi_{\mathcal{V}'}$ contains exactly κ votes, since otherwise, p' would have a Maximin score no less than that of p in $\mathcal{E}'$. Since all unregistered votes prefer p to p' to p'', p has a final Maximin score $\frac{2}{3}\eta + \kappa + 1$ in $\mathcal{E}'$. Therefore, for every $c_x \in U$, there is at least one vote π_s in $\Pi_{\mathcal{V}'}$ with preference $c_x^2 \succ_s c_x^3 \succ_s c_x^1$, since otherwise, c_x^1 would have a Maximin score no less than that of p in $\mathcal{E}'$. Since U contains exactly 3κ elements and every unregistered vote have for three different c_x preference $c_x^2 \succ c_x^3 \succ c_x^1$, S' must be an exact 3-set cover.

The reduction for the CCAV-Maximin-NON is the same as the above reduction with only the difference that we create one less registered vote of the first type.

Now we study CCDV-Maximin-UNI. Our reduction is again from X3C. However, in this case, we assume that each element $c_i \in U$ occurs in exactly three different 3-subsets of S. Therefore, we have that $|S| = 3\kappa$. Let $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S)$ be an instance of X3C, we construct an instance $\mathcal{E}$ of CCDV-Maximin-UNI as follows.

Candidates: For each $c_x \in U$, we create three candidates c_x^1, c_x^2, c_x^3 which form an interval denoted by $I(c_x)$. In addition, we have the distinguished candidate p which forms an interval denoted by $I(p) = \{p\}$.

Single-Peaked Partition: $(I(p), I(c_1), I(c_2), ..., I(c_{3\kappa}))$.

Votes: We create in total $2|S| - \kappa$ votes with $|S|$ of them having peak at the interval $I(c_{3\kappa})$ and $|S| - \kappa$ having peak at the interval $I(p)$. In particular, we create the following votes with peak at the interval $I(p)$.

(1) 2 votes defined as

$$p \succ c_1^1 \succ c_1^2 \succ c_1^3 \succ, ..., \succ c_{3\kappa}^1 \succ c_{3\kappa}^2 \succ c_{3\kappa}^3$$

(2) $|S| - \kappa - 3$ votes defined as

$$p \succ c_1^2 \succ c_1^1 \succ c_1^3 \succ, ..., \succ c_{3\kappa}^2 \succ c_{3\kappa}^1 \succ c_{3\kappa}^3$$

(3) 1 vote defined as

$$p \succ c_1^3 \succ c_1^1 \succ c_1^2 \succ, ..., \succ c_{3\kappa}^3 \succ c_{3\kappa}^1 \succ c_{3\kappa}^2$$

The votes with peak at the interval $I(c_{3\kappa})$ correspond to the subsets in S. In particular, for each $s = \{c_i, c_j, c_k\} \subseteq S$, we create a vote π_s with preference $\succ_s$. Since the peak of the vote is at $I(c_{3\kappa})$, for every two candidates $a \in I(c_x)$ and $b \in I(c_y)$ with $x < y$, we have that $b \succ_s a$. Moreover, p is ranked in the last. With regard to the preference in each $I(c_x)$ for every $1 \leq x \leq 3\kappa$, we set $c_x^2 \succ c_x^3 \succ c_x^1$ if $x \in \{i, j, k\}$ and set $c_x^3 \succ c_x^1 \succ c_x^2$ otherwise.

In summary, for each c_x, there are in total 2 votes with preference $c_x^1 \succ c_x^2 \succ c_x^3$, $|S| - \kappa - 3$ votes with preference $c_x^2 \succ c_x^1 \succ c_x^3$, 3 votes with preference $c_x^2 \succ c_x^3 \succ c_x^1$ and $|S| - 2$ votes with preference $c_x^3 \succ c_x^1 \succ c_x^2$. The comparisons between candidates in the same interval $I(c_x)$ are summarized as follows.

- $N_{\mathcal{E}}(c_x^1, c_x^2) = |S|$ and $N_{\mathcal{E}}(c_x^1, c_x^3) = |S| - \kappa - 1$.
- $N_{\mathcal{E}}(c_x^2, c_x^1) = |S| - \kappa$ and $N_{\mathcal{E}}(c_x^2, c_x^3) = |S| - \kappa + 2$.
- $N_{\mathcal{E}}(c_x^3, c_x^1) = |S| + 1$ and $N_{\mathcal{E}}(c_x^3, c_x^2) = |S| - 2$.

Now it is easy to verify that $c^3_{3\kappa}$ is the current winner.

Number of Deleted Votes: $\mathcal{R} = \kappa$.

In the following, we prove the correctness of the reduction.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. We claim that deleting all votes corresponding to S', that is, all votes in $\Pi_{\mathcal{V}'} = \{\pi_s \mid s \in S'\}$, makes p the unique winner. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in $\Pi_{\mathcal{V}'}$. Since in the final election $\mathcal{E}'$ the number of votes with peaks at $I(p)$ is equal to the number of votes with peaks at $I(c_{3\kappa})$, and there is no vote with peak between $I(p)$ and $I(c_{3\kappa})$, any two candidates from different intervals are tied, that is $N_{\mathcal{E}'}(a,b) = N_{\mathcal{E}'}(b,a) = |S| - \kappa$, for every two candidates a, b who come from different intervals. Moreover, since S' is an exact 3-set cover, for each c_x there is exactly one $s \in S'$ with $c \in S'$, which corresponds to a vote in $\Pi_{\mathcal{V}'}$ with preference $c^2_x \succ c^3_x \succ c^1_x$, and exactly $\kappa - 1$ many 3-subsets $s \in S'$ with $c \notin s$, which correspond to exactly κ votes in $\Pi_{\mathcal{V}'}$ with preference $c^3_x \succ c^1_x \succ c^2_x$. According to this, for each candidates in each $I(c_x)$ we have the following facts.

- $N_{\mathcal{E}'}(c^1_x, c^2_x) = |S| - \kappa + 1$ and $N_{\mathcal{E}'}(c^1_x, c^3_x) = |S| - \kappa - 1$.
- $N_{\mathcal{E}'}(c^2_x, c^1_x) = |S| - \kappa - 1$ and $N_{\mathcal{E}'}(c^2_x, c^3_x) = |S| - \kappa + 1$.
- $N_{\mathcal{E}'}(c^3_x, c^1_x) = |S| - \kappa + 1$ and $N_{\mathcal{E}'}(c^3_x, c^2_x) = |S| - \kappa - 1$.

Therefore, all candidate except p have Maximin score $|S| - \kappa - 1$ in the final election $\mathcal{E}'$. Since none of the $|S| - \kappa$ votes with peak at $I(p)$ is deleted, we know that the Maximin score of p is $|S| - \kappa$, and thus, p becomes the unique winner in the final election.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and $\Pi_{\mathcal{V}'}$ is a solution. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by deleting all votes in $\Pi_{\mathcal{V}'}$. Observe first that none of the votes in $\Pi_{\mathcal{V}'}$ has peak at $I(p)$, since otherwise, $c^3_{3\kappa}$ would be a winner. Therefore, the votes that are deleted must from the votes corresponding to the 3-subsets in S. Let $S' = \{s \mid \pi_s \in \Pi_{\mathcal{V}'}\}$. We claim that S' is an exact 3-set cover of $\mathcal{F}$. Clearly, $\Pi_{\mathcal{V}'}$ contains exactly κ votes with peaks at $I(c_{3\kappa})$, since otherwise, either $c^2_{3\kappa}$ or $c^3_{3\kappa}$ would be a winner in $\mathcal{E}'$. Therefore, after deleting all votes in $\Pi_{\mathcal{V}'}$, every two candidates from two different intervals must be tied, that is $N_{\mathcal{E}'}(a,b) = N_{\mathcal{E}'}(b,a) = |S| - \kappa$, for every two candidates a, b who come from different intervals. Moreover, the Maximin score of p is $|S| - \kappa$. Since p is the unique winner in $\mathcal{E}'$, for every $c_x \in U$ and the candidate c^2_x, there must be at least one vote π_s in $\Pi_{\mathcal{V}'}$ which prefers c^2_x to c^1_x (this is due to that $N_{\mathcal{E}}(c^2_x, c^1_x) = |S| - \kappa$). Due to the construction, such a vote corresponds to an $s \in S$ with $c_x \in s$. Since $|S'| = \kappa$, each $s \in S'$ is a 3-subset of U and $|U| = 3\kappa$, S' must be an exact 3-set cover of $\mathcal{F}$.

The proof for CCDV-Maximin-NON can be modified from the construction for CCDV-Maximin-UNI by creating one less vote defined as

$$p \succ c_1^1 \succ c_1^2 \succ c_1^3 \succ, ..., \succ c_{3\kappa}^1 \succ c_{3\kappa}^2 \succ c_{3\kappa}^3.$$

By doing so, the Maximin score of p will be $|S| - \kappa - 1$ in the final election. The correctness argument is similar to the one for CCDV-Maximin-UNI. □

From the parameterized point of view, the above theorem implies that the constructive/destructive control by adding/deleting votes for Maximin are beyond $\mathcal{XP}$ when we take the single-peaked width as the parameter.

Now we study the destructive control problems for Maximin. Before proceeding further, we introduce some properties of Maximin elections with bounded single-peaked width. These properties are also helpful in understanding the behavior of the Maximin correspondence. The first property is formally stated in Lemma 3.5. In an informal way, it states that for each candidate c, the closer another candidate c' lies to c according to the single-peaked partition, the less is the number of voters who prefer c to c'. For a positive integer n, let $[n]$ be the set $\{1, 2, ..., n\}$.

Lemma 3.5. *Let $(C_1, C_2, ..., C_\omega)$ be the single-peaked partition of a given election and c be a candidate in a certain interval C_i. Then, $N(c, b_1) \leq N(c, b_2)$ for all $b_1 \in C_{x_1}$ and $b_2 \in C_{x_2}$ with $i < x_1 < x_2 \leq \omega$, and $N(c, a_1) \leq N(c, a_2)$ for all $a_1 \in C_{z_1}$ and $a_2 \in C_{z_2}$ with $1 \leq z_2 < z_1 < i$.*

Proof. We first prove the first part of the claim. Let b_1 and b_2 be the two candidates as stated in the lemma. For all $j \in [\omega]$, we denote the multiset of votes with peaks at C_j or on the right-side of C_j by $\mathcal{V}_j^r$, and denote the multiset of votes with peaks at C_j or on the left-side of C_j by $\mathcal{V}_j^l$. It is obvious that all votes in $\mathcal{V}_i^l$ prefer c to b_1 to b_2 and all votes in $\mathcal{V}_{x_1}^r$ prefer b_1 to c. Let $\mathcal{V}_{i,x_1}^{c \succ b_1}$ be the multiset of votes with peaks between C_i and C_{x_1} and prefer c to b_1. Thus, $N(c, b_1) = |\mathcal{V}_i^l| + |\mathcal{V}_{i,x_1}^{c \succ b_1}|$. Due to the definition of single-peaked partition, all votes in $\mathcal{V}_{i,x_1}^{c \succ b_1}$ prefer c to b_2. Therefore, $N(c, b_2) \geq |\mathcal{V}_i^l| + |\mathcal{V}_{i,x_1}^{c \succ b_1}| = N(c, b_1)$.

Due to symmetry, the second part is also correct. □

Recall that the Maximin score of a candidate c is equal to $N(c, c')$ where c' achieves the minimum value of $N(c, \cdot)$. Let c be a candidate from a certain interval C_i. Let $MIN(c)$ be the set of candidates that achieve the minimum value of $N(c, \cdot)$; hence, we have that $Maximin(c) = N(c, c')$ for every $c' \in MIN(c)$, where $Maximin(c)$ is the Maximin score of the candidate c. According to Lemma 3.5, we have that $(C_{i-1} \cup C_i \cup C_{i+1}) \cap MIN(c) \neq \emptyset$. Therefore, to determine the Maximin score of c, it is sufficient to consider the election restricted to $C_{i-1} \cup C_i \cup C_{i+1}$ whose size is bounded by 3κ, where κ is the single-peaked width. In the following, we introduce another property which helps to improve the upper bound.

Lemma 3.6. *Let c be a candidate and C' be an interval with $c \notin C'$. Then, $N(c,a) = N(c,b)$ for every two candidates $a, b \in C'$.*

Proof. Since C' is an interval, all votes rank the candidates in C' contiguously. Therefore, each vote either prefers c to all candidates in C' or prefers all candidates in C' to c, implying that for every two candidates $a, b \in C'$, $N(c,a) = N(c,b)$. □

According to Lemmas 3.5 and 3.6, the Maximin score of a candidate c is determined by all candidates in the interval including c, together with any two arbitrary candidates from the two neighbor intervals of the interval including c, one from each. With Lemmas 3.5 and 3.6, we arrive at the following lemma.

Lemma 3.7. *Let $\mathcal{E}$ be an election with single-peaked partition $P = (C_1, C_2, ..., C_\omega)$ and c be a candidate in an interval C_i. Then the Maximin score of c in $\mathcal{E}$, denoted by $Maximin_{\mathcal{E}}(c)$, is*

$$Maximin_{\mathcal{E}}(c) = Maximin_{\mathcal{E}|C_i \cup \{a,b\}}(c)$$

Here, a and b are any two arbitrary candidates in C_{i-1} and C_{i+1}, respectively (only b appears if $i = 1$ and only a appears if $i = \omega$).

Now we are ready to show the some $\mathcal{FPT}$ results.

Theorem 3.8. *DCAV-Maximin-UNI, DCAV-Maximin-NON, DCDV-Maximin-UNI and DCDV-Maximin-NON are $\mathcal{FPT}$ with respect to single-peaked width.*

Proof. Let $P = (C_1, C_2, ..., C_\omega)$ be the single-peaked partition. Based on Lemma 3.7, we derive $\mathcal{FPT}$-algorithms for the problems stated in the theorem as follows. Let's consider the nonunique-winner model first. Let m be the number of candidates.

To make the distinguished candidate p not a winner, we need to make at least one other candidate p' have a higher Maximin score than that of p. The algorithms firstly enumerate all such candidates p'. This results in at most m subinstances, each seeks for at most $\mathcal{R}$ votes addition/deletion to make p' have a higher Maximin score than that of p. Let C_i and C_j be the intervals which contain p and p', respectively. Note that C_i and C_j could be the same interval. Due to Lemma 3.7, we can limit our attention to the election restricted to $C_i \cup C_j \cup \{a_i, b_i, a_j, b_j\}$, where a_z, b_z with $z \in \{i, j\}$ are any two arbitrary candidates in the intervals C_{z-1} and C_{z+1}, respectively. Then, we can use ILP to solve the problem in $\mathcal{FPT}$-time. To this end, we further enumerate all possible candidates $\bar{p} \in C_i \cup \{a_i, b_i\} \setminus \{p\}$ and $\bar{p}' \in C_j \cup \{a_j, b_j\} \setminus \{p'\}$ which are the prospective candidates that achieve the minimum values of $N(p, \cdot)$ and $N(p', \cdot)$, respectively. Then we divide the modifiable votes (recall that the modifiable votes refer to the unregistered votes in the adding votes case, and refer to all votes in the given election in the deleting votes case) into at most $(2\mathcal{k}+4)!$ parts each containing all votes with the same preference over the candidates $C_i \cup C_j \cup \{a_i, b_i, a_j, b_j\}$. Each part then is assigned a variable $x_{\succ}$, where the index $\succ$ indicates the preference of all votes in this part over all candidates in $C_i \cup C_j \cup \{a_i, b_i, a_j, b_j\}$. For DCAV-Maximin-NON, the ILP is subject to the following constraints.

(1) for all $c \in C_i \cup \{a_i, b_i\} \setminus \{p\}$

$$\sum_{p \succ c} x_{\succ} + N(p, c) - \sum_{p \succ \bar{p}} x_{\succ} - N(p, \bar{p}) \geq 0$$

(2) for all $c \in C_j \cup \{a_j, b_j\} \setminus \{p'\}$

$$\sum_{p' \succ c} x_{\succ} + N(p', c) - \sum_{p' \succ \bar{p}'} x_{\succ} - N(p', \bar{p}') \geq 0$$

(3) In addition, we have the following two restrictions.

$$\begin{cases} \sum_{\succ} x_{\succ} \leq \mathcal{R} \\ \sum_{p' \succ \bar{p}'} x_{\succ} + N(p', \bar{p}') - \sum_{p \succ \bar{p}} x_{\succ} - N(p, \bar{p}) > 0 \end{cases}$$

In all above three constraints, the value of $N(,\cdot,)$ is counted solely based on the registered votes. Moreover, (1) is to ensure that $\bar{p}$ is the candidate in $C_i \cup \{a_i, b_i\}$ that achieves the minimum value of $N_{\mathcal{E}'}(p, \cdot)$ in the final election $\mathcal{E}'$; (2) is to ensure that $\bar{p}'$ is the candidate in $C_j \cup \{a_j, b_j\}$ that achieves the minimum value of $N_{\mathcal{E}'}(p', \cdot)$ in the final election $\mathcal{E}'$; and (3) is to ensure that we add at most $\mathcal{R}$ votes and the Maximin score of p' is strictly greater than that of p in the final election.

For DCDV-Maximin-NON, the ILP subjects to the following constraints.

(1) for all $c \in C_i \cup \{a_i, b_i\} \setminus \{p\}$

$$N(p, c) - \sum_{p \succ c} x_{\succ} - N(p, \bar{p}) + \sum_{p \succ \bar{p}} x_{\succ} \geq 0$$

(2) for all $c \in C_j \cup \{a_j, b_j\} \setminus \{p'\}$

$$N(p', c) - \sum_{p' \succ c} x_{\succ} - N(p', \bar{p}') + \sum_{p' \succ \bar{p}'} x_{\succ} \geq 0$$

(3) In addition, we have the following two restrictions.

$$\begin{cases} \sum_{\succ} x_{\succ} \leq \mathcal{R} \\ N(p', \bar{p}') - \sum_{p' \succ \bar{p}'} x_{\succ} - N(p, \bar{p}) + \sum_{p \succ \bar{p}} x_{\succ} > 0 \end{cases}$$

Notice that in DCDV-Maximin-NON, we have only one multiset of votes. The value of $N(, \cdot,)$ is based on all the given votes.

The algorithms for DCAV-Maximin-UNI and DCDV-Maximin-UNI are similar to DCAV-Maximin-NON and DCDV-Maximin-NON, respectively, with only the difference that, in both cases, the last inequations are indicated by $\geq$ other than $>$. □

3.5 A General Framework

In this section, we consider elections containing an odd number of votes. Elections with an odd number of votes have been studied in different context [44, 113, 201, 233]. In such elections, there is no tie, while comparing two candidates. In addition, several theorems have been achieved for such elections, for example, see page 5 for May's theorem, page 234 for Sen's theorem and page 239 for Black's theorem in [233]. Especially, the Black's theorem implies that the Condorcet winner always exists in single-peaked elections with odd number of votes. Moreover it must be the top candidate of the median vote. The following lemma implies that with the odd-votes elections, the Smith set must be included in the median interval. Observe that if the number of votes is odd, the median group contains exactly one interval.

Lemma 3.8. *For every election with the median group containing only one interval, the median interval is a superset of the Smith set.*

Proof. To check the correctness of the lemma, observe that every candidate in the median interval beats every other candidate not in the median interval. □

Our main contribution of this section is a general theorem which can be used to derive $\mathcal{FPT}$ results for the constructive/destructive control by adding/deleting votes problems in odd-votes elections, that is, adding/deleting votes resulting in elections with odd number of votes.

Theorem 3.9. *For an odd-votes election with a voting correspondence passing the Smith-IIA criterion, if a constructive/destructive control by adding/deleting votes problem is $\mathcal{FPT}$ with the number of candidates as parameter, then the same problem is also $\mathcal{FPT}$ with single-peaked width as parameter. This claim holds for both the unique-winner and the nonunique-winner models.*

Proof. We first consider the constructive control. Since there are odd number of votes in the final election, the median group contains only one interval (the median interval). Due to Lemma 3.8, all voting correspondences passing the Smith-IIA criterion always select winners from the median interval. Thus, to solve the problems stated in the theorem, we have two objectives. One is to make the interval C_i containing the distinguished candidate p the median interval. The other objective is then to make p a winner. By the Smith-IIA criterion, we can focus on the election restricted to C_i, once the first objective has been reached. These observations motivate us to propose a general reduction rule, which significantly shrinks the size of the candidate set. The main idea of the reduction rule is to replace the "irrelevant candidates" by only two candidates x and y, where $\{x\}$ will be an interval and be placed on the left-side of C_i, and $\{y\}$ will also be an interval and be placed on the right-side of C_i in the single-peaked partition P. The role of the two candidates is to preserve the information of the peaks of all votes. More precisely, the reduction rule replaces each vote with a new vote containing only the candidates $C_i \cup \{x, y\}$. In particular, if a vote has its peak on the left-side (resp. right-side) of C_i, the new vote will have its peak at $\{x\}$ (resp. $\{y\}$). If the vote has its peak at C_i, the new vote will also have its peak at C_i. In all three cases, the new vote preserves the preference of the original one over the candidates in C_i. A formal description of the reduction rule is as follows.

Reduction Rule. Let $P = (C_1, C_2, ..., C_i, ..., C_\omega)$ be a single-peaked partition of the given election $\mathcal{E}$. We do the following operations (for control by adding votes, the operations should be implemented on both the registered votes and the unregistered votes) to get a new election $\mathcal{E}'$.

1. Add two new intervals $C_0 = \{x\}$ and $C_{\omega+1} = \{y\}$ such that C_0 is in the leftmost position of P and $C_{\omega+1}$ is in the rightmost position of P;

2. Replace each vote with preference $\succ$ whose peak is on the left-side (resp. right-side) of C_i with a new vote with preference $\succ'$ such that $x \ \succ' C_i \succ' \ y$ (resp. $y \ \succ' C_i \succ' \ x$), where for every two candidates $a, b \in C_i$ it holds that $a \succ' b$ if and only if $a \succ b$;

3. Replace each vote with preference $\succ$ whose peak is at C_i with a new vote with preference $\succ'$ such that $(\succ (C_i)) \ \succ' \ x \ \succ' \ y$, where for every two candidates $a, b \in C_i$ it holds that $a \succ' b$ if and only if $a \succ b$; and

4. Delete all intervals except C_i, C_0 and $C_{\omega+1}$.

It is clear that the single-peaked width of the resulting election $\mathcal{E}'$ is bounded by $\mathcal{k}$. After applying the reduction rule, each instance contains at most $\mathcal{k}+2$ candidates. If a control problem can be solved in $O(f(m) \cdot |\mathcal{E}|^{O(1)})$ time for m being the number of candidates, then it admits an $O(f(\mathcal{k}) \cdot |\mathcal{E}|^{O(1)})$-time algorithm as well. The correctness of Theorem 3.9 follows.

Now we discuss the destructive case. In this case, we first check whether we can make the distinguished candidate p not in the median interval by adding/deleting at most $\mathcal{R}$ votes. This can be done in polynomial time. If we can do so, the given instance is a yes-instance, and we are done. Otherwise, p will be in the median interval. In this case, we can use the above Reduction Rule to reduce the size of the candidates to $\mathcal{k}+2$. Clearly, if a control problem can be solved in $O(f(m) \cdot |\mathcal{E}|^{O(1)})$ time for m being the number of candidates, then it admits an $O(f(\mathcal{k}) \cdot |\mathcal{E}|^{O(1)})$-time algorithm as well. □

Theorem 3.9 requires that the voting correspondence must pass the Smith-IIA criterion and the considered problems must be $\mathcal{FPT}$ with respect to the number of candidates as the parameter. At first glance, it seems that the conditions, especially the second one, are very restrictive. However, we show several voting correspondences which satisfy both conditions.

The following voting correspondences pass the Smith-IIA criterion [227]: Ranked Pairs, Schulze's, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$, Condorcet, Kemeny, and Slater's. One can modify a voting correspondence φ which does not pass the Smith-IIA criterion to a new one passing the Smith-IIA criterion by restricting the election to the candidates in the Smith set. We use φ-Smith to denote the new correspondence.

Faliszewski et al. [112] showed that Copeland$^\alpha$ satisfy the second condition. In addition, Hemaspaandra, Lavaee and Menton [148] showed both Ranked pairs and Schulze's correspondences satisfy the second condition. Xia[246] and Yang[249] independently prove that many common voting correspondences, including all the above ones and many others, satisfy the second condition. Due to these, we have the following corollary.

Corollary 3.2. *In an odd-votes election, the constructive/desctructive control by adding/deleting votes problem for both the unique-winner and the nonunique-winner models are $\mathcal{FPT}$ with single-peaked width as parameter for the following voting correspondences: Ranked Pairs, Schulze's, Copeland$^\alpha$, Kemeny, Slater's, and φ-Smith, where φ can be many voting correspondences such as a positional scoring correspondence, Bucklin's, Maximin, Nanson's or Baldwin's.*

We remark that Theorem 3.9 can be extended to control by partition of votes problems. The definition of the problems can be found in [112].

3.6 Conclusion

In this chapter, we have studied the (parameterized) complexity of control problems in elections with bounded single-peaked width under the prominent Condorcet, Maximin and Copeland$^\alpha$ voting correspondences. Our main results are summarized in Table 3.1 and Theorem 3.9. In particular, we proved that, with respect to the parameter single-peaked width, the constructive/destructive control by adding/deleting votes under Condorcet, and the destructive control by adding/deleting votes under Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ are $\mathcal{FPT}$. Moreover, we derived a general framework for identifying $\mathcal{FPT}$ control problems under voting correspondences passing the Smith-IIA criterion. In contrast to the $\mathcal{FPT}$-solvability of the destructive control problems, we proved that the constructive control by adding/deleting votes for both Copeland$^\alpha$ and Maximin become $\mathcal{NP}$-hard even in elections with a small constant single-peaked width, implying that these problems cannot be $\mathcal{FPT}$ with respect to single-peaked width. In particular, the constructive control by adding/deleting votes for Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ become $\mathcal{NP}$-hard in elections with single-peaked width 3. In elections with single-peaked 2, the constructive control by adding/deleting votes problems are polynomial-time solvable for both Maximin and Copeland^1, while remains $\mathcal{NP}$-hard for Copeland$^\alpha$ for every $0 \leq \alpha < 1$.

Apart from elections with bounded single-peaked width, many of our results apply to other restricted elections. In the flowing, we discuss these restricted elections.

3.6.1 Single-Crossing Width

Single-crossing domain was first studied by Mirrelees [198] and Roberts [218]. It has been widely studied due to its importance in the area of income redistribution, coalition formation, local public goods and stratification, etc [7, 8, 52, 77, 94]. Intuitively, An election is single-crossing if there is an order of the voters such that for every pair of candidates there is a demarcation line such that all voters in each side have the same preference over these two candidates. The formal definition is as follows.

Single-Crossing. An election $\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V})$ is a single-crossing election if there is an order $\mathcal{L} = (\pi_1, \pi_2, ..., \pi_n)$ of the votes $\Pi_\mathcal{V}$ so that for every pair of candidates a, b there is an $x \in [n]$ so that

(1) all votes π_i with $i \leq x$ have the same preference over a and b; and

(2) all votes π_i with $i > x$ have the same preferences over a and b.

$\succ_1$	$\succ_2$	$\succ_3$	$\succ_4$	$\succ_5$	$\succ_6$	$\succ_7$
1	3	4	4	4	5	5
2	4	3	5	5	4	4
3	5	5	3	3	3	3
4	1	1	1	1	2	2
5	2	2	2	2	1	1

Figure 3.2: An example of a single-crossing election with five candidates $\{1, 2, 3, 4, 5\}$ and seven votes with preferences $\succ_1, \succ_2, ..., \succ_7$, respectively. The left vertical line is the demarcation for the pair $\{3, 5\}$. We can see that all the voters on the left-side of this line prefer 3 to 5 while all the voters on the right-side of the line prefer 5 to 3. The right vertical line is the demarcation for the pair $\{1, 2\}$ and the pair $\{4, 5\}$.

An example of a single-crossing election is shown in Figure 3.2.

Single-Crossing Width. The single-crossing width of an election is defined in a similar way as of single-peaked width. Precisely, the *single-crossing width* of an election is the minimum integer k so that the candidates can be grouped into intervals of size at most k each. Moreover, if we contract every interval the election is single-crossing.

Recently, strategic voting problems in single-crossing elections and elections with bounded single-crossing width were studied [71, 188]. We remark that all our $\mathcal{NP}$-hardness results in this chapter apply to elections with small constant single-crossing width.

Theorem 3.10. *CCAV-φ-UNI, CCAV-φ-NON, CCDV-φ-UNI and CCDV-φ-NON are $\mathcal{NP}$-hard in elections with single-crossing width 3, for φ being Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. Moreover, CCAV-Copeland$^\alpha$-UNI, CCAV-Copeland$^\alpha$-NON, CCDV-Copeland$^\alpha$-UNI and CCDV-Copeland$^\alpha$-NON for every $0 \leq \alpha < 1$ are $\mathcal{NP}$-hard in elections with single-crossing width 2.*

Proof. Check that in all the $\mathcal{NP}$-hardness reductions in this chapter, we create only two types of votes: the votes with peaks at the left-most interval and the votes with peak at the right-most interval. After contracting all intervals, the elections constructed in the reductions are clearly single-crossing. $\square$

Magiera and Faliszewski [188] studied control problems in single-crossing elections recently. They proved that constructive/destructive control by adding/deleting votes/candidates are polynomial-time solvable for Plurality (1-Approval) and Condorcet in single-crossing elections. The above theorem complements their results.

In fact, our reductions apply to further restricted domain: elections with bounded single-peaked and single-crossing width. Elections that is both single-peaked and single-crossing (SPSC for short) has been recently studied by Elkind, Faliszewski and Skowron [95]. The SPSC width of an election is defined as the minimum integer k so that the candidates can be grouped into intervals, and moreover, if all the intervals are contracted, the election is an SPSC election. Since the $\mathcal{NP}$-hardness reductions of the control problems studied in this chapter apply to both elections with bounded single-peaked width and elections with bounded single-crossing width simultaneously, the reductions also apply to elections with bounded SPSC width.

3.6.2 Euclidean Elections

Euclidean domain is another well studied restriction on preferences. In this scenario, both the voters and candidates are mapped to points in a d-dimension Euclidean space. Moreover, each voter ranks the candidates according to the Euclidean distances between the candidates and himself. In particular, a candidate with small Euclidean distance to the voter is ranked higher than a candidate with great Euclidean distance. It is easy to check that 1-dimension Euclidean elections are SPSC elections [140]. According to this fact, the complexity of strategic voting problems in single-peaked elections directly apply to 1-dimension Euclidean elections (see [44, 111, 113] for the results in single-peaked elections). Recently, Elkind, Faliszewski and Skowron [95] showed that 1-dimension Euclidean domain is a proper subset of SPSC domain by showing a voting profile which is SPSC but not 1-dimension Euclidean election.

We mark that many of our results apply to d-dimension Euclidean elections. In the following, we assume that every voter ranks the candidates which have the same Euclidean distance to himself in his own favor.

Theorem 3.11. *CCAV-Copeland$^\alpha$-UNI, CCAV-Copeland$^\alpha$-NON, CCDV-Copeland$^\alpha$-UNI and CCDV-Copeland$^\alpha$-NON for every $0 \leq \alpha < 1$ are $\mathcal{NP}$-hard in d-dimension Euclidean elections for every $d \geq 2$.*

Proof. To check this theorem, recall that in the reduction for CCAV-Copeland$^\alpha$-UNI, CCAV-Copeland$^\alpha$-NON, CCDV-Copeland$^\alpha$-UNI and CCDV-Copeland$^\alpha$-NON (in Theorem 3.2), we created only two types of votes: votes with peaks on the left-most interval and votes with peaks at the right-most interval. Furthermore, each interval contains only two candidates. Therefore, we can map the votes and candidates in 2-dimension Euclidean space, as shown in Figure 3.3. □

Several of our results also apply to 3-dimension Euclidean elections.

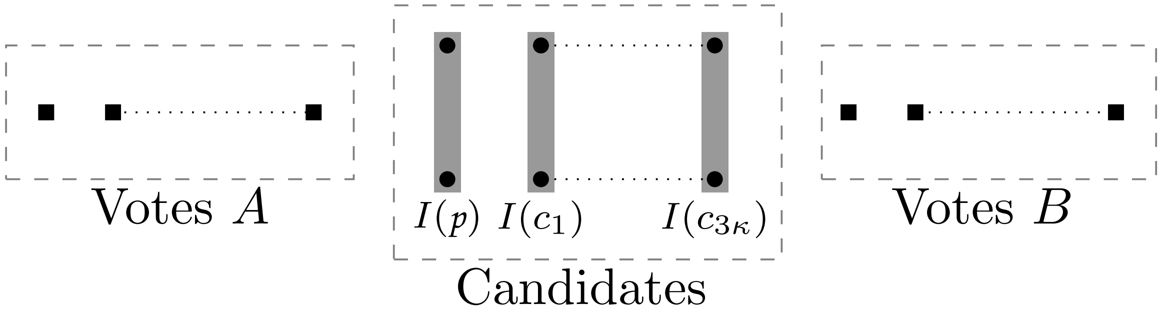


Figure 3.3: An illustration of how to extend the results in Theorem 3.2 to 2-dimension Euclidean elections. Here, "Votes A" are all the votes that have peak at the left-most interval (that is, $I(p)$), and "Votes B" are all the votes that have peak at the right-most interval (that is, $I(c_{3\kappa})$). The order of the votes within "Votes A" and "Votes B" is arbitrary. Moreover, every vote has the same Euclidean distance to both candidates in each interval. Clearly, every vote in "Votes A" ranks the interval $I(p)$ in the highest position, and ranks every $I(c_i)$ above $I(c_j)$ for all $1 \leq i < j \leq 3\kappa$. Furthermore, every vote in "Votes B" ranks the interval $I(c_{3\kappa})$ in the highest position, and ranks every $I(c_i)$ above $I(c_j)$ for all $3\kappa \geq i > j \geq 1$.

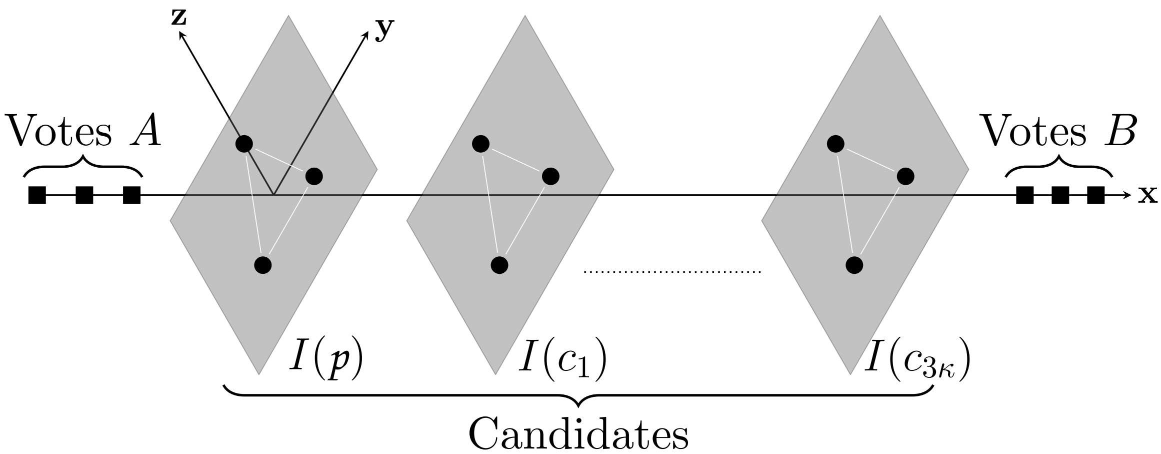


Figure 3.4: An illustration of how to extend the results in Theorems 3.4 and 3.7 to the 3-dimension Euclidean elections. Here, "Votes A" are all the votes that have peak at the left-most interval (that is, $I(p)$), and "Votes B" are all the votes that have peak at the right-most interval (that is, $I(c_{3\kappa})$). The orders of the votes within "Votes A" and "Votes B" are arbitrary. All three candidates in the same interval are mapped on a plain which is perpendicular to the **x**-axis. Moreover, they are mapped to three vertices which form a equilateral triangle. Therefore, all three candidates in the same interval have the same distance to every vote. Clearly, every vote in "Votes A" ranks the interval $I(p)$ in the highest position, and ranks every $I(c_i)$ above $I(c_j)$ for all $1 \leq i < j \leq 3\kappa$. Furthermore, every vote in "Votes B" ranks the interval $I(c_{3\kappa})$ in the highest position, and ranks every $I(c_i)$ above $I(c_j)$ for all $3\kappa \geq i > j \geq 1$.

Theorem 3.12. *CCAV-Copeland$^\alpha$-UNI, CCAV-Copeland$^\alpha$-NON, CCDV-Copeland$^\alpha$-UNI, CCDV-Copeland$^\alpha$-NON for every $0 \leq \alpha \leq 1$, CCAV-Maximin-UNI, CCAV-Maximin-NON, CCDV-Maximin-UNI and CCDV-Maximin-NON are $\mathcal{NP}$-hard in*

d-dimension Euclidean elections for every $d \geq 3$.

Proof. The $\mathcal{NP}$-hardness for Copeland$^{\alpha}$ is from the reductions in Theorem 3.4, and the $\mathcal{NP}$-hardness for Maximin is from the reductions in Theorem 3.7. Figure 3.4 depicted how we map the voters and candidates so that the reductions in Theorems 3.4 and 3.7 can apply to the d-dimension Euclidean elections. □

We remark in the last that Theorems 3.11 and 3.12 rely on the assumption that ties are allowed, in the sense that for a voter there might be more than one candidate which has the same Euclidean distance to himself. Moreover, ties are broken in the voter's favor. It is interesting to investigate whether those $\mathcal{NP}$-hardness results stated in Theorems 3.11 and 3.12 still hold if we discard these two assumptions.

4

Bribery with Restricted Distances

Bribery is another type of strategic behavior that has been widely studied in COMSOC. In this setting, an external agent distributes valuable resources (e.g., money, gifts, shopping cards, politic promises, etc.) to voters, and in return to ask them to recast their votes in his favor. For the external agent who wants to bribe the voters, the complexity of determining whether he can reach his goal (e.g., making a given distinguished candidate win or lose an election) by distributing a limited number of resources is of particular importance. In many real-world settings, the voters who are bribed do not want to deviate too far from their original opinions. We take this natural assumption into account in the study of the bribery problem. In order to measure the similarities between different votes, we adopt several prominent distance concepts such as the Hamming distance and the Kendall-Tau distance.

4.1 Introduction

This chapter is dedicated to the complexity of the distance restricted bribery problems under several prominent voting systems.

4.1.1 Motivation

We have studied control behavior in previous chapters where an external agent has incentives to influence the result of a given election by adding or deleting votes or candidates. In addition to the control settings, there also exist other circumstances where an external agent may alter some of the already submitted votes, or the votes that the voters intend to submit. One example scenario is when a candidate can attempt to change the voter's preferences by running a campaign, which may be targeted at a particular group of voters or in more extreme case where this strategy involves paying voters to change their votes, or bribing election officials to get access to already submitted votes in order to modify them.

In this chapter, we study the model in which an external agent attempts in switching the voter's preferences in his own favor. The external agent's capacity is bounded by a budget constraint. We observe that, while the voter is willing to recast a new vote persuaded by an external agent, he may nevertheless prefer to submit a preference that deviates as little as possible from his true preference. Indeed, if voting is public, he may be worried that switching his preference completely may harm his reputation, yet he will not be caught out if his final preference is sufficiently similar to his true preference. We call this model *distance restricted bribery*. Analogous to the control problems, we distinguish between the *constructive case* and the *destructive case*, the *unique-winner model* and the *nonunique-winner model*. To quantify the amount of deviation of the new recast vote and the original vote of a bribed voter, we use two *distance* measures. Particularly, we consider what is arguably the most prominent distances on votes, namely, the *Hamming distance* (see, e.g., [38, 97, 171, 175, 195] for interesting discussions of Hamming distance in the context of voting) and *Kendall-Tau distance* (see,. e.g., [17, 23, 24, 40] for interesting discussions on Kendall-Tau distance). The definitions of these two distances are in Section 4.1.2. We obtain a broad range of results showing that the complexity of bribery depends closely on the settings. Our results are summarized in Table 4.1.

Related Works. Our model is clearly related to the bribery problems which have been widely studied in COMSOC. Faliszewski, Hemaspaandra and Hemaspaandra [107] introduced the *bribery* problem, where is to decide whether a distinguished candidate can become a winner (constructive) or be prevented from being a winner (destructive) by recasting at most $\mathcal{R}$ (a given integer) votes. In their paper, they also considered the \$bribery where each voter has a price to change its vote. Later Faliszewski [106] proposed a new notion of bribery, which he called *nonuniform bribery* where a voter's price may depend on the nature of changes she is asked to implement. A similar notion called *mictrobribery* was considered in [112]. Elkind, Faliszewski and Slinko [98] introduced the framework of *swap bribery* where the briber can ask a voter to perform a sequence of swaps; each swap changes the relative order of two candidates that are currently adjacent in this voter's preference list. Moreover, each swap may have a different price; and the price of a bribery is the sum of the prices of all swaps that it involves. In the same paper [98], the authors also studied the *shift bribery* problem, which is a restricted variant of swap bribery. In particular, in the shift bribery problem, only swaps involving the distinguished candidate are allowed. Parameterized complexity studies of the swap bribery problem and the shift bribery problem can be found in [50, 84]. Recently, Pini, Rossi and Venable [208] investigated the complexity of bribery in voting with soft constraints, where each candidate is an element of the Cartesian product of the domains of some variables, and agents express their preferences over the candidate via soft constraints. Mattei et al. [190] studied the complexity of bribery in CP-nets.

Our study is highly related to Obraztsova and Elkind's work [205] where a manipulator aims to make a distinguished candidate win or loss the election by casting an untruthful vote. Here, the untruthful vote should be as close as to the truthful vote of the manipulator. They examined this problem for several voting correspondences with the adoption of three prominent distances, namely, the KT-distance, the footrule distance, and the maximum displacement distance. Our model differs from theirs in the following aspects. First, in our settings, at most $\mathcal{R}$ voters might be bribed, however, they considered only one such voter. Second, their problems ask the manipulator to cast an untruthful vote which is as close as possible to the truthful vote. However, we mainly focus on the settings where the bribed voters must cast their votes which have a small constant discrepancy from their original votes.

4.1.2 Preliminaries

In this section, we introduce the distance measurements and formal definitions of the problems concerned in this chapter.

Distance. A *distance* on a space X is a mapping $D : X \times X \mapsto \mathbb{R}$ such that:

(1) $D(v, u) \geq 0$ for every two $v, u \in X$;

(2) $D(v, u) = 0$ if and only if $v = u$;

(3) $D(v,u) = D(u,v)$ for every $v, u \in X$; and

(4) $D(v,u) + D(u,w) \geq D(v,w)$ for every three $v, u, w \in X$.

This chapter mainly focuses on distances over votes, i.e., mappings of the form $D : \mathcal{L}(\mathcal{C}) \times \mathcal{L}(\mathcal{C}) \mapsto \mathbb{R}$, where $\mathcal{L}(\mathcal{C})$ is the set of all linear orders over the candidates in $\mathcal{C}$.

Hamming distance. The Hamming distance, named after Richard Hamming, is initially defined on strings [144]. In particular, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. For example, the Hamming distance between the string "a 1 b b" and the string "a b 1 b" is two since there are two positions (the second and the third positions) where the symbols are different. In the context of Hamming distance in this chapter, we regard each vote as a string with each element being (the name of) a candidate. For example, the vote with preference $a \succ b \succ c \succ d$ will be considered as the string "a b c d". Hence, the *Hamming distance* between every two votes with preferences $\succ_1, \succ_2$, denoted as $D_{HAM}(\succ_1, \succ_2)$, is the Hamming distance between the two strings from the two votes, respectively. In fact, votes (linear orders) over a fixed set of candidates $\mathcal{C}$ can be also considered as permutations over $\mathcal{C}$. Hamming distance on permutations has been widely studied in the literature [193, 226]. We remark that any two different permutations (and thus votes defined as linear orders) has Hamming distance at least two.

Kendall-Tau distance (KT-distance for short). The KT-distance was coined by Maurice Kendall [165]. In particular, it counts the number of pairwise disagreements between two linear orders (votes). In a formal way, the KT-distance between two linear orders $\succ_1$ and $\succ_2$ over a set $\mathcal{C}$ is defined as follows.

$$\mathrm{D}_{KT}(\succ_1, \succ_2) = |\{(a,b) | a, b \in \mathcal{C}, a \succ_1 b \text{ and } b \succ_2 a\}|$$

Equivalently, the KT-distance between two linear orders can be defined as the minimum number of swaps of adjacent candidates needed to transform one into the other [22]. In addition, the KT-distance also turns out to be equal to the number of exchanges needed in a bubble sort (see [6] for an introduction to bubble sort) to convert one full ranking to the other [104]. Due to this fact, the KT-distance is also referred to as *bubble-sort distance* in the literature [35, 61, 104, 105].

Problem Definitions. We mainly study the following problems under different voting correspondences. In the following, let τ be a voting correspondence and "DIST" a distance function. In this chapter, "DIST" can be KT-distance and Hamming distance. For two votes with preferences $\succ_1, \succ_2$ and a distance "DIST", we say these two votes are DIST(d)-close if $D_{DIST}(\succ_1, \succ_2) \leq d$.

Constructive Distance Restricted Bribery under τ (C-DIST(d)-τ-UNI/NON)
Input: An election $(\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, and two integers $\mathcal{R} \geq 0$ and $d \geq 0$. Here, p is not the unique winner/a winner under the voting correspondence τ.
Question: Is it possible to make p the unique winner/a winner by replacing (recasting) at most $\mathcal{R}$ votes, under the voting correspondence τ? Here, a vote can only be replaced with a DIST(d)-close vote.

Destructive Distance Restricted Bribery under τ (D-DIST(d)-τ-UNI/NON)
Input: An election $(\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$, and two integers $\mathcal{R} \geq 0$ and $d \geq 0$. Here, p is the unique winner/a winner under the voting rule τ.
Question: Is it possible to prevent p from being the unique winner/a winner by replacing (recasting) at most $\mathcal{R}$ votes, under the voting correspondence τ? Here, a vote can only be replaced with a DIST(d)-close vote.

We give either polynomial-time algorithms or $\mathcal{NP}$-hardness reductions for the above problems. Our hardness proofs in this chapter are reduced from the XdC problem which is defined as follows.

> Exact d-Set Cover (XdC)
> *Input:* A universal set $U = \{c_1, c_2, ..., c_{d \cdot \kappa}\}$ and a collection $S = \{s_1, s_2, ..., s_m\}$ of d-subsets of U.
> *Question:* Is there an $S' \subseteq S$ such that $|S'| = \kappa$ and each $c_i \in U$ appears in exactly one set of S'?

It is clear that when $d = 3$, we get the X3C problem. In the following, we show the $\mathcal{NP}$-hardness of the XdC problem for every $d \geq 4$.

Lemma 4.1. *XdC is $\mathcal{NP}$-hard for every constant $d \geq 3$.*

Proof. It is well known that the X3C problem is $\mathcal{NP}$-hard [138]. In the following, we show how to reduce from X$(d-1)$C to XdC to prove the $\mathcal{NP}$-hardness of XdC for every $d \geq 4$.

Let (U, S) be an instance of X$(d-1)$C, where $U = \{c_1, c_2, ..., c_{(d-1)\cdot\kappa}\}$ is the universal set and $S = \{s_1, s_2, ..., s_m\}$ is the collection of $(d-1)$-subsets of U. We construct an instance (U', S') of XdC as follows.

Let $W = \{1, 2, ..., \kappa\}$. Then, we set the universal set of XdC as $U' = U \cup W$. Now we construct the subsets in S'. In particular, for each $s_i \in S$ and each $j \in W$, we create a d-subset $s_i^j = s_i \cup \{j\}$ in S'. Therefore, we have in total $|S| \cdot \kappa$ subsets in S'.

Now we show the correctness. Suppose that $\bar{S}$ is an exact set cover of the instance (U, X). Let $(s_{i_1}, s_{i_2}, ..., s_{i_\kappa})$ be any arbitrary but fixed order of $\bar{S}$. Then it is easy to see that $\bar{S}' = \{s^j_{i_j} \mid j = 1, 2, ..., \kappa\}$ is an exact d-set cover of (U', S'). The other direction is analogous.

Since X3C is $\mathcal{NP}$-hard, we can conclude that XdC is $\mathcal{NP}$-hard for every $d \geq 3$. $\square$

A different $\mathcal{NP}$-hardness reduction for the X4C problem can be found in [20, 21]. In particular, the reduction in [20, 21] implies that the X4C problem remains $\mathcal{NP}$-hard even when every element from the universal set occurs in exactly three subsets in the collection. Recall that under the same restriction, the X3C problem is also $\mathcal{NP}$-hard. Moreover, in both problems under this restriction, we have that $|S| = 3\kappa$.

The two words "promote" and "degrade" are often used in $\mathcal{NP}$-hardness reductions and description of polynomial-time algorithms with specific meanings in this chapter. In particular, for a vote π and a candidate c, *promoting* the candidate c by ℓ positions means recast the vote π as follows. First, rank c in the $(pos_\pi(c) - \ell)$-th position. Then, rank every candidate c' with $pos_\pi(c) > pos_\pi(c') \geq pos_\pi(c) - \ell$ in the $(pos_\pi(c') + 1)$-th position. Finally, rank all the remaining candidates in their original positions. See Figure 4.1 for an example.

$$a \succ b \succ c \succ d \succ e \succ f \qquad\qquad a \succ e \succ b \succ c \succ d \succ f$$

Figure 4.1: This figure shows how to recast a vote by promoting the candidate e by three positions. The left-hand is the preference of the original vote and the right-hand is the recast vote after promoting e by three positions.

Degrading the candidate c by ℓ positions means recast the vote π as follows. First, rank c in the $(pos_\pi(c) + \ell)$-th position of π. Then, rank every candidate c' with $pos_\pi(c) < pos_\pi(c') \leq pos_\pi(c) + \ell$ in the $(pos_\pi(c') - 1)$-th position. Finally, rank all the remaining candidates in their original positions. See Figure 4.2 for an example.

$$a \succ b \succ c \succ d \succ e \succ f \qquad\qquad a \succ c \succ d \succ b \succ e \succ f$$

Figure 4.2: This figure shows how to recast a vote by degrading the candidate b by two positions. The left-hand is the preference of the original vote and the right-hand is the recast vote after degrading b by two positions.

It is easy to see that the recast vote obtained from the original vote by promoting or degrading a candidate by ℓ positions has KT-distance ℓ from the original vote.

	General		KT-distance							Hamming distance		
	Const	Dest	Constructive			Destructive				Constructive	Destructive	
			$d = 1, 2$	$d = 3$	$d \geq 4$	$d = 1$	$d = 2$	$d = 3$	$d \geq 4$	$d = 2$	$d = 2$	$d \geq 3$
Borda	$\mathcal{NP}$-h$^\diamondsuit$	$\mathcal{P}$		$\mathcal{NP}$-h		$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$		$\mathcal{P}$	$\mathcal{P}$
Condorcet	$\mathcal{NP}$-h$^\clubsuit$	$\mathcal{P}^\clubsuit$	$\mathcal{P}$	$\mathcal{NP}$-h		$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{P}$	$\mathcal{NP}$-h	$\mathcal{P}$	$\mathcal{P}$
Maximin	$\mathcal{NP}$-h$^\clubsuit$	$\mathcal{NP}$-h$^\clubsuit$			$\mathcal{NP}$-h	$\mathcal{P}$			$\mathcal{NP}$-h	$\mathcal{NP}$-h	$\mathcal{NP}$-h	
Copeland$^\alpha$ $0 \leq \alpha \leq 1$	$\mathcal{NP}$-h$^\heartsuit$	$\mathcal{NP}$-h$^\heartsuit$		$\mathcal{NP}$-h				$\mathcal{NP}$-h UNI: $d \geq 5$		$\mathcal{NP}$-h	$\mathcal{NP}$-h	

Table 4.1: A summary of the complexity of the bribery problems. Here, the general case refers to the bribery problem studied in [107]. In particular, in the general setting each bribed voter can recast a new vote without the distance restriction. The abbreviation "Const" refers to "Constructive" and "Dest" refers to "Destructive". Moreover, "d" is the distance upper bound. Furthermore, "$\mathcal{P}$" stands for "polynomial-time solvable" and "$\mathcal{NP}$-h" stands for "$\mathcal{NP}$-hard". The results for Copeland$^\alpha$ apply to every $0 \leq \alpha \leq 1$. All results shown in this table apply to both the unique-winner model and the nonunique-winner model. The $\mathcal{NP}$-hardness of the destructive bribery for Copeland$^\alpha$ under the unique-winner model only holds for KT-distance upper bound d with $d \geq 5$. The complexity of the problems whose distance bound d is not showed in the table remain open. Notice that every two different votes has Hamming distance at least 2. The result marked by $\diamondsuit$ is from [54], by $\clubsuit$ from [110], and by $\heartsuit$ from [112]. All results for KT-distance and Hamming distance are our new results.

4.2 Kendall-Tau Distance Restricted Bribery

In this section, we investigate the bribery problem with KT-distance restrictions. In the following, we summarize our results in several theorems. We begin with some polynomial-time solvability results.

Theorem 4.1. *All the following problems are polynomial-time solvable: C-KT($\mathscr{d}$)-Condorcet-UNI, C-KT($\mathscr{d}$)-Condorcet-NON for $\mathscr{d} = 1, 2$, D-KT($\mathscr{d}$)-Borda-UNI, D-KT($\mathscr{d}$)-Borda-NON for every possible $\mathscr{d}$, D-KT($\mathscr{d}$)-Condorcet-UNI, D-KT($\mathscr{d}$)-Condorcet-NON for every possible $\mathscr{d}$, D-KT(1)-Maximin-UNI and D-KT(1)-Maximin-NON.*

Proof. We prove this theorem by deriving polynomial-time algorithms for the problems stated in the theorem. For simplicity, the following algorithms are based on the unique-winner model. The algorithms for the nonunique-winner model can be easily adapted from the algorithms for the unique-winner model. In the following, let $\mathcal{E} = (\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}})$ be the given election, where p is the distinguished candidate. Moreover, let m be the number of candidates and n the number of votes, that is, $m = |\mathcal{C} \cup \{p\}|$ and $n = |\Pi_{\mathcal{V}}|$.

C-KT($\mathscr{d}$)-Condorcet-UNI for $\mathscr{d} = 1, 2$. The C-KT(1)-Condorcet-UNI can be easily solved with the following greedy algorithm: for each candidate c which is not beaten by p, recast up to $\mathscr{R}$ votes (we also adjust the value of $\mathscr{R}$ as $\mathscr{R} := \mathscr{R} - x$, where x is the number of votes that are recast for c) which rank c immediately above p by swapping p and c until p beats c. If p becomes the Condorcet winner after doing so, return “Yes”; otherwise, return “No”. To solve C-KT(2)-Condorcet-UNI, we reduce the problem to the Simple b-Edge Cover of Multigraphs problem which is polynomial-time solvable [180]. The definition of the problem is as follows.

> Simple b-Edge Cover of Multigraphs
> *Input:* An undirected multigraph $G = (U, E)$ where U is the set of vertices and E is the set of edges, a function $f : U \rightarrow \mathbb{Z}^+$ and a positive integer κ.
> *Question:* Does there exist a subset of at most κ edges $E' \subseteq E$ such that every vertex $u \in U$ is incidents to at least $f(u)$ edges in E'?

Now we show how to reduce C-KT(2)-Condorcet-UNI to the Simple b-Edge Cover of Multigraphs problem. For each candidate c which is not beaten by p, we create a vertex. For simplicity, we still use c to denote the vertex corresponding to the candidate c. We define $\overleftarrow{p}_v$ for a vote π_v where p is not ranked in the top as follows: if p is not ranked in the top 2 positions in π_v, then $\overleftarrow{p}_v$ is the set containing the two candidates which are ranked immediately above p in π_v; if p is ranked in the second-highest position in π_v, then $\overleftarrow{p}_v$ is the set containing the candidate that ranked in the highest position in π_v. For example, for a vote π_v with preference $a \succ b \succ c \succ p \succ d$, $\overleftarrow{p}_v = \{b, c\}$, while for a vote π_u with preference $a \succ p \succ c \succ b \succ d$, $\overleftarrow{p}_u = \{a\}$. The edges are created according to the votes.

Precisely, for each vote π_v with $|\overleftarrow{p}_v| = 2$, if both candidates of $\overleftarrow{p}_v = \{c, c'\}$ are not beaten by p, we create an edge between c and c'. On the other hand, if only one of $\overleftarrow{p}_v$ is not beaten by p, we introduce a new degree-1 vertex adjacent to the vertex in $\overleftarrow{p}_v$ that is not beaten by p. For each vote π_v with $|\overleftarrow{p}_v| = 1$, if the candidate in $\overleftarrow{p}_v$ is not beaten by p, we introduce a new degree-1 vertex adjacent to the candidate in $\overleftarrow{p}_v$.

Now we come to the capacities of the vertices. Each vertex corresponding to a candidate c has a capacity $f(c) = (N(c,p)-N(p,c))/2+1$ whenever $N(c,p)-N(p,c) \equiv 0 \mod 2$, and has a capacity $f(c) = (N(c,p)+1-N(p,c))/2$ otherwise. Moreover, each newly introduced degree-1 vertex has capacity 0. The value of the capacity $f(c)$ indicates the minimum number of votes which rank c to p, that are needed to be replaced with votes which rank p above c in order to make p beat c.

Now we get an instance of the Simple b-Edge Cover of Multigraphs problem which is solvable in polynomial time [180]. Moreover, given a solution E' of Simple b-Edge Cover of Multigraphs, we can get a solution for C-KT(2)-Condorcet in polynomial time. In particular, we recast the votes according to the edges in E': if there is an edge $(c, c') \in E'$ where none of $\{c, c'\}$ is a newly introduced degree-1 vertex, then we recast the corresponding vote by promoting p by two positions; if there is an edge $(c, c') \in E'$ where one of $\{c, c'\}$ is a newly introduced degree-1 vertex, we recast the corresponding vote by promoting p by one position.

D-KT($\mathscr{d}$)-Condorcet. The algorithm first guesses a candidate p' which is not beaten by p in the final election. This leads to at most m subinstances, each determining whether we can replace at most $\mathscr{R}$ votes with $\mathscr{R}$ many KT($\mathscr{d}$)-close new votes so that p' is not beaten by p. Now we focus on solving each subinstance. Observe first that we will not replace any vote which has ranked p' above p. Moreover, due to the distance restriction, for a vote that ranks p above p', if there are more than $\mathscr{d}-1$ candidates ranked between them, we will not replace this vote. Let A be the multiset of the votes which rank p above p', and where there are no more than $\mathscr{d}-1$ candidates ranked between them. Let $n' = |A|$. If $\min\{\mathscr{R}, n'\} + N(p', p) \geq \frac{n}{2}$, we terminate the algorithm and return "Yes", since we can get a solution by the following way: arbitrarily choose $\min\{\mathscr{R}, n'\}$ votes in A and replace each of them with a new vote obtained from the original vote by promoting p' to the position immediately above p. On the other hand, if $\min\{\mathscr{R}, n'\} + N(p', p) < \frac{n}{2}$, we cannot make p' beat or tie p by replacing at most $\mathscr{R}$ votes; and thus, in this case we discard the current subinstance and proceed to the next one. If none of the subinstances leads to a "Yes" answer, then return "No".

D-KT(1)-Maximin: The algorithm first carries out a polynomial number of guesses. In particular, the algorithm guesses a candidate p' which prevents p from being the unique winner, an integer s which plays the role as an upper bound of the

Maximin score of p in the final election and a lower bound of the Maximin score of p' in the final election, and a candidate q with $N(p,q) \leq s$ in the final election. These lead to at most $m^2 \times n$ subinstances where m is the number of candidates and n is the number of votes. To make it clear, we give the formal definition of the subproblem.

> Sub-D-KT(1)-Maximin
> *Input:* An election $\mathcal{E} = \{\mathcal{C} \cup \{p, p', q\}, \Pi_{\mathcal{V}}\}$, and two integers s and $\mathcal{R}$.
> *Question:* Is there a submultiset $\Pi_{\mathcal{T}} \sqsubseteq \Pi_{\mathcal{V}}$ of votes such that
>
> (1) $\Pi_{\mathcal{T}}$ contains at most $\mathcal{R}$ votes; and
>
> (2) we can replace every vote $\pi_v \in \Pi_{\mathcal{T}}$ with a new vote obtained from π_v by swapping two consecutively ranked candidates so that $N(p,q) \leq s$ in the final election, and the Maximin score of p' is at least s in the final election?

Now we focus on solving the subproblem. Let Π_p be the multiset of votes which rank p immediately above q. Let $A = \{c \in \mathcal{C} \mid N_{\mathcal{E}}(p', c) < s\}$. For each $c \in A$, let $\Pi_c \sqsubseteq \Pi_{\mathcal{V}} \uplus \Pi_p$ be the multiset of votes that rank c immediately above p'. Clearly, for every two candidates $c, c' \in A$, $\Pi_c \cap \Pi_{c'} = \emptyset$. Moreover, for every $c \in A$, let $f(c) = s - N_{\mathcal{E}}(p', c)$. The algorithm works as follows. For each $c \in A$, arbitrarily choose $\min\{f(c), |\Pi_c|\}$ votes in Π_c, and replace each of them with a new vote obtained from the original vote by swapping c and p'; then, set $f(c) := f(c) - \min\{f(c), |\Pi_c|\}$ and $\mathcal{R} := \mathcal{R} - \min\{f(c), |\Pi_c|\}$. If $\mathcal{R} < 0$ after doing so, we cannot make p' have a Maximin score at least s by replaying at most $\mathcal{R}$ votes; and thus, the algorithm returns "No". Otherwise, let $B = \{c \in A \mid f(c) > 0\}$. Then, for each $c \in B$, let $\bar{\Pi}_c$ be the multiset of votes in Π_p that rank c immediately above p'. If $|\bar{\Pi}_c| < f(c)$, the given instance is a no-instance (since we cannot make p' have a Maximin score at least s in the final election); and thus, we return "No". Otherwise, we arbitrarily choose $\min\{f(c), |\bar{\Pi}_c|\}$ votes in $\bar{\Pi}_c$, and

(1) replace each of them with a new vote obtained from the original vote by swapping c and p';

(2) remove them from the multiset Π_p; and

(3) set $\mathcal{R} := \mathcal{R} - \min\{f(c), |\bar{\Pi}_c|\}$.

Again, if $\mathcal{R} < 0$ after doing so, we return "No". Otherwise, if $\min\{|\Pi_p|, \mathcal{R}\} < N_{\mathcal{E}}(p,q) - s$, the given instance is a no-instance, and we return "No". Otherwise, we return "Yes" since we can get a solution by replacing arbitrary $N_{\mathcal{E}}(p,q) - s$ votes in Π_p by new votes obtained from the original votes by swapping p and q.

D-KT(d)-Borda. The algorithm first guesses a candidate p' which prevents p from being the unique winner. This leads to at most m subinstances, each determining whether we can make p' have a no less score than that of p by replacing at most $\mathcal{R}$ votes with $\mathcal{R}$ many KT(d)-close new votes. We focus on the subinstances. It is clear that for any vote, promoting p' by one position has the same effect as degrading p by one position, in the sense that both cases increase the score gap between p' and p by one. Therefore, the algorithm can choose up to $\mathcal{R}$ specific votes and replace them with new votes which are obtained from the original votes by promoting p' or degrading p by a total number of at most d positions. Precisely, we first order the votes π_v according to the non-increasing order of $pos_v(p') - pos_v(p)$. Then, we choose the first $\mathcal{R}$ votes. Then, for every π_v of the chosen votes, we replace it with a KT(d)-close vote obtained from π_v by promoting p' by $h = \min\{d, pos_v(p') - 1\}$ positions, and degrade p by $\min\{d - h, m - pos_v(p)\}$ positions. After all these replacements, if p is no longer the unique winner, then we return "Yes". Otherwise, we discard the current subinstance and proceed to the next one. If none of the subinstances leads to a "Yes" answer, we terminate the algorithm and return "No". The above algorithm applies to every natural integer d. □

Now we discuss the $\mathcal{NP}$-hardness results. We first investigate the constructive distance restricted bribery for Borda. We have seen from the above theorem that the destructive counterpart turned out to be polynomial-time solvable for every possible values of d. The following theorem shows, however, that the constructive distance restricted bribery for Borda is $\mathcal{NP}$-hard even when the distance is bounded by a small constant. Before proceeding further, we define some notations which will be used in this chapter.

For an order $X = (x_1, x_2, ..., x_i)$ over the set $\{x_1, x_2, ..., x_i\}$, we denote by $\overleftarrow{X}$ the reverse order of X, that is, $\overleftarrow{X} = (x_i, ..., x_2, x_1)$. For a subset $Y \subseteq \{x_1, x_2, ..., x_i\}$, $X \setminus Y$ is the order obtained from X by deleting all the elements in Y. For example, for $X = (1, 4, 3, 8, 5)$ and $Y = \{4, 8\}$, $X \setminus Y = (1, 3, 5)$.

For two candidate subsets X and Y and a vote with preference $\succ$, $X \succ Y$ means that every candidate in X is ranked above every candidate in Y in the vote.

Theorem 4.2. *C-KT(d)-Borda-UNI and C-KT(d)-Borda-NON are $\mathcal{NP}$-hard, for every $d \geq 3$.*

Proof. We first consider C-KT(3)-Borda-NON. The reduction is from X3C. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_m\})$ of X3C, we create an instance $\mathcal{E}$ for C-KT(3)-Borda-NON as follows.

Candidates: For each $c \in U$, we create a corresponding candidate. For convenience, we still use c to denote the corresponding candidate. We create $6m - 6$ dummy candidates $Y = \{y_1, y_2, ..., y_{6m-6}\}$ each of which has considerably less Borda score than that of any other candidate not in Y. For ease of exposition, we divide the dummy candidates into subsets $Z_1, Z_2, ..., Z_m$. To be precise, for each $i = 1, 2, ..., m-2$, $Z_i = \{y_{6i-5}, y_{6i-4}, y_{6i-3}, y_{6i-2}, y_{6i-1}, y_{6i}\}$. Moreover, $Z_{m-1} = \{y_{6m-11}, y_{6m-10}, y_{6m-9}\}$ and $Z_m = \{y_{6m-8}, y_{6m-7}, y_{6m-6}\}$. Finally, we create a distinguished candidate p. In summary, the candidate set is $U \cup \{p\} \cup Y$, where $Y = \cup_{i=1,2,...,m} Z_i$.

Votes: We create $2m+2$ votes in total. In the following, we do not distinguish between the terms "set" and "order". Thus, U is also considered as an order $(c_1, c_2, ..., c_{3\kappa})$, and every $s = \{x_i, x_j, x_k\} \in S$ is considered as an order (x_i, x_j, x_k) with $i < j < k$. In the following votes, the candidates in each Z_i are ranked arbitrarily.

For each $s_j \in S$ with $j = 1, 2, ..., m-2$, we create two votes as follows (notation of the vote: preference of the vote).

$\pi_{s_j} : s_j \succ p \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$

$\pi'_{s_j} : \overleftarrow{U \setminus s_j} \succ Z_j \succ p \succ \overleftarrow{s_j} \succ Y \setminus Z_j$

Note that with the above $2(m-2)$ votes, all candidates in $U \cup \{p\}$ have the same Borda score. The following four votes are created according to the last two 3-subsets $s_{m-1}, s_m \in S$.

$\pi_{s_{m-1}} : s_{m-1} \succ p \succ Z_m \cup Z_{m-1} \succ U \setminus s_{m-1} \succ Y \setminus (Z_m \cup Z_{m-1})$

$\pi'_{s_{m-1}} : \overleftarrow{U \setminus s_{m-1}} \succ Z_{m-1} \succ p \succ Z_m \succ \overleftarrow{s_{m-1}} \succ Y \setminus (Z_m \cup Z_{m-1})$

$\pi_{s_m} : s_m \succ p \succ Z_m \cup Z_{m-1} \succ U \setminus s_m \succ Y \setminus (Z_m \cup Z_{m-1})$

$\pi'_{s_m} : \overleftarrow{U \setminus s_m} \succ Z_{m-1} \succ p \succ Z_m \succ \overleftarrow{s_m} \succ Y \setminus (Z_m \cup Z_{m-1})$

With the above two votes and the previously created $2(m-2)$ votes, p has exactly 6 more points than every candidate $c \in U$.

Finally, we have two votes defined as follows.

$U \succ Z_m \succ p \succ Y \setminus Z_m$

$\overleftarrow{U} \succ Z_m \succ p \succ Y \setminus Z_m$

With all the $2m+2$ votes created as above, p has exactly $3\kappa + 1$ less points than every candidate $c \in U$.

Number of Replaced Votes: $\mathcal{R} = \kappa$

In the following, let $A = \{\pi'_{s_j} \mid s_j \in S\}$ and B the set of the last two created votes. Now we discuss the correctness of the reduction. Let's first check the score gap between every two candidates. As discussed above, in the election, all candidates in U have the same Borda score. Moreover, p has exactly $3\kappa + 1$ less points than every $c \in U$. Finally, every dummy candidate y has a considerably smaller Borda score than every other nondummy candidate.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_{s_j} \mid s_j \in S'\}$ be the multiset of the votes of the first type corresponding to S'. Every vote in $\Pi_{S'}$ ranks the three candidates corresponding to a 3-subset s_j above p in π_{s_j}. Consider the election $\mathcal{E}'$ obtained from the original election $\mathcal{E}$ by replacing each $\pi_{s_j} \in_+ \Pi_{S'}$ with a vote obtained from π_{s_j} by promoting p to the highest position.

Precisely, for each $\pi_{s_j} \in \Pi_{S'}$ defined as $s_j \succ p \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$, we replace it with a vote defined as $p \succ s_j \succ Z_j \succ U \setminus s_j \succ Y \setminus Z_j$. Clearly, each replacement increases the score of p by 3, and decreases the score of every candidate in s_j by 1. Since there are exactly κ votes in $\Pi_{S'}$, the score of p is finally increased by 3κ. Since S' is an exact 3-set cover, for every $c \in U$, there is only one vote in $\Pi_{S'}$ which ranks c above p. Therefore, all replacements decrease the score of each candidate in U by 1. Since p has exactly $3\kappa + 1$ less points than every candidate $c \in U$ in the original election $\mathcal{E}$, p has exactly the same score as every candidate $c \in U$ in the final election $\mathcal{E}'$. Therefore, p becomes a winner in $\mathcal{E}'$.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. We assume that $\Pi_{S'}$ does not contain any vote in $A \cup B$. This assumption is sound due to the following lemma.

Lemma 4.2. *If $\mathcal{E}$ is a yes-instance, there must be a solution wherein no vote in $A \cup B$ is replaced.*

Proof. We prove the claim by showing that it is always better to replace a vote not in $A \cup B$ than to replace a vote in $A \cup B$. Suppose that π is a vote in $A \cup B$ that is replaced. Observe that promoting p is always better than degrading candidates in U, since promoting p by one position decreases the score gap between every candidate in U and p by one, while degrading some candidate $c \in U$ by one position only decreases the score gap between c and p by one (sometimes even increases the score gap between some other candidate $c' \in U$ and p). Moreover, the amount of points that can be decreased in the score gap between every candidate in U and p by promoting p in π, can be also achieved by promoting p in any vote that is not in $A \cup B$. In fact, since in every vote in $A \cup B$ there is at least three dummy candidates ranked below some candidates in U but ranked above p, replacing votes which are not in $A \cup B$ can always do better: replacing a vote $\pi_s \notin A \cup B$ with preference $c_x \succ c_y \succ c_z \succ p$... (where $s = \{c_x, c_y, c_z\}$) with a vote with preference $p \succ c_x \succ c_y \succ c_z$... does not only decrease the score gap between every candidate in $U \setminus s$ and p by 3, but better yet, decreases the score gap between every candidate in s and p by 4. $\square$

Due to the above analysis, we assume that $\Pi_{S'}$ contains only the votes in $\{\pi_{s_j} \mid s_j \in S\}$, where S is the collection of 3-subsets in $\mathcal{F}$. Let $S' = \{s_j \mid \pi_{s_j} \in \Pi_{S'}\}$ be the subcollection corresponding to $\Pi_{S'}$. First observe that for any vote $\pi_s \in \Pi_{S'}$, promoting p by three positions is always better than any other combinations: by doing so, the score gap between every candidate in U and p is decreased by at least 3 (for candidates in s, the score gaps decrease by 4). Therefore, we can assume that in the solution, every vote in $\Pi_{S'}$ is replaced with a new vote obtained from the original vote by promoting p by three positions. Since p has $3\kappa + 1$ less points than every candidate in U in the original election $\mathcal{E}$, and we can replace at most κ votes, every candidate in U must be degraded by one position at least once. This implies that for every $c \in U$, there must be a vote $\pi_s \in \Pi_{S'}$ with $c \in s$, further implying that S' is an exact 3-set cover of $\mathcal{F}$.

Now we consider C-KT(3)-Borda-UNI. The reduction is similar to the above one for C-KT(3)-NON, with the difference in the last created vote. Precisely, we remove the last vote created in the reduction for C-KT(3)-UNI, and instead, we create a vote defined as follows.

$$U \succ Z_m \cup \{y_{6m-12}\} \succ p \succ Y \setminus Z_m \cup \{y_{6m-12}\}.$$

By ranking the candidate y_{6m-12} between Z_m and p, the score gap between every candidate in U and p decreases to 3κ, one point less than that in the reduction for C-KT(3)-Borda-NON. This ensures the correctness.

The $\mathcal{NP}$-hardness of C-KT(d)-Borda-UNI and C-KT(d)-Borda-NON for every $d \geq 4$ can be proved via reductions from XdC which is $\mathcal{NP}$-hard as shown in Lemma 4.1. The reductions are analogous to C-KT(3)-Borda-UNI and C-KT(3)-Borda-NON, respectively (in both reductions, we need to add further $2(d-3)$ dummy candidates to each Z_i). □

Now we come to Condorcet. The C-KT(d)-Condorcet problem is related to Dodgson voting. The Dodgson correspondence was introduced by Charles Lutwidge Dodgson who is better known as Lewis Carroll [80]. In this setting, each candidate has a Dodgson score which is defined as the minimum number of swaps of adjacent candidates needed to make the candidate the Condorcet winner. Calculating the Dodgson score of a candidate is proved $\mathcal{NP}$-hard [145]. Recall that the KT-distance between two votes is equal to the minimum number of swaps of adjacent candidates needed to transform one into the other. Therefore, if a candidate can become the Condorcet winner by recasting at most R votes with respect to KT-distance upper bound d, then the Dodgson score of the candidate is at most $R \cdot d$. In Theorem 4.1, we have shown that both C-KT(1)-Condorcet and C-KT(2)-Condorcet are polynomial-time solvable. In the following, we show that the polynomial-time solvability does not hold for C-KT(d)-Condorcet for every $d \geq 3$. Recall that in the general case, the constructive bribery for Condorcet is $\mathcal{NP}$-hard [110, 112].

In the following, we assume that in both the X3C problem and the X4C problem, each element c_i in the universal set occurs in exactly three subsets of the collection $\mathcal{S}$. This assumption does not change the $\mathcal{NP}$-hardness of both problems [20, 21, 138]. Note that under this assumption, we have $n = 3\kappa$ in both the X3C problem and the X4C problem.

Theorem 4.3. *C-KT(d)-Condorcet-UNI and C-KT(d)-Condorcet-NON are $\mathcal{NP}$-hard for every $d \geq 3$.*

	p	c_j	y_1	y_2	y_3
p	-	$3\kappa - 3$	3κ		
c_i	$3\kappa - 2$	$\cdots$	$6\kappa - 5$		
y_1	$3\kappa - 5$	0	-	$3\kappa - 5$	
y_2	$3\kappa - 5$	0	0	-	$3\kappa - 5$
y_3	$3\kappa - 5$	0	0	0	-

Table 4.2: Comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for C-KT(3)-Condorcet-UNI in Theorem 4.3. There are $6\kappa - 5$ votes in total. Entries with '$\cdots$' signifies that the comparison does no affect the correctness of the reduction.

Proof. We first consider C-KT(3)-Condorcet-UNI. The reduction is from the X3C problem. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_{3\kappa})$ of X3C, we create an instance $\mathcal{E}$ for C-KT(3)-Condorcet-UNI as follows:

Candidates: For each $c \in U$, we create a corresponding candidate. For simplicity, we still use the same notation c to denote this candidate. In addition, we have a distinguished candidate p and three dummy candidates $Y = \{y_1, y_2, y_3\}$.

Votes: For each $s = \{c_i, c_j, c_k\} \in S$, we create a vote π_s defined as $s \succ p \succ U \setminus s \succ Y$. Here, the candidates in s, in $U \setminus s$ and in Y are ranked according to the increasing order of the indices. In addition, we create $3\kappa - 5$ votes defined as $U \succ Y \succ p$. Here, the candidates in U and in Y are ranked according to the increasing order of the indices. In total, we have $6\kappa - 5$ votes.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we discuss the correctness. First observe that c_1 is the current Condorcet winner, and no candidate in Y can become the Condorcet winner by replacing at most κ votes with respect to the distance restriction. The comparisons between every two candidates are shown in Table 4.2.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_{s_j} \mid s_j \in S'\}$ be the multiset of votes corresponding to S'. Consider replacing each vote $\pi_s \in \Pi_{S'}$ by another vote which is obtained from π_s by promoting p to the highest position, that is, replacing each vote $\pi_s \in \Pi_{S'}$ defined as $s \succ p \succ U \setminus s \succ Y$ with a vote defined as $p \succ s \succ U \setminus s \succ Y$. Since s is a 3-subset, the KT-distance between the original vote and the new vote is 3. Since S' is an exact 3-set cover, for every $c \in U$ there is exactly one vote in $\Pi_{S'}$ which ranks c above p (and p is ranked above c after the replacement). Therefore, after κ replacements as discussed above, for every $c \in U$, there are exactly $3\kappa - 2$ votes ranking p above c, implying that p is the Condorcet winner in the final election.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. Since $|Y| = 3$ and each vote can be replaced only with a vote which has KT-distance at most 3 from it, replacing any of the last $3\kappa - 5$ votes does not help improving the wining status of p (In other words, replacing a vote in the last $3\kappa - 5$ vote is not helpful for p to beat any candidate in U, since the dummy candidates in Y are ranked between U and p; and thus, according to the distance restriction, p cannot be ranked above any candidate in U via a replacement of a vote in the last $3\kappa - 5$ votes.). Therefore, we know that $\Pi_{S'}$ contains only the votes corresponding to S. Let $S' = \{s \in S \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of S corresponding to S'. In order to make p the Condorcet winner, for every $c \in U$ there must be at least one vote, corresponding to some s with $c \in s$, which is replaced with a vote ranking p above c. This implies that S' is an exact 3-set cover of $\mathcal{F}$.

The above reduction directly applies to C-KT(3)-Condorcet-NON. This is because that we created an odd number of votes in the above reduction; and thus, ties do not occur in comparison between every two candidates, implying that the Condorcet winner and the weak Condorcet winner coincide.

The $\mathcal{NP}$-hardness of C-KT(4)-Condorcet-UNI and C-KT(4)-Condorcet-NON can be proved via reductions from X4C (exact 4-set cover) which is $\mathcal{NP}$-hard [20, 21]. The reductions are analogous to C-KT(3)-Condorcet-UNI and C-KT(3)-Condorcet-NON, respectively (in reductions for C-KT(4)-Condorcet-UNI and C-KT(4)-Condorcet-NON, we need to create one more dummy candidate y_4 and add it to Y).

The $\mathcal{NP}$-hardness of C-KT(d)-Condorcet-UNI and C-KT(d)-Condorcet-NON for every $d \geq 5$ is implied by the $\mathcal{NP}$-hardness reduction in Theorem 3.2 in [112]. □

Now we come to Copeland$^{\alpha}$.

Theorem 4.4. *C-KT(d)-Copeland$^{\alpha}$-UNI and C-KT(d)-Copeland$^{\alpha}$-NON, D-KT(d)-Copeland$^{\alpha}$-NON for every $d \geq 3$, and D-KT(d)-Copeland$^{\alpha}$-UNI for every $d \geq 5$ are $\mathcal{NP}$-hard for every $0 \leq \alpha \leq 1$.*

Proof. We first consider C-KT(3)-Copeland$^{\alpha}$-UNI. The reduction is from the X3C problem with the restriction that every element of the universal set occurs in exactly three subsets in the collection. For an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_{3\kappa}\})$ of X3C where each $c_i \in U$ occurs in exactly three 3-subsets of S, we create an instance $\mathcal{E}$ for C-KT(3)-Copeland$^{\alpha}$-UNI as follows.

Candidates: We have $|U| + 8$ candidates in total. In particular, for each $c_i \in U$, we create a candidate. For simplicity, we still use c_i to denote this candidate. In addition, we have 8 candidates $p, y, Z = \{z_1, z_2, z_3\}$ and $Z' = \{z'_1, z'_2, z'_3\}$, where p is the distinguished candidate.

Votes: Let $n = |S| = 3\kappa$. We create $2n + 1$ votes in total. In particular, for each $s = \{c_i, c_j, c_k\} \in S$, we create one vote π_s defined as $y \succ Z' \succ s \succ p \succ U \setminus s \succ Z$.

Here, the candidates in $Z, Z', s, U \setminus s$ are ranked according to the increasing order of the indices, respectively. In addition, we create $n-2$ votes each defined as $U \succ Z \succ p \succ y \succ Z'$. Finally, we create 3 votes each defined as $p \succ y \succ Z' \succ U \succ Z$. In the above $n+1$ votes, the candidates in U, Z and Z' are ranked according to the increasing order of the indices. It is easy to verify that the candidate y is the current (unique) winner. The comparisons between every two candidates are shown in Table 4.3.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

	p	y	c_i	z_i	z'_i
p	-	$n+1$	n	$n+3$	$n+1$
y	n	-	$n+3$	$n+3$	$2n+1$
c_i	$n+1$	$n-2$	...	$2n+1$	$n-2$
z_i	$n-2$	$n-2$	0	...	$n-2$
z'_i	n	0	$n+3$	$n+3$	...

Table 4.3: The comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for C-KT(3)-Copeland$^\alpha$-UNI in Theorem 4.4. The distinguished candidate is p. The current winner is y. The comparisons corresponding to entries filled with "..." mean that the comparisons do not play any role in the correctness argument.

Now we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{E}$ is a yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S'. Consider the election after replacing all the votes in $\Pi_{S'}$ in the following way: each vote $\pi_s \in \Pi_{S'}$ with $s \in S'$ is replaced with a vote defined as $y \succ Z' \succ p \succ s \succ X \setminus s \succ Z$. Clearly, the KT-distance between these two votes is 3. Since S' is an exact 3-set cover, for each $c_i \in U$ there is exactly one vote $\pi_s \in \Pi_{S'}$ with $c_i \in s$. Due to the construction, c_i is ranked above p in π_s, while ranked below p in the new vote which replaces π_s. Therefore, after κ replacements as discussed above, for every $c_i \in U$ there are $n+1$ votes which rank p above c_i, implying that p beats every candidate $c_i \in U$, further implying that p is the unique Copeland$^\alpha$ winner (holds for every $0 \leq \alpha \leq 1$).

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and $\Pi_{S'}$ is the multiset of votes which are replaced. Let $\mathcal{E}'$ be the final election obtained form $\mathcal{E}$ by replacing the votes in $\Pi_{S'}$ with κ many new votes (we discuss later what are the new votes). Observe that the candidate y beats every other candidate except p in $\mathcal{E}$. A deeper observation is that y still beats those candidates in the final election $\mathcal{E}'$.

Lemma 4.3. *The candidate y beats all the candidates in $U \cup Z \cup Z'$ in $\mathcal{E}'$.*

Proof. Clearly, y beats every candidate in Z' in the final election $\mathcal{E}'$ since all votes rank y above Z'. Now we consider the candidates in $U \cup Z$. Observe first that every vote in $\mathcal{E}$ either ranks y above all candidates in $U \cup Z$, or ranks all candidates in $U \cup Z$ above y. Moreover, the votes that rank y above all candidates in $U \cup Z$ are those that corresponding to S, and the last three created votes. However, in these votes, the candidates in Z' ($|Z'| = 3$) are ranked between y and every candidate in $U \cup Z$; thus, we cannot replace a vote which ranks y above a candidate $a \in U \cup Z$ by a 3-KT-close vote which, however, ranks a above y. Therefore, the votes which rank y above a candidate $a \in U \cup Z$ will still rank y above a in the final election $\mathcal{E}'$. The lemma follows. □

Due to the above lemma and the fact that p is the unique winner in the final election $\mathcal{E}'$, we know that p beats every other candidate in $\mathcal{E}'$. Observe that in the original election $\mathcal{E}$, p is beaten by every candidate in U. Then, due to the distance restriction, $\Pi_{S'}$ must be from the votes corresponding to S. Let $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of 3-subsets corresponding to $\Pi_{S'}$. Since p beats all candidates in U in the final election $\mathcal{E}'$ and we can replace at most $\mathcal{R} = \kappa$ votes, for each c_i there must be a vote $\pi_s \in \Pi_{S'}$ with $c_i \in s$, which is replaced with a new vote obtained from π_s by promoting p by three positions. Since $|S| = 3\kappa$, it follows that S' must be an exact 3-set cover.

The above reduction applies to D-KT(3)-Copeland$^\alpha$-NON if we set y as the distinguished candidate. To check the correctness, observe that the candidate p is the only candidate which can have a higher score than that of y by replacing at most κ votes with κ many KT(3)-close votes: due to Lemma 4.3, a candidate which has a higher score than that of y in the final election has to beat every other candidate. Since y beats every candidate in $U \cup Z \cup Z'$ in the final election (due to Lemma 4.3), no candidate in $U \cup Z \cup Z'$ can have a higher score than that of y in the final election.

Now we consider C-KT(3)-Copeland$^\alpha$-NON. The above reduction does not apply here since in C-KT(3)-Copeland$^\alpha$-NON, p does not need to beat every other candidate in the final election to become a winner (p could also become a winner even there is no exact 3-set cover). In order to overcome this situation, we introduce a new dummy candidate y' which beats p, but is beaten by y in the original election. To this end, we adopt the votes constructed as above together with the newly introduced candidate y'. In particular, we rank y' immediately after y in all the votes corresponding to S and all the followed $n-2$ votes. Moreover, we rank y' above p in all the three votes created in the last. The comparisons between every two candidates are summarized in Table 4.4. Provided with the above votes, we have the following lemma.

Lemma 4.4. *If p is a winner in the final election, then p is beaten by y' in the final election.*

	p	y	y'	c_i	z_i	z'_i
p	-	$n+1$	$n-2$	n	$n+3$	$n+1$
y	n	-	$2n-2$	$n+3$	$n+3$	$2n+1$
y'	$n+3$	3	-	$n+3$	$n+3$	$2n+1$
c_i	$n+1$	$n-2$	$n-2$	-	$2n+1$	$n-2$
z_i	$n-2$	$n-2$	$n-2$	0	-	$n-2$
z'_i	n	0	0	$n+3$	$n+3$	-

Table 4.4: The comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for C-KT(3)-Copeland$^\alpha$-NON in Theorem 4.4. The distinguished candidate is p. The current winners are y and y'.

Proof. We prove the lemma by contradiction. Suppose that p becomes a winner, but is not beaten by y' in the final election. Then, we have to replace the last three votes with three new votes obtained from the original votes by swapping the positions of y' and p. Therefore, at most $\kappa - 3$ votes corresponding to S can be replaced, implying that p would be beaten by at least 9 candidates in U in the final election, contradicting that p is a winner in the final election. □

Due to the above lemma, to make p a winner, p has to beat every candidate in U in the final election. Then, we can use the argument for C-KT(3)-Copeland$^\alpha$-UNI to check the correctness of the reduction.

The $\mathcal{NP}$-hardness of C-KT(4)-Copeland$^\alpha$-UNI, C-KT(4)-Copeland$^\alpha$-NON and D-KT(4)-Copeland$^\alpha$-NON can be proved via reductions from the X4C problem which is $\mathcal{NP}$-hard [20, 21]. The reductions are analogous to C-KT(3)-Copeland$^\alpha$-UNI, C-KT(3)-Copeland$^\alpha$-NON, and D-KT(3)-Copeland$^\alpha$-NON, respectively.

The $\mathcal{NP}$-hardness of C-KT(d)-Copeland$^\alpha$-UNI, C-KT(d)-Copeland$^\alpha$-NON, D-KT(d)-Copeland$^\alpha$-NON and D-KT(d)-Copeland$^\alpha$-UNI for every $d \geq 5$ is implied by the $\mathcal{NP}$-hardness reduction in Theorem 3.2 in [112]. □

We have just examined the $\mathcal{NP}$-hardness of C-KT(d)-Copeland$^\alpha$-UNI, C-KT(d)-Copeland$^\alpha$-NON and D-KT(d)-Copeland$^\alpha$-NON for every $d \geq 3$, and D-KT(d)-Copeland$^\alpha$-UNI for every $d \geq 5$, but left the complexity of D-KT(d)-Copeland$^\alpha$-UNI for $d = 3, 4$ unexamined. We cannot simply adopt the reductions for C-KT(3)-Copeland$^\alpha$-NON and C-KT(4)-Copeland$^\alpha$-NON to prove the $\mathcal{NP}$-hardness of D-KT(4)-Copeland$^\alpha$-UNI and D-KT(4)-Copeland$^\alpha$-UNI, since both candidates y and y' win the election, and thus, no candidate is valid to be the distinguished candidate.

Now we investigate the complexity of the distance restricted bribery problem for Maximin. Before proceeding further, we examine a special kind of Maximin voting profiles. These profiles will be useful in many of the $\mathcal{NP}$-hardness reductions concerning Maximin voting. In particular, these voting profiles contain votes which cyclicly rank the candidates so that the Maximin score of every candidate is at most $\lceil n/m \rceil$, where n is the number of votes and m is the number of candidates. See the example below.

Example: Three cyclic voting profiles with four candidates a, b, c, d.

$1 : a \succ b \succ c \succ d$	$2 : a \succ b \succ c \succ d$	$k \quad : a \succ b \succ c \succ d$
$1 : b \succ c \succ d \succ a$	$2 : b \succ c \succ d \succ a$	$k \quad : b \succ c \succ d \succ a$
$1 : c \succ d \succ a \succ b$	$2 : c \succ d \succ a \succ b$	$k-1 : c \succ d \succ a \succ b$
$1 : d \succ a \succ b \succ c$	$2 : d \succ a \succ b \succ c$	$k-1 : d \succ a \succ b \succ c$
Profile 1	**Profile 2**	**Profile 3**

The voting profile 1 contains four votes which cyclicly rank the candidates with each candidate being ranked in the first position once. It is easy to see that every candidate has Maximin score one in the voting profile 1. In particular, $Min(a) = \{d\}, Min(b) = \{a\}, Min(c) = \{b\}$ and $Min(d) = \{c\}$, where for a candidate q

$$Min(q) = \{q' \in \mathcal{C} \mid \forall (q'' \in \mathcal{C})[N(q, q') \leq N(q, q'')]\}$$

is the set of candidates that achieves the minimum value of $N(q, \cdot)$. That is, for every candidate q, $Min(q)$ is exactly the candidate which is ranked in the last position in the vote where q is ranked in the first position. This rule applies to the voting profile 2 and the voting profile 3. The voting profile 2 is obtained from the voting profile 1 by adding another copy of voting profile 1; thus every candidate is ranked in the first position twice. It is easy to see that every candidate has Maximin score two in the voting profile 2. Voting profile 3 contains $n = 4k - 2$ votes that cyclicly rank the candidate so that every candidate is either ranked in the first position k times or $k - 1$ times. It is easy to check that every candidate has Maximin score $k = \lceil \frac{n}{4} \rceil = \lceil \frac{4k-2}{4} \rceil$.

Every cyclic voting profile, as the ones discussed above, with m candidates has at most m different votes, corresponding to m linear orders over the candidates. We call these m linear orders the *rotate orders* of the cyclic voting profile containing m candidates.

Theorem 4.5. *D-KT(d)-Maximin-UNI, D-KT(d)-Maximin-NON, C-KT(d)-Maximin-UNI and C-KT(d)-Maximin-NON are $\mathcal{NP}$-hard for every $d \geq 4$.*

Proof. We first examine D-KT(4)-Maximin-NON. We prove its $\mathcal{NP}$-hardness by a reduction from the X3C problem. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{x_1, s_2, ..., s_n\})$ of X3C, we create an instance $\mathcal{E}$ for D-KT(4)-Maximin-NON as follows. We assume that every c_i occurs in exactly three subsets in S, and thus, $n = 3\kappa$.

Candidates: The candidate set is $U \cup \{p, q\}$, where q is the distinguished candidate.

Votes: We create $2n-5$ votes in total. We first create n votes corresponding to S. In particular, for each $s_i \in S$, we create one vote π_{s_i} defined as $q \succ \widetilde{s_i} \succ p \succ \widetilde{U \setminus s_i}$, where $\widetilde{s_i}$ and $\widetilde{U \setminus s_i}$ are specific orders over s_i and $U \setminus s_i$, respectively, that are defined as follows. Let Π_{cyclic} be a list of 3κ linear orders over U that rank the candidates in U cyclicly with respect to the rotate order $(c_1, c_2, ..., c_{3\kappa})$, as discussed above. Let $\succ_i$ be the i-th linear order in Π_{cyclic}. Then, $\succ_i$ is defined as

$$c_i \succ_i c_{i+1} \succ_i, ..., \succ_i c_{3\kappa} \succ_i c_1 \succ_i c_2 \succ_i, ..., \succ_i c_{i-1}.$$

Then, $\widetilde{s_i}$ and $\widetilde{U \setminus s_i}$ are defined as the linear order $\succ_i$ restricted to s_i and $U \setminus s_i$, respectively. That is, for every two candidates $a, b \in s_i$ (resp. $a, b \in U \setminus s_i$), the vote π_{s_i} has preference $a \succ b$ if and only if $a \succ_i b$.

In addition to the above votes corresponding to S, we create $n - 2\kappa + 1$ votes defined as $U \succ p \succ q$. Finally, we have $\kappa - 3$ votes defined as $q \succ U \succ p$, and $\kappa - 3$ votes defined as $U \succ p \succ q$. Moreover, the election restricted to U of the last $n-5$ votes rank the candidates in U cyclicly with respect to the rotate order $(c_1, c_2, ..., c_{3\kappa})$. The comparisons between every two candidates are shown in Table 4.5. An example of the construction is shown below.

Example. Let $U = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$ and $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\}$, where $s_1 = \{c_1, c_2, c_8\}, s_2 = \{c_1, c_3, c_5\}, s_3 = \{c_1, c_2, c_9\}, s_4 = \{c_2, c_5, c_7\}, s_5 = \{c_3, c_4, c_9\}, s_6 = \{c_4, c_5, c_6\}, s_7 = \{c_3, c_4, c_7\}, s_8 = \{c_6, c_7, c_8\}, s_9 = \{c_6, c_8, c_9\}$. We have that $\kappa = 3$ and $n = 9$.

The votes corresponding to S are defined as follows.

$\pi_{s_1} : q \succ c_1 \succ c_2 \succ c_8 \succ p \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_9$

$\pi_{s_2} : q \succ c_3 \succ c_5 \succ c_1 \succ p \succ c_2 \succ c_4 \succ c_6 \succ c_7 \succ c_8 \succ c_9$

$\pi_{s_3} : q \succ c_9 \succ c_1 \succ c_2 \succ p \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8$

$\pi_{s_4} : q \succ c_5 \succ c_7 \succ c_2 \succ p \succ c_4 \succ c_6 \succ c_8 \succ c_9 \succ c_1 \succ c_3$

$\pi_{s_5} : q \succ c_9 \succ c_3 \succ c_4 \succ p \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_1 \succ c_2$

$\pi_{s_6} : q \succ c_6 \succ c_4 \succ c_5 \succ p \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ c_3$

$\pi_{s_7} : q \succ c_7 \succ c_3 \succ c_4 \succ p \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ c_5 \succ c_6$

$\pi_{s_8} : q \succ c_8 \succ c_6 \succ c_7 \succ p \succ c_9 \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5$

$\pi_{s_9} : q \succ c_9 \succ c_6 \succ c_8 \succ p \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_7$

The remaining $n - 5 = 4$ votes are defined as follows.

$c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ p \succ q$

$c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ p \succ q$

$c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ p \succ q$

$c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ c_3 \succ p \succ q$

In the above election, each candidate c_i has Maximin score at most 5. In fact, this holds for every candidate in U in the constructed $\mathcal{E}$.

Lemma 4.5. *Every candidate in U has Maximin score at most 5 in $\mathcal{E}$.*

Proof. Due to the definition of Maximin, the Maximin score of a candidate in an election is no greater than the Maximin score of the candidate in a restricted election to a subset of candidates including the candidate. Therefore, to prove the lemma, it is sufficient to consider the election restricted to U. Let c_i be any arbitrary candidate in U. We consider the comparison between c_i and c_{i-1} for every $i = 1, 2, ..., 3\kappa$ (we assume that $c_0 = c_{3\kappa}$). It is easy to see that in the last $n - 5 = 3\kappa - 5$ votes, there is at most one vote that ranks c_i above c_{i-1}. Now we consider the votes corresponding to S. Let Π_{cyclic} be the list of the 3κ linear orders over U as discussed above. In Π_{cyclic}, there is only one vote which ranks c_i above c_{i-1}. Let s_x, s_y, s_z be the 3 subsets in S that contain c_i. Due to the construction of the votes, in the three votes corresponding to $\{s_x, s_y, s_z\}$, the candidate c_i is ranked in the top three positions. This gives three additional votes that potentially rank c_i above c_{i-1}. Therefore, there are at most 4 votes which rank c_i above c_{i-1} in the $n = 3\kappa$ votes corresponding to S. Putting all together, we know that there are at most 5 votes in total that rank c_i above c_{i-1}, implying that the Maximin score of c_i is at most 5. Since c_i is arbitrarily chosen, the lemma follows. □

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we prove the correctness of the reduction.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S'. Consider the final election obtained from $\mathcal{E}$ by replacing the votes in $\Pi_{S'}$. In particular, each vote π_s is replaced with a vote obtained from π_s by promoting p to the first position, that is, the vote defined as $p \succ q \succ s \succ U \setminus s$. It is easy to verify that in the final election p has Maximin

	q	p	c_j
q	-	$n+\kappa-3$	$n+\kappa-3$
p	$n-\kappa-2$	-	$n-3$
c_i	$n-\kappa-2$	$n-2$	≤ 5

Table 4.5: Comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for D-KT(4)-Maximin-NON in Theorem 4.5. The comparisons between c_i and c_j with $i \neq j$ should be read as follows: "for every candidate $c_i \in U$, there exists one candidate $c_j \in U$ such that $N_{\mathcal{E}}(c_i, c_j) \leq 5$".

score $n-2$ ($Min(p) = U \cup \{q\}$) and q has Maximin score $n-3$ ($Min(q) = \{p\}$), implying that q is no longer a winner.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. As shown in Lemma 4.5, every candidate in U has Maximin score at most 5. Given that κ is not a constant and $n = 3\kappa$, we know that no candidate in U can have a higher score than that of q by replacing at most κ votes. Therefore, p is the only candidate which can prevent q from being a winner. This means that in the final election, p has a higher score than that of q. Since $N_{\mathcal{E}}(q, c) = n+\kappa-3$ for every candidate $c \in U \cup \{p\}$, the Maximin score of q in the final election must be at least $n-3$. Hence, the Maximin score of p in the final election must be at least $n-2$. Since $N_{\mathcal{E}}(p, q) = n-\kappa-2$, there must be κ votes ranking q above p that are replaced with κ new votes ranking p above q. Moreover, these new votes should be KT(4)-close to their corresponding original votes. Due to these, we know that the replaced votes must be from the votes corresponding to S. Let $\Pi_{S'}$ be the replaced votes and $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of 3-subsets corresponding to $\Pi_{S'}$. Since for every candidate $c \in U$ we have that $N_{\mathcal{E}}(p, c) = n-3$, there must be at least one vote ranking p above c that is replaced with a new vote ranking c above p. Moreover, the new vote should be KT(4)-close to the original vote. This can only be achieved by replacing a vote π_s with $c \in s$ with a vote defined as $p \succ q \succ s \succ U \setminus s$. Hence, for every $c \in U$ there is at least one vote $\pi_s \in \Pi_{S'}$ with $c \in s$. Since $\Pi_{S'}$ has κ votes and $|U| = 3\kappa$, S' must be an exact 3-set cover.

Now we consider D-KT(4)-Maximin-UNI. The reduction for D-KT(4)-Maximin-UNI is similar to the above reduction with the difference that we create one more vote defined as $q \succ U \succ p$, and create one more vote defined as $U \succ p \succ q$. The comparisons between every two candidates are shown in Table 4.6.

The reductions for C-KT(4)-Maximin-UNI and C-KT(4)-Maximin-NON are similar to that for D-KT(4)-Maximin-NON and D-KT(4)-Maximin-UNI, respectively, with only the difference that in both cases we set p as the distinguished candidate.

The $\mathcal{NP}$-hardness of D-KT(d)-Maximin-UNI, D-KT(d)-Maximin-NON, C-KT(d)-Maximin-UNI and C-KT(d)-Maximin-NON for every $d \geq 5$ is implied by the $\mathcal{NP}$-

	q	p	c_j
q	-	$n+\kappa-2$	$n+\kappa-2$
p	$n-\kappa+1$	-	$n-3$
c_i	$n-\kappa+3$	n	≤ 5

Table 4.6: Comparison between every two candidates in the $\mathcal{NP}$-hardness reduction for D-KT(4)-Maximin-UNI in Theorem 4.5. The comparison between c_i and c_j with $i \neq j$ should be read as follows: "for every candidate $c_i \in U$, there exists one candidate $c_j \in U$ such that $N_{\mathcal{E}}(c_i, c_j) \leq 5$".

hardness reduction in Theorem 4.6 in [110]. □

4.3 Hamming Distance Restricted Bribery

In this section, we study bribery problems with Hamming distance restrictions. We begin with several polynomial-time solvability results.

Theorem 4.6. *D-HAM(d)-Condorcet-UNI, D-HAM(d)-Condorcet-NON, D-HAM(d)-Borda-UNI, and D-HAM(d)-Borda-NON are all polynomial-time solvable, for every possible integer d.*

Proof. We prove the theorem by deriving polynomial-time algorithms for the problems stated in the above theorem. We only describe the algorithms for the unique-winner model in detail. The algorithms for the nonunique-winner model are similar. In the following, let R be the number of votes that can be replaced, and let p be the distinguished candidate.

D-HAM(d)-Condorcet. We first consider D-HAM(2)-Condorcet. The algorithm first guesses a candidate p' which is not beaten by p in the final election. This leads to at most m subinstances, each asking whether we can make p' not be beaten by p by replacing at most R votes with R many HAM(2)-close votes. To solve each subinstance, we need only to arbitrarily choose up to R votes which rank p above p', and replace each of them with a new vote obtained from the original vote by swapping p and p'. After this, if p' is not beaten by p, the subinstance is a yes-subinstance, and thus, we return "Yes"; otherwise, the subinstance is a no-subinstance, and thus, we discard the subinstance and proceed to the next one. If there is no subinstance that leads to a "Yes" answer, we return "No". The above algorithm directly applies to D-HAM(d)-Condorcet for every possible $d \geq 2$.

D-HAM(2)-Borda. The algorithm first guesses a candidate p' which prevents p from being the unique-winner in the final election. This leads to at most m subinstances, each asking whether we can make p' have an equal or greater Borda score than that of p by replacing at most $\mathcal{R}$ votes with $\mathcal{R}$ many HAM(2)-close votes. To solve each subinstance, we order the votes π_v according to the nonincreasing order of

$$\max\{pos_v(p') - 1, m - pos_v(p), 2 \cdot (pos_v(p') - pos_v(p))\}.$$

Here, $pos_v(c)$ is the position of the candidate c in the vote π_v and m is the number of candidates. Let Π be the multiset of the first $\mathcal{R}$ votes according to this order. Then, we replace every vote in Π in the following way. For each $\pi_v \in \Pi$, if $pos_v(p') - 1 \geq m - pos_v(p)$ and $pos_v(p') - 1 \geq 2 \cdot (pos_v(p') - pos_v(p))$, then replace π_v with a new vote obtained from π_v by swapping p' and the first ranked candidate in π_v; otherwise, if $m - pos_v(p) \geq pos_v(p') - 1$ and $m - pos_v(p) \geq 2 \cdot (pos_v(p') - pos_v(p))$, replace π_v with a vote obtained from π_v by swapping p and the last ranked candidate in π_v; finally, if $2 \cdot (pos_v(p') - pos_v(p)) \geq pos_v(p') - 1$ and $2 \cdot (pos_v(p') - pos_v(p)) \geq m - pos_v(p)$, replace π_v with a vote obtained from the original vote by swapping p and p'. After doing this for every vote in Π, if p' has an equal or greater Borda score than that of p, the subinstance is a yes-subinstance, and thus, we return “Yes”; otherwise, the subinstance is a no-subinstance, and thus, we discard this subinstance and proceed to the next one. If there is no subinstance that leads to a “Yes” answer, we return “No”.

D-HAM(3)-Borda. The algorithm carries out at most m guesses as in the above algorithm for D-HAM(2)-Borda. Now we restrict our attention to the subinstances. We divide the votes into two multisubsets Π_1 and Π_2, where Π_1 includes all votes that rank p above p', and Π_2 includes all votes that rank p' above p. Then, we maintain two Stacks S_1 and S_2 to store the votes in Π_1 and Π_2, respectively[i]. Precisely, for each $i = 1, 2$, the votes π_v in Π_i are inserted into S_i one by one, according to the nondecreasing order of $\max\{pos(p') - 1, m - pos(p)\}$ (therefore, a vote π_u with minimum $\max\{pos_u(p') - 1, m - pos_u(p)\}$ is at the button and a vote π_u with maximin $\max\{pos_u(p') - 1, m - pos_u(p)\}$ is at the top of the stack). Then, we call Algorithm 4.1 to deal with the subinstances. Clearly, the given instance of D-HAM(3)-Borda is a yes-instance if and only if at least one of the subinstance is a yes-subinstance.

D-HAM(4)-Borda. Similar to the above algorithms for Borda, the algorithm for D-HAM(4)-Borda first carries out at most m guesses, leading to at most m subinstances. Now we restrict our attention to these subinstances. For each subinstance, we divide the votes into two multisubsets Π_1 and Π_2, where Π_1 includes all the votes that rank p above p' and Π_2 includes all the votes that rank p' above p. Then, we order the votes in Π_1 according to the nonincreasing order of $pos(p') - pos(p)$. Then, we choose the first

[i] A *stack* is a collection of objects that are inserted and removed according to the last-in, first-out (LIFO) principle [139]. The function *pop*() removes and returns the top element from the stack, and the function *top*() returns the top element of the stack, without removing it. We refer to Chapter 6 of [139] or Chapter 3 of [192] for further discussions on stacks.

Algorithm 4.1: A procedure to deal with subinstances in the algorithm for D-HAM(3)-Borda.

while $\mathcal{R} > 0$ **do**
    Let $\pi_v = S_1.top()$;
    Let $\pi_u = S_2.top()$;
    Let $a = \max\{2 \cdot pos_v(p') - pos_v(p) - 1, m + pos_v(p') - 2pos_v(p)\}$;
    Let $b = \max\{pos_u(p') - 1, m - pos_u(p)\}$;
    if $a \geq b$ **then**
        $S_1.pop()$;
        if $2 \cdot pos_v(p') - pos_v(p) - 1 \geq m + pos_v(p') - 2pos_v(p)$ **then**
            replace π_v with a vote obtained from π_v by first swapping p and p', and then swapping p' and the first ranked candidate;
        else
            replace π_v with a vote obtained from π_v by first swapping p and p', and then swapping p and the last ranked candidate;
        end
    else
        $S_2.pop()$;
        if $pos_u(p') - 1 \geq m - pos_u(p)$ **then**
            replace π_u with a vote obtained from π_u by swapping p' and the first ranked candidate;
        else
            replace π_u with a vote obtained from π_u by swapping p and the last ranked candidate;
        end
    end
    $\mathcal{R} := \mathcal{R} - 1$;
end
if *p' has a no less Borda score than that of p* **then**
    Return "Yes";
else
    Return "No";
end

up to $\mathcal{R}$ votes, and replace each of them with a vote obtained from the original vote by swapping p' and the first ranked candidate, and swapping p and the last ranked candidate. After doing so, if p' has a no less score than that of p, we return "Yes". Otherwise, if p' has a less score than that of p, we distinguish between two cases. If $|\Pi_1| \geq \mathcal{R}$, we return "No" immediately. In the case that $|\Pi_1| < \mathcal{R}$, we order the votes in Π_2 according to the nondecreasing order of $pos(p) - pos(p')$. Then, we choose the first $\mathcal{R} - |\Pi_1|$ votes, and replace each of them with a vote obtained from the original vote by swapping p' and the first ranked candidate, and swapping p with the last ranked candidate. After doing this, if p' has a no less score than that of p, we return "Yes"; otherwise, return we discard the current considered subinstance and proceed to the next one. If no subinstance leads to a “Yes” answer, we return “No”.

D-HAM(d)-Borda. The algorithm for D-HAM(d)-Borda with $d > 4$ is exactly the same as for D-HAM(4)-Borda. □

Now we show our hardness results. We begin with the distance restricted bribery problem for Copeland$^\alpha$. All $\mathcal{NP}$-hardness in this section is reduced from the X3C problem. In the following, we assume that in the X3C problem, each element c_i in the universal set occurs in exactly three subsets of the collection $\mathcal{S}$. This assumption does not change the $\mathcal{NP}$-hardness of the problem [138]. Note that under this assumption, we have that the size of the collection is $|\mathcal{S}| = 3\kappa$.

Theorem 4.7. *C-HAM(2)-Copeland$^\alpha$-UNI, C-HAM(2)-Copeland$^\alpha$-NON, D-HAM(2)-Copeland$^\alpha$-UNI and D-HAM(2)-Copeland$^\alpha$-NON are $\mathcal{NP}$-hard for every $0 \leq \alpha \leq 1$.*

Proof. We first consider C-HAM(2)-Copeland$^\alpha$-UNI. Given an instance $\mathcal{F} = (U = \{c_1, c_2, ..., c_{3\kappa}\}, S = \{s_1, s_2, ..., s_{3\kappa}\})$ of X3C, we create an instance $\mathcal{E}$ for C-HAM(2)-Copeland$^\alpha$-UNI as follows.

Candidates: We create $3\kappa+2$ candidates in total. In particular, for each element $c_i \in U$, we create one candidate. For convenience, we still use c_i to denote the candidate corresponding to c_i. In addition, we have two candidates p and q with p being the distinguished candidate.

Votes: For each $s \in S$, we create a vote π_s defined as $q \succ U \setminus s \succ p \succ s$. In addition, we create $\kappa - 1$ votes defined as $p \succ q \succ U$, and two votes defined as $U \succ p \succ q$. In total, we have $4\kappa + 1$ votes. The comparisons between every two candidates are summarized in Table 4.7. It is easy to verify the candidate q beats every other candidate; and thus, q is the current unique winner.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

	q	p	c_j
q	-	3κ	$4\kappa - 1$
p	$\kappa + 1$	-	$\kappa + 2$
c_i	2	$3\kappa - 1$	$\cdots$

Table 4.7: Comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for C-HAM(2)-Copeland$^\alpha$-UNI in Theorem 4.7. The comparisons between c_i and c_j for $i \neq j$ do not play any role in the correctness argument.

Now we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{E}$ is a yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and S' is an exact 3-set cover. Let $\Pi_{S'} = \{\pi_s \mid s \in S'\}$ be the set of votes corresponding to S'. Consider the final election $\mathcal{E}'$ obtained from $\mathcal{E}$ by replacing every vote π_s with a vote obtained from π_s by swapping p and q. More precisely, each $\pi_s \in \Pi_{S'}$ defined as $q \succ U \setminus s \succ p \succ s$ is replaced with the vote defined as $p \succ U \setminus s \succ q \succ s$. Clearly, the Hamming distance between these two votes is two. Moreover, we have that $N_{\mathcal{E}'}(p,q) = 2\kappa + 1$. Now we consider the comparison between p and every $c_i \in U$. Since S' is an exact 3-set cover, for every c_i there are exactly $\kappa - 1$ votes $\pi_s \in \Pi_{S'}$ with $c_i \notin s$. All these votes rank c_i above p in $\mathcal{E}$. However, these votes are replaced with $\kappa - 1$ votes which rank p above c_i as discussed above, in the final election $\mathcal{E}'$. Therefore, for every $c_i \in U$, there are $(\kappa+2)+(\kappa-1) = 2\kappa+1$ votes which rank p above c_i, implying that p beats every $c_i \in U$ in $\mathcal{E}'$. Summary all above, p becomes the unique winner in $\mathcal{E}'$.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by replacing at most κ votes. Since $N_{\mathcal{E}}(q, c_i) = 4\kappa - 1$, we know that q beats every candidate $c_i \in U$ in the final election $\mathcal{E}'$. Due to this, we know that q is beaten by p in $\mathcal{E}'$, since otherwise, q would beat every other candidate in the final election $\mathcal{E}'$, and thus, remains the unique winner. Moreover, since p is the unique winner in $\mathcal{E}'$, p must beat every other candidate in the final election $\mathcal{E}'$. Since $N_{\mathcal{E}}(p,q) = \kappa + 1$, in order to make p beat q, there has to be κ votes ranking q above p that are replaced by κ new votes ranking p above q. Due to this, we know that the replaced votes are from the votes corresponding to S, since any other vote has already ranked p above q. Let $\Pi_{S'}$ be the replaced votes, and Let $S' = \{s \mid \pi_s \in \Pi_{S'}\}$ be the subcollection of 3-subsets corresponding to $\Pi_{S'}$. As discussed above, p beats every candidate $c_i \in U$ in $\mathcal{E}'$. Since $N_{\mathcal{E}}(p, c_i) = \kappa + 2$, for every $c_i \in U$, there must be at least $\kappa - 1$ votes in $\Pi_{S'}$ ranking c_i above p that are replaced by $\kappa - 1$ votes ranking p above c_i. This happens only if S' is an exact 3-set cover.

Now we consider C-HAM(2)-Copeland$^\alpha$-NON. The reduction is adapted from the above one for C-HAM(2)-Copeland$^\alpha$-UNI by introducing another dummy candidate y which beats p but is beaten by q in both the original election and the final election. This ensures that p can become a winner only if p beats every candidate in U. Precisely, we create the following votes. For each $s \in S$, we create a vote π_s defined as $q \succ U \setminus s \succ y \succ p \succ s$. In addition, we create $\kappa - 1$ votes defined as $p \succ q \succ U \succ y$, and two votes defined as $U \succ y \succ p \succ q$. The comparisons between every two candidates are summarized in Table 4.8.

Now we consider D-HAM(2)-Copeland$^\alpha$-UNI and D-HAM(2)-Copeland$^\alpha$-NON. The reduction for D-HAM(2)-Copeland$^\alpha$-NON is the same as for C-HAM(2)-Copeland$^\alpha$-UNI with only the difference that we set q as the distinguished candidate. The correctness argument relies on the fact that no candidate in U can have a higher score than that of q by replacing at most κ votes (since $N_{\mathcal{E}}(c_i, q) = 2$). The reduction for D-HAM(2)-Copeland$^\alpha$-UNI is similar to the one for C-HAM(2)-Copeland$^\alpha$-NON. The differences are as follows. First, we set q as the distinguished candidate in the reduction for D-HAM(2)-Copeland$^\alpha$-UNI. Second, we rank the candidates in U in a cyclic way (as in the proof for D-KT(4)-Maximin-NON in Theorem 4.5) so that for

	q	p	c_j	y
q	-	3κ	$4\kappa - 1$	$4\kappa - 1$
p	$\kappa + 1$	-	$\kappa + 2$	$\kappa - 1$
c_i	2	$3\kappa - 1$	$\cdots$	$4\kappa + 1$
y	2	$3\kappa + 2$	0	-

Table 4.8: Comparison between every two candidates in the $\mathcal{NP}$-hardness reduction for C-HAM(2)-Copeland$^\alpha$-NON in Theorem 4.7. The comparisons between c_i and c_j for $i \neq j$ do not play any role in the correctness argument.

every candidate $c_i \in U$ there exists another candidate $c_j \in U$ with $N_\mathcal{E}(c_i, c_j) \leq \Omega$, where Ω is a small constant (in fact, $\Omega \leq 5$). See below for an example. By doing so, every candidate $c_i \in U$ is beaten by at least two candidates (q and some $c_j \in U$ with $N_\mathcal{E}(c_i, c_j) \leq \Omega$) in the final election $\mathcal{E}'$. This ensures that the only candidate which can prevent q from being the unique winner is p. Then, the correctness follows from the argument for C-HAM(2)-Copeland$^\alpha$-NON.

Example. Let $U = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, c_9\}$ and $S = \{s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9\}$, where $s_1 = \{c_1, c_2, c_8\}, s_2 = \{c_1, c_3, c_5\}, s_3 = \{c_1, c_2, c_9\}, s_4 = \{c_2, c_5, c_7\}, s_5 = \{c_3, c_4, c_9\}, s_6 = \{c_4, c_5, c_6\}, s_7 = \{c_3, c_4, c_7\}, s_8 = \{c_6, c_7, c_8\}, s_9 = \{c_6, c_8, c_9\}$.

The votes corresponding to S are as follows.

$\pi_{s_1} : q \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_9 \succ y \succ p \succ c_1 \succ c_2 \succ c_8$

$\pi_{s_2} : q \succ c_2 \succ c_4 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ y \succ p \succ c_3 \succ c_5 \succ c_1$

$\pi_{s_3} : q \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ y \succ p \succ c_9 \succ c_1 \succ c_2$

$\pi_{s_4} : q \succ c_4 \succ c_6 \succ c_8 \succ c_9 \succ c_1 \succ c_3 \succ y \succ p \succ c_5 \succ c_7 \succ c_2$

$\pi_{s_5} : q \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_1 \succ c_2 \succ y \succ p \succ c_9 \succ c_3 \succ c_4$

$\pi_{s_6} : q \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ c_3 \succ y \succ p \succ c_6 \succ c_4 \succ c_5$

$\pi_{s_7} : q \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ c_5 \succ c_6 \succ y \succ p \succ c_7 \succ c_3 \succ c_4$

$\pi_{s_8} : q \succ c_9 \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ y \succ p \succ c_8 \succ c_6 \succ c_7$

$\pi_{s_9} : q \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_7 \succ y \succ p \succ c_9 \succ c_6 \succ c_8$

The following $\kappa - 1 = 2$ votes are as follows.

$p \succ q \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ y$

$p \succ q \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ y$

The final two votes are as follows.

$c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ y \succ p \succ q$

$c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ c_9 \succ c_1 \succ c_2 \succ c_3 \succ y \succ p \succ q$ □

Now we study Condorcet.

Theorem 4.8. *C-HAM(2)-Condorcet-UNI and C-HAM(2)-Condorcet-NON are $\mathcal{NP}$-hard.*

Proof. The reductions for both problems are exactly the same as for C-HAM(2)-Copeland$^\alpha$-UNI in Theorem 4.7. □

Finally, we consider Maximin. The following theorem summarizes our results.

Theorem 4.9. *D-HAM(2)-Maximin-UNI, D-HAM(2)-Maximin-NON, C-HAM(2)-Maximin-UNI and C-HAM(2)-Maximin-NON are $\mathcal{NP}$-hard.*

Proof. We first consider D-HAM(2)-Maximin-NON. We show the $\mathcal{NP}$-hardness via a reduction from a variant of Hitting Set problem which is also $\mathcal{NP}$-hard.

> Square Hitting Set
>
> Input: A universal set $X = \{x_1, x_2, ..., x_n\}$ and a collection $\mathcal{S} = \{s_1, s_2, ..., s_m\}$ of κ-subsets of X.
>
> Question: Is there a subset $X' \subseteq X$ of size κ which hits each κ-subset in $\mathcal{S}$, that is $X' \cap s_i \neq \emptyset$ for every $s_i \in \mathcal{S}$?

The $\mathcal{NP}$-hardness of Square Hitting Set can be reduced from the $\mathcal{NP}$-hard problem 3-Hitting Set [131]: for each 3-subset introduce $\kappa - 3$ new elements to this 3-subset. Each of these new elements occurs in exactly one 3-subset (of original instance). We assume that each element $x \in X$ occurs in at most 3 subsets in $\mathcal{S}$. This does not change the $\mathcal{NP}$-hardness of the problem, since the 3-Hitting Set problem remains $\mathcal{NP}$-hard when each element of the universal set occurs in at most 3 subsets of the collection.

Given an instance $\mathcal{F} = (X, \mathcal{S}, \kappa)$ of Square Hitting Set, we construct an instance $\mathcal{E}$ for D-HAM(2)-Maximin-NON as follows.

Candidates: We have $m + 2$ candidates $\mathcal{S} \cup \{p, q\}$, where p is the distinguished candidate.

Votes: For an $x \in X$, let $A(x)$ be the set of κ-subsets in $\mathcal{S}$ which contain x, that is, $A(x) = \{s \in \mathcal{S} \mid x \in s\}$. We create the votes as follows. For each $x \in X$, we create a vote π_x defined as $p \succ A(x) \succ q \succ \mathcal{S} \setminus A(x)$. In addition, we have $n - 2\kappa + 1$

votes each defined as $\mathcal{S} \succ q \succ p$. Therefore, we have $2n - 2\kappa + 1$ votes in total. We remark that among all these votes, we cyclicly changing the order of candidates in $\mathcal{S}$, as discussed in the proof for D-KT(4)-Maximin-NON in Theorem 4.5, so that each candidate in $\mathcal{S}$ has Maximin score at most $\lceil \frac{2n-2\kappa+1}{n+2} \rceil + 3 \leq 5$, which is extremely smaller than the Maximin score of p and q; and thus, none of the candidates in $\mathcal{S}$ can be a winner. The comparisons between every two candidates are summarized in Table 4.9. Clearly, p is the current winner.

	q	p	s_j
q	-	$n - 2\kappa + 1$	$n - \kappa$
p	n	-	n
s_i	$n - \kappa + 1$	$n - 2\kappa + 1$	5

Table 4.9: Comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for D-HAM(2)-Maximin-NON in Theorem 4.9.

Number of Replaced Votes: $\mathcal{R} = \kappa$.

Now we prove the correctness.

($\Rightarrow$:) Suppose that $\mathcal{F}$ is a yes-instance and X' is a hitting set of size κ. Let $\Pi_{X'} = \{\pi_x \mid x \in X'\}$ be the votes corresponding to X'. Let $\mathcal{E}'$ be the final election obtained from $\mathcal{E}$ by replacing every vote $\pi_x \in \Pi_{X'}$ with a new vote defined as $q \succ A(x) \succ p \succ \mathcal{S} \setminus A(x)$. It is easy to verify that $N_{\mathcal{E}'}(p, q) = n - \kappa$; thus, the Maximin score of p is at most $n - \kappa$. Now let's calculate the Maximin score of q in the final election $\mathcal{E}'$. Clearly, $N_{\mathcal{E}'}(q, p) = n - \kappa + 1$. It remains to examine $N_{\mathcal{E}'}(q, s_i)$ for every $s_i \in \mathcal{S}$. Since X' is a hitting set, for every s_i there must be at least one $x \in X'$ with $s_i \in A(x)$. Therefore, for every $s_i \in \mathcal{S}$, there is at least one vote in $\Pi_{X'}$ ranking s_i above q that is replaced with a vote ranking q above s_i, in the final election $\mathcal{E}'$. It follows that $N_{\mathcal{E}'}(q, s_i) \geq n - \kappa + 1$ for all $s_i \in \mathcal{S}$. Clearly, p is no longer a winner in the final election $\mathcal{E}'$.

($\Leftarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Let $\mathcal{E}'$ be the final election wherein p is no longer a winner. Observe that no candidate $s_i \in \mathcal{S}$ can have a higher score than that of p in $\mathcal{E}'$, since the candidates in $\mathcal{S}$ are ranked cyclicly so that each of them has only a constant Maximin score. This leaves only the possibility that q has a higher score than that of p in $\mathcal{E}'$. Since p has Maximin score n in $\mathcal{E}$, we know that the Maximin score of p in $\mathcal{E}'$ is at least $n - \kappa$ (since we can replace at most κ votes); therefore, the Maximin score of q in $\mathcal{E}'$ must be at least $n - \kappa + 1$. Since $N_{\mathcal{E}}(q, p) = n - 2\kappa + 1$, there must be κ votes ranking p above q in $\mathcal{E}$ that are replaced with κ votes ranking q above p in the final election $\mathcal{E}'$. Due to this, we know that all replaced votes are from the votes corresponding to X. Let $\Pi_{X'}$ be the replaced votes and $X' = \{x \in X \mid \pi_x \in \Pi_{X'}\}$ be the subsets corresponding to $\Pi_{X'}$. Since $N_{\mathcal{E}}(q, s_i) = n - \kappa$ for all $s_i \in \mathcal{S}$, we know that for every $s_i \in \mathcal{S}$ there must be at least one vote $\pi_x \in \Pi_{X'}$ with $s_i \in A(x)$ that is replaced with a vote that ranks q above s_i. This happens only if X' is a hitting set.

The reduction for D-HAM(2)-Maximin-UNI is similar to the above reduction for D-HAM(2)-Maximin-NON with the following differences. First, we twist the Square Hitting Set problem a bit. In particular, we require that every $s \in \mathcal{S}$ is a $(\kappa+1)$-subset of X other than a κ-subset as in the Square Hitting Set problem defined above (the question is still to determine whether there is a hitting set of size κ). This does not change the complexity (the hardness can be reduced from the 3-Hitting set problem, with similar method as for Square Hitting Set discussed above). Second, we create one less vote defined as $\mathcal{S} \succ q \succ p$. All the remaining parts remain the same. The comparisons between every two candidates are summarized in Table 4.10.

	q	p	s_j
q	-	$n-2\kappa$	$n-\kappa-1$
p	n	-	n
s_i	$n-\kappa+1$	$n-2\kappa$	≤ 5

Table 4.10: Comparisons between every two candidates in the $\mathcal{NP}$-hardness reduction for D-HAM(2)-Maximin-UNI in Theorem 4.9. The comparisons between c_i and c_j with $i \neq j$ do not play any role in the correctness argument.

Now we consider the constructive case. The reductions for C-HAM(2)-Maximin-UNI is the same as for D-HAM(2)-Maximin-NON with the only difference that we set q as the distinguished candidate. The reduction for C-HAM(2)-Maximin-NON is the same as for D-HAM(2)-Maximin-UNI with the only difference that we set q as the distinguished candidate. □

4.4 Conclusion

We have studied the complexity of distance restricted bribery problem which differs from the traditional bribery problem in the sense that the bribed voters only recast new votes which are “close” to their original votes. In particular, we adopted the Hamming distance and the KT-distance to measure the closeness between two votes (linear orders). Our results are summarized in Table 4.1.

There remain several open problems as shown in Table 4.1. Another possible revenue of research would be to explore these problems from the parameterized complexity viewpoint. Furthermore, exploring the same problems with respect to further distance measurements (see [92, 97, 165] for several distance measurements on linear orders) is also an interesting direction for future research.

5

Possible Winners in Partial Tournaments

Partial tournaments play a significant role in many areas linked to our daily life. For example, in the group stage of the World Cup matches, every pair of the four teams in the same group play against each other. In each match, the winner gets 3 points and the loser gets 0 points. If they tie, both get 1 point. Equivalently, we can say that the winner gets 2 points, the loser -1 point, and both get 0 points if they tie. The two teams with the highest and the second highest scores are qualified to compete in the second stage and the remaining two teams are knocked out. This procedure can be represented by a partial tournament: if team A wins in the compete with team B, introduce an arc from A to B and label the arc with $(2, -1)$, meaning that A gets 2 points and B gets -1 point in the match between A and B. The score of a team is the sum of the first components of the labels to the arcs leaving from the team, plus the sum of the second components of the labels to the arcs arriving at the team.

5.1 Introduction

A tournament can be expressed as a directed graph where between every pair of vertices there is exactly one arc (for readers who are unfamiliar with directed graphs and tournaments, we refer to the textbook by Bang-Jensen and Gutin [15]). Tournaments play a significant role in voting systems due to their nice expression ability in many winner determination problems. For example, tournaments can perfectly illustrate the Condorcet winner determination problem (when the number of voters is odd, or more generally when there is no tie in comparisons between every two candidates): create a vertex for each candidate and add an arc (v, u) from the vertex v to the vertex u if more than half of the voters prefer v to u. Then, the Condorcet winner is the candidate who has an arc to every other candidate. Several other winner determination methods are also based on tournaments, such as Banks, Slater, and Schwartz winners [48, 153]. However, in practical settings, we might not be able to access the full information of an election to build the tournament. For example, the number of candidates is too huge to give a full preference at once, or consider an online voting where in each time only part of the votes is submitted. We refer to [170] for more detailed discussion. In these cases, a partial tournament may be a useful tool, and thus, the problems of deciding which candidates have positive possibility to win the election should be of particular importance (A partial tournament is a tournament with some arcs missing). Partial tournaments also appear in settings where ties occur in pairwise comparisons between candidates. For example, we have an election to select the Condorcet winner. If the number of voters is even, then, it is possible that for two candidates v and u, exactly half of the voters prefer v to u and the others prefer u to v.

Tournament solutions have wide applications in decision-making problems and in social choice area, and have received considerable attention recently [19, 42, 43, 46, 91, 200, 228]. Precisely, a *tournament solution* maps a tournament to a non-empty set of vertices in the tournament. We refer to Chapter 3 of [47] for a survey of tournament solutions. Banks set and Uncovered set are two of the most important tournament solutions which have been extensively studied from the viewpoints of game theory, economics, computational complexity, etc [45, 91, 132, 197]. Banks set is named by its introducer Banks [16]. Given a tournament, a candidate (a vertex in the tournament) v is a Banks winner, if there is a maximal transitive subtournament with v being the 0-indegree vertex. Here, "transitive" means that for every three vertices v, u, w in a tournament D, the existence of arcs (v, u) and (u, w) in D implies that (v, w) is in D. The *Banks set* then contains all Banks winners. Clearly, if the Condorcet winner exists, then the Banks set contains exactly the Condorcet winner. The *Uncovered set* of a tournament is a maximal subset C of candidates such that no candidate outside C dominates a candidate in C. Here, a candidate v *dominates* a candidate u if all out-neighbors of u are also out-neighbors of v. Thus, an *Uncovered set* includes exactly all vertices each of which can reach any other vertex in no more than two steps (a precise definition is in Section 5.1.2). The vertices in an Uncovered set are called *kings* from the viewpoint of graph theory. It is well-known that every tournament contains at least one king [174]. Moreover, if the Condorcet winner exists, then the Uncovered set contains only the Condorcet winner. It is a folklore that Banks set is a subset of Uncovered set [115]. Nevertheless, Uncovered set has some advantages compared with Banks set. For example, determining whether a candidate is a Banks winner is $\mathcal{NP}$-hard [242], while computing the Uncovered set is solvable in polynomial time [153]. Selecting the elements from the Uncovered set as the winners of the given tournament has been independently suggested by Fishburn [124] and Miller [196].

5.1.1 Motivation

In this chapter, we study some parameterized problems related to Uncovered set and Banks set on partial tournaments. We first study the *possible winners of Uncovered set* problem [12]: given a partial tournament and a subset X of vertices, we are seeking for a completion of D such that all vertices in X become kings, or equivalently, all vertices in X are in the Uncovered set. For convenience, in the following we use the terminology "kings" instead of "Uncovered set". We study the problem with the size of X as the parameter. The motivation is based on the observation that in practical settings, one is mostly interested in making few vertices, which correspond to candidates, to become winners. We prove that this problem is in $\mathcal{XP}$; thus, when the size of X is bounded by a constant, it can be solved in polynomial time. In addition, we study two variants of the problem where we are asked to make all vertices of X kings by modifying few number of arcs. We study two kinds of modifications: adding arcs and reversing arcs. In the "adding arcs" case we are allowed to add at most $\mathcal{R}$ arcs to the partial tournament, while in the "reversing arcs" case we are allowed to reverse at most $\mathcal{R}$ arcs in the partial tournament. For both problems, $\mathcal{R}$ is the parameter. These two parameterized variants could illustrate a bribery strategic behavior. For example, consider a politician in a political election who wants to make one of his accomplices win the election. Then, the arc reversal and arc addition problems illustrate the case where the politician has limited money and to bribe voters to change the pairwise compared relationship between every two candidates needs a cost. We prove that, somewhat surprisingly, both variants are $\mathcal{W}[2]$-hard, even when X contains only a single vertex. Furthermore, our $\mathcal{W}[2]$-hardness proof for the "reversing arcs" case applies to the special case where the input is a tournament and X contains only a single vertex. These results imply that the problems of finding the minimum number of arcs which are needed to add (resp. to reverse) to make all vertices of X kings are beyond $\mathcal{XP}$, when consider the size of X as the parameter. Finally, we study a *possible winner problem related to Banks set* on partial tournaments, where we are

Problems	Parameterized Complexity	Evidence
PWU	$\mathcal{XP}$	Thm. 5.1
PWU-ADD	$\mathcal{W}[2]$-h even when $\lvert X\rvert = 1$	Thm. 5.2
PWU-REVERSE	$\mathcal{W}[2]$-h even on tournaments and with $\lvert X\rvert = 1$	Thm. 5.3
TW	$\mathcal{W}[2]$-h	Thm. 5.4
TW-INDEGREE	$\mathcal{W}[1]$-h	Thm. 5.5
TW-OUTDEGREE	$\mathcal{FPT}$ but no polynomial kernel unless $\mathcal{PH} = \sum_{\mathcal{P}}^{3}$	Thm. 5.6

Table 5.1: A summary of our results concerning possible winner(s) problems in partial tournaments. Here, "$\mathcal{W}[2]$-h" stands for "$\mathcal{W}[2]$-hard" and "$\mathcal{W}[1]$-h" stands for "$\mathcal{W}[1]$-hard". Moreover, "Thm. #" means that the corresponding result is from Theorem #. The precise definitions of the problems can be found in Section 5.1.2.

given a partial tournament D and a distinguished vertex p, and asked whether D has a maximal transitive subtournament with p being the 0-indegree vertex. This problem is a natural generalization of Banks winner to partial tournaments. Here we study three parameterizations. The first parameter we study is the size of the subtournament we are looking for. We prove that this parameter leads to a $\mathcal{W}[2]$-hardness result. Then, we study the parameter defined as the number of candidates who defeat p. We show that the problem is $\mathcal{W}[1]$-hard with respect to this parameter. Finally, we consider the Copeland^0 score of p (the number of candidates defeated by p) as the parameter. Different from the previous results, we show that the problem with the Copeland^0 score of p as the parameter is $\mathcal{FPT}$. However, we prove that the problem does not have a polynomial kernel unless the polynomial hierarchy collapses to the third level. Our main results of this chapter are summarized in Table 5.1.

5.1.2 Preliminaries

A *directed graph* D is a pair (V, A) where V is the set of vertices and A is the set of arcs. An arc from a vertex v to a vertex u is denoted by (v, u). We say v is the *tail* of (v, u) and u is the *head* of (v, u). For simplicity, we also use $A(D)$ and $V(D)$ to denote the set of arcs and the set of vertices of D, respectively. For a vertex v, we use $N_D^-(v)$ and $N_D^+(v)$ to denote the set of its *in-neighbors* and the set of its *out-neighbors* in D, respectively, that is, $N_D^-(v) = \{u \mid (u, v) \in A(D)\}$ and $N_D^+(v) = \{u \mid (v, u) \in A(D)\}$. We drop the index D if it is clear from context. The *in-degree* and *out-degree* of v, denoted by $d^-(v)$ and $d^+(v)$, are the sizes of $N^-(v)$ and $N^+(v)$, respectively. Meanwhile, we say that v is a $d^-(v)$-*indegree vertex* or a $d^+(v)$-*outdegree vertex*. The subgraph induced by a subset $S \subseteq V(D)$, denoted by $D[S]$, is $D[S] = (S, \{(u,v) \mid u \in S, v \in S, (u,v) \in A(D)\})$.

A *partial tournament* is a directed graph such that $|\{(v,u),(u,v)\} \cap A(D)| \leq 1$ for all $v, u \in V$ and $(v,v) \notin A(D)$ for all $v \in V$. If there is no arc between two vertices v and u in D, then we call (v,u) and (u,v) *missing arcs*. A *tournament* is a partial tournament without missing arcs. A tournament D is a *completion* of a partial tournament D' if $V(D) = V(D')$ and $A(D') \subseteq A(D)$.

A tournament D is *transitive* if there is an order $(v_1, v_2, ..., v_n)$ of $V(D)$ such that there is no arc (v_j, v_i) with $j > i$ (or, equivalently, for every three vertices v, u, w, $(v,u) \in A(D)$ and $(u,w) \in A(D)$ implies $(v,w) \in A(D)$). Clearly, there is a unique 0-indegree vertex in every transitive tournament. For a partial tournament and a subset $S \subseteq V(D)$, we say $D[S]$ is a *maximal transitive subtournament* of D if $D[S]$ induces a transitive tournament and no other vertices outside S can be added to S to form a bigger induced transitive tournament.

For two vertices v and u, we say v *can reach* u if $(v,u) \in A(D)$ or there is a $w \in V(D) \setminus \{v,u\}$ with $(v,w) \in A(D)$ and $(w,u) \in A(D)$. In the former case we say v *reaches* u *directly*, while in the latter case we say that v *reaches* u *by (or through)* w. A *king* in a directed graph is a vertex which can reach all other vertices. For a subset $X \subseteq V(D)$ and a vertex $v \in V(D)$, v is a *serf with respect to* X if v can be reached by all vertices in $X \setminus \{v\}$.

In the following, when we say "adding an arc", we mean to add an arc between two vertices which have no arc between them in advance. Thus, adding an arc to a partial tournament still results in a partial tournament. *Reversing an arc* $(v,u) \in A(D)$ is the operation that firstly deletes (v,u) from D, and then adds a new arc (u,v) to D. The parameterized problems studied in this chapter are defined as follows.

> Possible Winners of Uncovered Set (Pwu)
>
> *Input:* A partial tournament $D = (V, A)$ and a subset $X \subseteq V$.
>
> *Parameter:* $|X|$.
>
> *Question:* Is there a completion of D such that all vertices in X are kings?

PWU-ADD (resp. PWU-REVERSE)

Input: A partial tournament $D = (V, A)$ and a subset $X \subseteq V$.

Parameter: A positive integer $\mathcal{R}$.

Question: Can we add (resp. reverse) at most $\mathcal{R}$ arcs such that all vertices in X are kings?

Transitive Winner on Partial Tournaments (TW)

Input: A partial tournament $D = (V, A)$ and a vertex $p \in V$.

Parameter: A positive integer $\mathcal{R}$.

Question: Is there a subset $S \subseteq V$ of size $\mathcal{R}$ such that $D[S]$ is a maximal transitive tournament with p being the 0-indegree vertex?

TW-INDEGREE (resp. TW-OUTDEGREE)

Input: A partial tournament $D = (V, A)$ and a vertex $p \in V$.

Parameter: $|N^-(p)|$ (resp. $|N^+(p)|$).

Question: Is there a subset $S \subseteq V$ such that $D[S]$ is a maximal transitive tournament with p being the 0-indegree vertex?

5.1.3 Related Works

Aziz et al. [12] studied possible and necessary winner(s) problems in partial tournaments for diverse tournament solution concepts. They mainly considered three topics: deciding whether a given candidate is a possible (resp. a necessary) winner, and deciding whether a given subset of candidates equals the set of winners in some completion. For the possible winners of Uncovered set (PWU)[i] defined as above, they proved that this problem is $\mathcal{NP}$-hard by a reduction from the satisfiability problem (SAT). In contrast, the problems of deciding whether a given candidate is a possible winner or a necessary winner for Uncovered set are both polynomial-time solvable [12]. Moreover, computing the Uncovered set of a partial tournament is polynomial-time solvable [153].

As for the problems related to Banks set, in spite of the polynomial-time solvability of computing a Banks winner, deciding whether a distinguished candidate is a Banks

[i] The authors use $\mathrm{PSW}_{\mathrm{UC}}$ to denote the problem in [12].

winner is $\mathcal{NP}$-hard [152, 242]. The latter problem is also related to the Dual Directed Feedback Vertex Set (Dual-Dfvs) problem [215]. In Directed Feedback Vertex Set (Dfvs) [215], we are given a directed graph D and a positive integer parameter κ, and asked to decide whether there is a subset of vertices of size κ whose removal results in a directed graph without a cycle. In Dual-Dfvs, we are given a directed graph and a positive integer parameter κ, and asked whether there is a subgraph of size κ containing no cycle. Dfvs has been proved $\mathcal{FPT}$ [65] over a long time of studying. In particular, when restricted to tournament, Dfvs has an $O(\kappa^3)$ kernel [82]. By a dichotomy theorem from [216], Dual-Dfvs is $\mathcal{W}[1]$-hard. However, when restricted to tournaments this problem is $\mathcal{FPT}$ [215]. It is well-known that a tournament contains no cycle if and only if it is transitive. These problems are also related to Slater set problems, where the main task is to reverse minimum number of arcs so that a given tournament become transitive. We refer to [153] for detailed complexity results about problems on Slater set.

5.2 Uncovered Set in Partial Tournaments

It is easy to see that all problems except Pwu defined above are in $\mathcal{XP}$: try all possibilities of selecting a subset of size $\mathcal{R}$ in $V(D)$, $A(D)$ or $\{(v,u) \mid (v,u) \notin A(D)\}$, where $\mathcal{R}$ is the parameter of the corresponding problem. All these algorithms run in $O(|D|^{2\mathcal{R}})$ time, where $|D|$ is the size of the given partial tournament and $\mathcal{R}$ is the related parameter.

In the following, we show that Pwu is also in $\mathcal{XP}$.

Theorem 5.1. Pwu *is in* $\mathcal{XP}$*.*

Proof. We prove the theorem by giving an $\mathcal{XP}$-algorithm. The following lemma is useful. Let $\mathcal{E} = (D = (V, A), X)$ be an instance of Pwu.

Lemma 5.1. *Let $v \in X$ be a serf with respect to X in D and $\mathcal{E}' = (D' = (V, A'), X)$ be a new instance with $A' = A \cup \{(v,u) \mid \{(v,u),(u,v)\} \cap A = \emptyset, u \in V \setminus X\}$, then $\mathcal{E}$ is a yes-instance if and only if $\mathcal{E}'$ is a yes-instance.*

Proof. It is clear that if $\mathcal{E}'$ is a yes-instance, then $\mathcal{E}$ must be a yes-instance. To prove the other direction, note that adding an arc from some vertex $u \in V \setminus X$ to v is to make v reachable by some vertex $w \in X \setminus \{v\}$ through u. However, since v is already a serf with respect to X, such an arc addition is then unnecessary. However, adding the arc (v,u) for $u \in V \setminus X$ to the partial tournament would make v reach further vertices. $\square$

Our algorithm first tries all possibilities of completions of $D[X]$. Clearly, there can be at most $2^{|X|\cdot(|X|-1)/2}$ such possibilities. In each of the completions, there may have some pairs (u, w) with $(u, w) \in A(D[X])$ such that w does not reach u. For all these pairs, we further try all possibilities of making w reach u by some vertex $v \in V \setminus X$ (thus, there are at most $|V \setminus X|$ possibilities for each pair, and in total at most $|V \setminus X|^{|X|\cdot(|X|-1)/2}$ possibilities for all pairs), by adding one or two new arcs between $\{w, u\}$ and v. Meanwhile, if there is no chance to make w reach u, then we give up the possibility. Clearly, if the given instance is a yes-instance, then at least one of the possibilities leads to a "Yes" answer. We have in total at most $2^{|X|\cdot(|X|-1)/2} \cdot |V \setminus X|^{|X|\cdot(|X|-1)/2}$ possibilities to check. Now, in each case, $D[X]$ induces a tournament and every vertex $v \in X$ is a serf with respect to X. Then, due to Lemma 5.1, we can safely add all missing arcs between X and $V \setminus X$ with tails in X and heads in $V \setminus X$. It remains to add arcs between vertices in $V \setminus X$ to make the vertices in X kings. For convenience, let's give a formal definition of the remaining part first.

$\overline{\text{PWU}}$

Input: A partial tournament $D = (V, A)$ and a subset $X \subseteq V$ such that $D[X]$ induces a tournament, every vertex $v \in X$ is a serf with respect to X in D and there is no missing arcs between X and $V \setminus X$, that is, $\{(v, u), (u, v)\} \cap A \neq \emptyset$ for all $v \in X$ and all $u \in V \setminus X$.

Question: Is there a completion of D such that all vertices in X are kings?

In the following, we prove that $\overline{\text{PWU}}$ is solvable in polynomial time. We begin with a useful observation.

Observation. *Let v and u be two vertices in $V \setminus X$ with missing arcs between them. If there is a vertex $x \in X$ such that x can reach v directly but x cannot reach u, then every yes-instance has a solution containing the arc (v, u).*

The observation is correct. The reasons are as follows. First observe that adding an arc (v', u') between $v', u' \in V \setminus X$ to the partial tournament is to make some vertex $w \in X$ reach u' by v'. Since x cannot reach u, all vertices in X which can directly reach u must also directly reach x. Therefore, no vertex in X needs an arc from u to v to reach v; since all such vertices have already reached v by x. Thus, adding (v, u) is the optimal choice.

Based on the above observation, we can solve $\overline{\text{PWU}}$ in polynomial time with Algorithm 5.1.

In summary, PWU is in $\mathcal{XP}$; since there are at most $2^{|X|\cdot(|X|-1)/2} \cdot |V \setminus X|^{|X|\cdot(|X|-1)/2}$ instances of $\overline{\text{PWU}}$ and $\overline{\text{PWU}}$ can be solved in polynomial time. $\square$

Algorithm 5.1: A polynomial-time algorithm for $\overline{\text{PWU}}$

forall the *vertices* $x \in X$ **do**
  Let $V_x = \{v \in V \setminus X \mid (x, v) \in A(D)\}$ be the set of vertices that x can reach directly;
  Let $V_{\bar{x}} = \{v \in V \setminus X \mid (v, x) \in A(D), \nexists y \in V$ with $(x, y) \in A(D)$ and $(y, v) \in A(D)\}$ be the set of vertices that x cannot reach;
  if $V_x = \emptyset$ *and* $V_{\bar{x}} \neq \emptyset$ **then**
    Return “No”;
  else
    forall the $v \in V_x$ *and* $u \in V_{\bar{x}}$ *with* $\{(v, u), (u, v)\} \cap A(D) = \emptyset$ **do**
      Add (v, u) to D;
    end
  end
end
Return “Yes” if all vertices in X are kings and return “No” otherwise;

With the above theorem, we can trivially get the following result.

Corollary 5.1. Pwu *is polynomial-time solvable if the size of the given subset X is bounded by a constant.*

Now we study the problems of deciding whether we can make all vertices of X kings by adding (resp. reversing) at most $\mathcal{R}$ arcs. In particular, we prove that both Pwu-Add and Pwu-Reverse are $\mathcal{W}[2]$-hard even when X contains only one single vertex. Furthermore, our $\mathcal{W}[2]$-hardness proof for Pwu-Reverse applies to the case that the input is a tournament and X contains only one single vertex. These results imply that the problem of finding the minimum number of arcs which are needed to add to the given partial tournament (resp. to reverse in the given (partial) tournament) to make all vertices of X kings is beyond $\mathcal{XP}$, in the case that $|X|$ is the parameter.

Theorem 5.2. Pwu-Add *is $\mathcal{W}[2]$-hard even when $|X| = 1$.*

Proof. We prove the theorem by an $\mathcal{FPT}$-reduction from the Set Cover problem which is $\mathcal{W}[2]$-hard (Theorem 13.29 of [203]).

Set Cover
Input: A base set $S = \{s_1, s_2, ..., s_n\}$ and a collection C of subsets of S, $C = \{c_1, c_2, ..., c_m\}$, $c_i \subseteq S$ for $1 \leq i \leq m$, and $\bigcup_{1 \leq i \leq m} c_i = S$.
Parameter: A positive integer κ
Question: Is there a subset $C' \subseteq C$ of size at most κ which covers all elements in S, that is, $\bigcup_{c \in C'} c = S$?

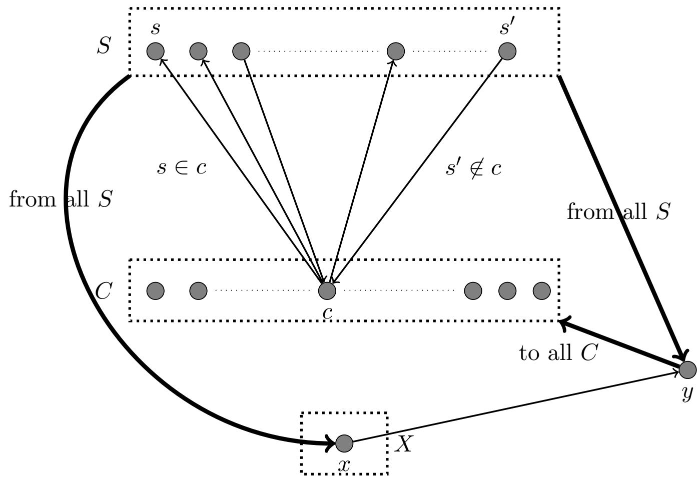


Figure 5.1: Illustration of the $\mathcal{W}[2]$-hardness reduction for Pwu-Add in Theorem 5.2. Here, $D[S]$ and $D[C]$ are made complete arbitrarily. The thick arcs labeled with "from all S" mean that there is an arc (s, x) and an arc (s, y) for all $s \in S$. The thick arc labeled with "to all C" means that there is an arc (y, c) for all $c \in C$. Finally, there is an arc (c, s) if $s \in c$ and an arc (s, c) otherwise, for every $c \in C$ and $s \in S$.

Given an instance $\mathcal{E} = (S, C, \kappa)$ of Set Cover, we construct an instance $\mathcal{E}' = (D = (V, A), X, \mathcal{R})$ of Pwu-Add as follows.

The partial tournament D contains $n + m$ vertices one to one labeled by the elements in $S \cup C$ together with further two vertices $\{x, y\}$. We further use S and C to denote the sets of vertices labeled by the elements in S and C, respectively. For each $c \in C$, there is an arc $(y, c) \in A(D)$. For each $s \in S$, there is an arc $(s, x) \in A(D)$ and an arc $(s, y) \in A(D)$. For each pair $\{s, c\}$ where $s \in S$ and $c \in C$, there is an arc $(c, s) \in A(D)$ if $s \in c$, and an arc $(s, c) \in A(D)$ otherwise. In addition, there is an arc $(x, y) \in A(D)$. Finally, we add arbitrary arcs in $D[S]$ and $D[C]$ to make both $D[S]$ and $D[C]$ complete. We set $X = \{x\}$ and $\mathcal{R} = \kappa$. See Figure 5.1 for an illustration.

Due to the construction, a vertex $c \in C$ can reach a vertex $s \in S$ only if c covers s, that is, $s \in c$. Meanwhile, x can reach every vertex in C by y but cannot reach any vertex in S. In order to make x a king, we must add some arcs from x to C to make x reach all vertices in S. We prove that $\mathcal{E}$ is a yes-instance if and only if $\mathcal{E}'$ is a yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Let C' be a solution of $\mathcal{E}$. Then, it is easy to verify that we can make x a king by adding arcs (x, c) in D for all $c \in C'$; thus, $\mathcal{E}'$ is a yes-instance.

($\Leftarrow$:) Suppose that $\mathcal{E}'$ is a yes-instance and B is a solution for $\mathcal{E}'$. Let $C' = \{c \mid (x, c) \in B\}$ (Due to the construction, we have that $C' \subseteq C$). We claim that C' is a solution for $\mathcal{E}$: the only way to make x reach a vertex $s \in S$ is to add an arc from x to some vertex $c \in C$ with $s \in c$. Since x is a king after adding all arcs in B to $\mathcal{E}$, x can reach every $s \in S$ by at least one vertex $c \in C'$ with $s \in c$, implying C' is a set cover for $\mathcal{E}$. $\square$

Now we study the parameterized complexity of the problem of determining whether we can make a certain set of vertices kings by reversing at most $\mathcal{R}$ arcs.

Theorem 5.3. Pwu-Reverse *is* $\mathcal{W}[2]$*-hard, even when the input is a tournament and* X *contains only a single vertex.*

Proof. We prove the theorem by an $\mathcal{FPT}$-reduction from Dominating Set on Tournaments which is $\mathcal{W}[2]$-hard [86].

> Dominating Set on Tournaments (Dst)
> *Input:* A tournament T.
> *Parameter:* A positive integer κ.
> *Question:* Does T have a dominating set of size at most κ? Here, a dominating set C for a tournament T is a subset of the vertices of T such that every vertex outside C has at least one of its in-neighbors in C.

Given an instance $\mathcal{E} = (T, \kappa)$ of Dst, we construct an instance $\mathcal{E}' = (T', X = \{x\}, \mathcal{R} = \kappa)$ for Pwu-Reverse as follows. T' contains a copy of T, which is denoted by $\bar{T}$, together with a further vertex x to which there is an arc from every vertex in $\bar{T}$, that is, $(\bar{v}, x) \in A(T')$ for all $\bar{v} \in V(\bar{T})$. We use $\bar{v}$ to refer to the copy of the vertex $v \in V(T)$. This complete the construction.

It is easy to verify that if T has a dominating set C of size at most κ, then reversing the arcs $\{(\bar{v}, x) \mid v \in C\}$ makes x a king. To check the other direction, first observe that if $\mathcal{E}'$ is a yes-instance, then there is a solution such that all reversed arcs are between x and $V(\bar{T})$. The observation is correct since each reversal of an arc $(\bar{v}, \bar{u})$ with $\bar{v}, \bar{u} \in V(\bar{T})$ can be replaced by a reversal of the arc $(\bar{v}, x)$ to form a new solution. Now suppose that $\mathcal{E}'$ is a yes-instance and B is a solution (represented by a set containing all reversed arcs) containing only arcs between x and $V(\bar{T})$. Let T'' be the tournament obtained from T' by reversing all arcs in B. We claim that $C = \{v \mid (\bar{v}, x) \in B\}$ is a dominating set of T (the size of C is clearly at most κ). To this end, we need to show that, in the tournament T, every vertex which is not in C has at least one of its in-neighbors in C. Let u be any arbitrary vertex in $V(T) \setminus C$. Due to the construction, there is an arc $(\bar{u}, x)$ in T''. Since x is a king in T'', we know that x reaches $\bar{u}$ by some vertex $\bar{v}$ with $(x, \bar{v}) \in T''$. Due to the construction, $(x, \bar{v})$ is in T'' only if $(\bar{v}, x)$ is in B, or equivalently, $v \in C$. Since $(\bar{v}, \bar{u}) \in A(\bar{T})$ and $\bar{T}$ is a copy of T, $(v, u) \in A(T)$. Therefore, we can conclude that every vertex u outside C has at least one vertex $v \in C$ with $(v, u) \in A(T)$, which completes the proof. □

5.3 Banks Set in Partial Tournaments

In this section, we study problems of deciding whether a distinguished vertex p is contained in a maximal transitive subtournament with p being the 0-indegree vertex.

We first prove that Tw is $\mathcal{W}[2]$-hard by an $\mathcal{FPT}$-reduction from the Multicolored Set Cover problem.

> κ-Multicolored Set Cover, (κ-Msc)
> *Input:* A base set $S = \{s_1, s_2, ..., s_n\}$ and a collection $C = \{c_1, c_2, ..., c_m\}$ of subsets of S, where each $c_i \in C$ has a color from $\{1, 2, ..., \kappa\}$, and moreover, $\bigcup_{1 \le i \le m} c_i = S$.
> *Parameter:* κ
> *Question:* Is there a subset $C' \subseteq C$ such that C' includes exactly one from the same colored subsets and C' covers all elements of S, that is, $\bigcup_{c \in C'} c = S$? We call such a C' a κ-multicolored set cover.

Lemma 5.2. κ-Msc *is* $\mathcal{W}[2]$*-hard.*

Proof. We prove the theorem by an $\mathcal{FPT}$-reduction from Set Cover. Given an instance $\mathcal{E} = (S, C, \kappa)$ of Set Cover, we construct a collection $\overline{C}$ by taking κ copies $\overline{c}_1, \overline{c}_2, ..., \overline{c}_\kappa$ of each $c \in C$, and then color each $\overline{c}_i$ with color $i \in \{1, 2, ..., \kappa\}$. The constructed instance for κ-Msc is $\mathcal{E}' = (S, \overline{C}, \kappa)$. It is straightforward to verify that $\mathcal{E}$ has a set cover of size κ if and only if $\mathcal{E}'$ has a multicolored set cover of size κ. $\square$

With the $\mathcal{W}[2]$-hardness of κ-Msc we now prove the $\mathcal{W}[2]$-hardness of Tw.

Theorem 5.4. Tw *is* $\mathcal{W}[2]$*-hard.*

Proof. We prove the theorem by an $\mathcal{FPT}$-reduction from κ-Msc. Given an instance $\mathcal{E} = (C, S, \kappa)$ of κ-Msc where C is the colorful collection, S is the base set and κ is the parameter, we construct an instance $\mathcal{E}' = (D = (V, A), p, \mathcal{R})$ of Tw as follows. Let C_i be the collection of subsets in C colored by $i \in \{1, 2, ..., \kappa\}$.

The partial tournament D contains $n + m$ vertices one to one labeled by the elements in $S \cup C$ together with the distinguished vertex p. We further use S and C to denote the sets of vertices labeled by the elements in S and C, respectively. For every $s \in S$ and $c \in C$, there is an arc from c to s if $s \in c$ and an arc from s to c otherwise. In addition, there is an arc from s to p for all $s \in S$ and an arc from p to c for all $c \in C$. Finally, there is an arc (c, c') for all $c \in C_i$ and $c' \in C_j$ with $i < j$. The parameter is $\mathcal{R} = \kappa + 1$. See Figure 5.2 for an illustration. We now prove that $\mathcal{E}$ is a yes-instance if and only if $\mathcal{E}'$ is a yes-instance.

($\Rightarrow$:) Suppose that $\mathcal{E}$ is a yes-instance. Let C' be a solution of $\mathcal{E}$. Clearly, $C' \cup \{p\}$ induces a transitive tournament with p being the 0-indegree vertex. Due to the construction, for each vertex $s \in S$, C' contains at least one of its in-neighbors; thus, no vertex in S can be added to $C' \cup \{p\}$ to make a bigger transitive tournament (since otherwise, there would be a triangle), implying that $C' \cup \{p\}$ is maximal in D.

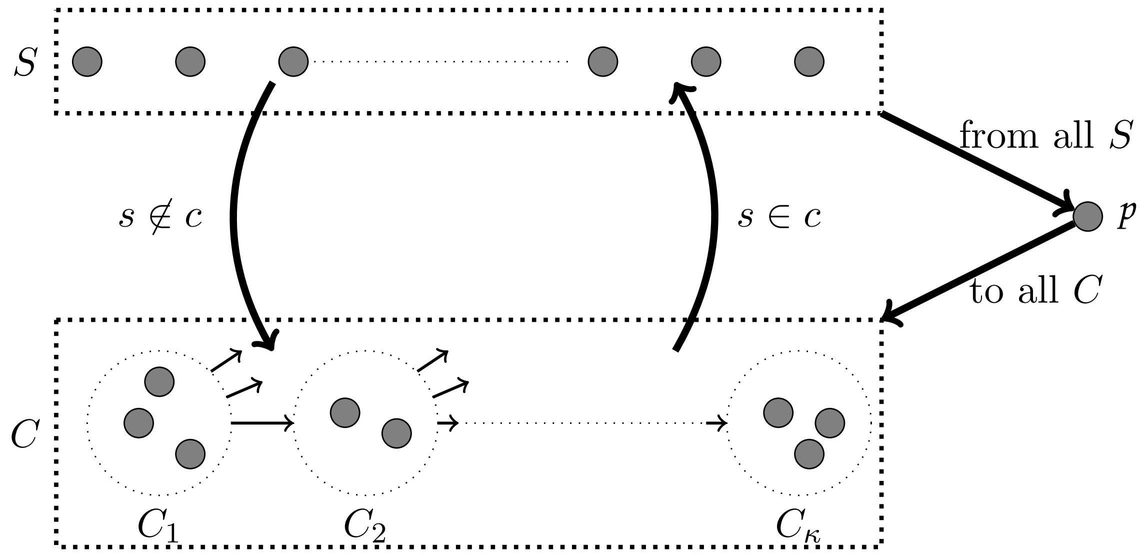


Figure 5.2: Illustration of the $\mathcal{W}[2]$-hardness reduction for Tw in Theorem 5.4.

($\Leftarrow$:) Suppose that $\mathcal{E}'$ is a yes-instance. Let $B \cup \{p\}$ be a solution of $\mathcal{E}'$ which induces a maximal transitive tournament with p being the 0-indegree vertex. Clearly, $B \subseteq C$. Due to the maximality of $D[B \cup \{p\}]$, $N^-(s) \cap B \neq \emptyset$ for all $s \in S$, implying that at least one subset in B covers s; thus, B must be a set cover of $\mathcal{E}$. Due to the construction, there is no arc in $D[C_i]$ for all $i \in \{1, 2, ..., \kappa\}$; thus, exactly one from each C_i can be in B. Therefore, B must be a κ-multicolored set cover for D. □

In the following, we study two further parameterizations of the problem of finding a Banks winner in a partial tournament. First, we study the parameter $|N^-(p)|$, that is, the number of candidates who beat p in a pairwise comparison. We show that this problem is $\mathcal{W}[1]$-hard.

Theorem 5.5. Tw-Indegree *is* $\mathcal{W}[1]$*-hard.*

Proof. We prove the theorem by an $\mathcal{FPT}$-reduction from κ-Multicolored Clique which is $\mathcal{W}[1]$-hard [117]. A *clique* Q (resp. An *independent set* I) of a graph G is a subset of $V(G)$ such that there is an (resp. no) edge between every pair of vertices in Q (resp. I).

> κ-Multicolored Clique
> *Input*: A vertex-colored undirected graph $G = (V, E)$, where each vertex has a color from $\{1, 2, ..., \kappa\}$.
> *Parameter:* κ.
> *Question:* Does G have a clique including vertices of all κ colors?

Let $\mathcal{E} = (G, \kappa)$ be an instance of κ-Multicolored Clique. Let V_i be the set of all vertices in G with color i. Due to the definition of the κ–Multicolored

Clique, we can safely assume that each V_i form an independent set of the graph G [117]. We construct an instance $\mathcal{E}' = (D = (N^-(p) \cup N^+(p) \cup \{p\}, A), p, |N^-(p)|)$ for Tw-Indegree from $\mathcal{E}$ as follows.

We create the vertices as follows. For each $v \in V(G)$ we create a vertex in D. For ease of exposition, we still use v to denote this vertex in D. In addition, for each color in $i \in \{1, 2, ..., \kappa\}$, we create a vertex c_i. Therefore, together with the distinguished candidate p, we have in total $|V(G)| + \kappa + 1$ vertices. The arcs are created as follows. For each $v \in V(D)$, we create an arc (p, v). For each vertex c_i, we create an arc (c_i, p). Therefore, we have that $N^+(p) = V(D)$ and $N^-(p) = \{c_1, c_2, ..., c_\kappa\}$. In addition, for each c_i, there is an arc (v, c_i) for all $v \in V_i$ and an arc (c_i, v) for all $v \in V_j$ with $j \neq i$. Finally, we create some arcs between V_i and V_j for $i \neq j$. Precisely, for two vertices $v \in V_i$ and $u \in V_j$ with $1 \leq i < j \leq \kappa$, there is an arc (v, u) in D if there is an edge between v and u in G. See Figure 5.3 for an illustration. In the following, we prove that $\mathcal{E}$ is a yes-instance if and only if $\mathcal{E}'$ is a yes-instance.

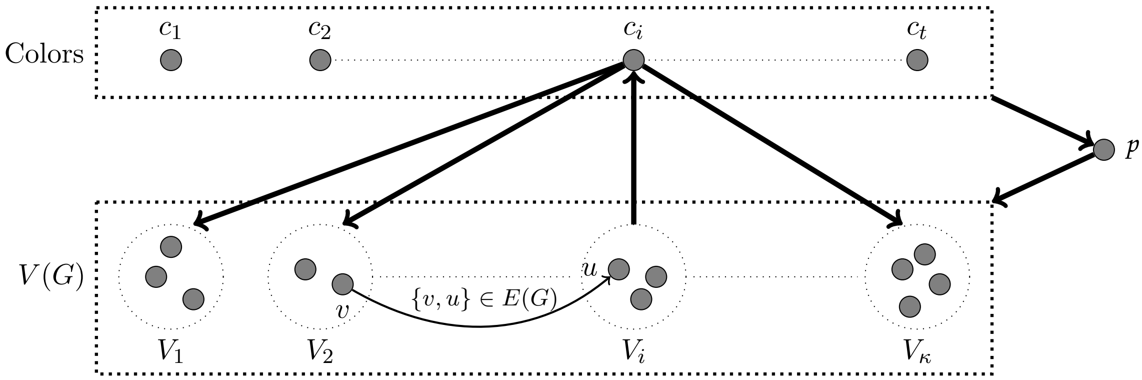


Figure 5.3: Illustration of the $\mathcal{W}[1]$-hardness reduction for Tw-Indegree in Theorem 5.5.

($\Rightarrow$:) Suppose that $\mathcal{E}$ is a yes-instance and Q is a clique including all κ colors, that is $\{u, v\} \in E$ for all $u, v \in Q$ and $|Q \cap V_i| = 1$ for all $1 \leq i \leq \kappa$. Due to the construction, Q induces a transitive tournament in D. Moreover, the induced transitive tournament is maximal in $D[N^+(p)]$ since there is no arc in $D[V_i]$ for all $1 \leq i \leq \kappa$. Since $Q \cap V_i \neq \emptyset$ and $V_i = N^-(c_i)$ for all $1 \leq i \leq \kappa$, every c_i has an in-neighbor in Q; thus, $D[Q \cup \{p\}]$ is a maximal transitive tournament in D with p being the 0-indegree vertex.

($\Leftarrow$:) Suppose that $\mathcal{E}'$ is a yes-instance and $Q \cup \{p\}$ induces a maximal transitive tournament in D with p being the 0-indegree vertex. Due to the construction, Q induces a clique in G. Since there is no arc in each $D[V_i]$ for all $1 \leq i \leq \kappa$, there can be at most one vertex of V_i in Q. Due to the maximality of $D[Q \cup \{p\}]$, for every V_i ($1 \leq i \leq \kappa$), at least one vertex of V_i must be in Q (since otherwise, c_i can be added to $D[Q \cup \{p\}]$ to form a bigger transitive subtournament). In summary, we conclude that Q is a clique of G including all colors. $\square$

The last parameter we study is $|N^+(p)|$, that is, the Copeland^0 score of p.

Theorem 5.6. Tw-Outdegree *is* $\mathcal{FPT}$*.*

The proof for Theorem 5.6 is straightforward: if there is a solution, it must be totally included in $N^+(p) \cup \{p\}$. Thus, the problem can be solved by enumerating all $2^{|N^+(p)|}$ subsets of $N^+(p)$, and checking whether at least one of them together with p forms a maximal transitive tournament with p being the 0-indegree vertex. The algorithm implies a $2^{|N^+(p)|}$-size vertex-kernel: if the input partial tournament D contains at most $2^{|N^+(p)|}$ vertices then we are done; otherwise, solve the problem in polynomial time (note that $2^{|N^+(p)|} \leq |V(D)|$) and return a trivial yes-instance or a trivial no-instance according to the output of the algorithm. A kernel of exponential size is far from satisfactory and thus a natural question arias: can the kernel be improved greatly? The following theorem answers the question negatively.

Theorem 5.7. Tw-Outdegree *does not admit a polynomial kernel unless the polynomial hierarchy collapses to the third level* ($\mathcal{PH} = \sum_{\mathcal{P}}^{3}$)*.*

Proof. We prove the theorem via polynomial parameter reduction technique as discussed in Section 1.3.2. Recall that in order to show the non-existence of a polynomial kernel for a specific problem Q, it suffices to derive a polynomial parameter reduction from a parameterized problem which does not have a polynomial kernel (under some assumption which is unlikely to happen) to Q (see Lemma 1.1).

In fact, the reduction from κ-Msc to Tw in the proof of Theorem 5.4 has already implied that Tw-Outdegree does not admit a polynomial kernel. This is because that the κ-Msc with parameter $|C|$, the size of the collection of subsets, is $\mathcal{FPT}$ but does not admit a polynomial kernel, unless the polynomial hierarchy collapses to the third level. Formally, the following problem is $\mathcal{FPT}$ but does not admit a polynomial kernel unless the polynomial hierarchy collapses to the third level.

> κ-Multicolored Set Cover-$|C|$, (κ-Msc-$|C|$)
> *Input:* A base set $S = \{s_1, s_2, ..., s_n\}$ and a collection $C = \{c_1, c_2, ..., c_m\}$ of subsets of S, where each $c_i \in C$ has a color from $\{1, 2, ..., \kappa\}$, and moreover, $\bigcup_{1 \leq i \leq m} c_i = S$.
> *Parameter:* $|C|$
> *Question:* Is there a subset $C' \subseteq C$ such that C' includes exactly one from the same colored subsets and C' covers all elements of S?

Lemma 5.3. *[83]* κ-Msc-$|C|$ *has no polynomial kernel unless the polynomial hierarchy collapses to the third level.*

The following lemma directly follows from the proof of Theorem 5.4.

Lemma 5.4. $|C|$-Msc *is polynomial parameter reducible to* Tw-Outdegree*.*

Theorem 5.7 directly follows from Lemmas 1.1, 5.3 and 5.4. □

5.4 Conclusion

In this chapter, we have studied some possible winner(s) problems related to Uncovered set and Banks set on partial tournaments, from the viewpoint of parameterized complexity. We have showed some $\mathcal{XP}$ results, $\mathcal{W}$-hardness results as well as $\mathcal{FPT}$ results along with a kernelization lower bound. Our results are summarized in Table 5.1.

There remains one intriguing open problem for further research: is Pwu $\mathcal{FPT}$?

6

Combinatorial Algorithms for Borda Manipulation

Manipulation is another widely studied strategic behavior. In this setting, we are given a set of candidates, a set of votes, a distinguished candidate and a set of voters who have not cast their votes yet. The problem is whether these voters can cast their votes in a way so that the given distinguished candidate wins the election. This chapter is concerned with manipulation in Borda voting.

6.1 Introduction

This chapter is devoted to deriving combinatorial algorithms for the Borda manipulation problems. In the *manipulation* problem, we are given an election consisting of a set of candidates and a multiset of votes cast by a set of voters, a distinguished candidate and a set of voters who have not cast their votes yet. These voters who have not cast their votes are called *manipulators*. The question is whether the manipulators can cast their votes, referred to as *manipulative votes*, in a way so that the given distinguished candidate wins the election. In the *weighted manipulation* problem, each vote (or voter) is associated with a positive integer weight w. Moreover, a vote with weight w is regarded as w individual votes each with weight 1. Therefore, the unweighted manipulation is a special case of the weighted manipulation with each vote having weight 1.

Both the weighted and the unweighted Borda manipulation problems are $\mathcal{NP}$-hard [26, 68, 75]. In particular, the unweighted Borda manipulation is $\mathcal{NP}$-hard even when there are only two manipulators and three non-manipulative votes [26, 75] (but the number of candidates is part of the input), and the weighted Borda manipulation is $\mathcal{NP}$-hard even when there are only three candidates (but the number of manipulators is part of the input) [68]. By enumerating all possibilities, the unweighted Borda manipulation problem can be solved in $O(m!^t)$ time [26], where m is the number of candidates and t is the number of manipulators. Betzler, Niedermeier and Woeginger posed an open question in [26] whether the unweighted Borda manipulation with two manipulators can be solved in single-exponential time with respect to the number of candidates. Recall that a problem is solvable in single-exponential time with respect to some parameter κ if there exists an algorithm solving it in time $2^{O(\kappa)} \cdot |I|^{O(1)}$, where I is the input. Deriving or improving single-exponential algorithms for intractable combinatorial optimization problems is of particular importance and has received a considerable attention recently [24, 31, 85, 167, 168, 194]. Many single-exponential algorithms have been proved practical for instances of moderate sizes. A prominent example is the Vertex Cover problem which starts from an $O^*(2^\kappa \kappa^{2\kappa+2})$-time[i] algorithm [56], and then, after many rounds of improvement, it turns out that this problem admits a single-exponential algorithm of running time $O^*(1.2738^\kappa)$ [64] which has been shown very efficient for κ up to 400 [62, 116]. Here, κ denotes the size of the vertex cover. We refer to [62, 116, 243] for further discussions on this issue.

In this chapter, we answer the question asked in [26] affirmatively by deriving combinatorial algorithms for both the weighted and unweighted Borda manipulation problems. Our algorithms remain single-exponential even for the weighted manipulation problem, with respect to the number of manipulators or the number of candidates

[i] $O^*()$ is the $O()$ notation with suppressed factors polynomial in the size of the input.

whenever one of these two parameters is bounded by a constant. Therefore, we not only answer the open question in [26], but also answer several more general questions.

6.1.1 Preliminaries

In this chapter, we use bijections to denote votes. Formally, a vote in this chapter will be represented by a bijection $\pi_v : \mathcal{C} \to [|\mathcal{C}|]$, where $[n]$ denotes the set $\{1, 2, ..., n\}$. The value of $\pi_v(c)$ for a candidate c is the number of candidates ranked below c plus one. For example, a vote with preference $a \succ b \succ c$ is represented by a bijection π with $\pi(a) = 3, \pi(b) = 2$ and $\pi(c) = 1$. Notice that for a candidate c and a vote π with $\pi(c) = i$, the position of the candidate c in the vote π is $m - j + 1$, where m is the number of candidates.

In the following, let m denote the number of candidates. The *Borda correspondence* (see also Section 1.2.1) can be defined by a vector $\langle m-1, m-2, ..., 0\rangle$. Each voter contributes $m-1$ points to his most preferred candidate, $m-2$ to his second preferred candidates, and so on. The candidates who have the highest total score are the winners. In the *weighted Borda system*, each voter v is associated with a positive integer weight $f(v)$ and contributes $f(v) \cdot (m-1)$ points to his most preferred candidate, $f(v) \cdot (m-2)$ to his second preferred candidate, and so on. Accordingly, candidate having the highest total score win the election. Therefore, the unweighted Borda system is a special case of the weighted Borda system with $f(v) = 1$ for every voter v.

For a candidate c and a voter v, we use $BSC_v(c)$ to denote the Borda score of c contributed by v, that is, $BSC_v(c) = f(v) \cdot (\pi_v(c) - 1)$. Let $BSC_{\mathcal{V}}(c)$ denote the total score of c contributed by voters in $\mathcal{V}$, that is, $BSC_{\mathcal{V}}(c) = \sum_{v \in \mathcal{V}} BSC_v(c)$.

In the settings of *manipulation*, we have, in addition to $\mathcal{V}$, a set $\mathcal{V}'$ of voters which are called *manipulators*. The manipulators form a coalition and desire to coordinate their votes to make a distinguished candidate win the new election with votes in $\Pi_{\mathcal{V}} \uplus \Pi_{\mathcal{V}'}$, where $\Pi_{\mathcal{V}'}$ is the multiset of votes cast by the manipulators. As in the previous chapters, we distinguish between the unique-winner model and the nonunique-winner model. However, for simplicity, our algorithms are mainly described for the unique-winner model in this chapter. All algorithms in this chapter can be easily adapted to the nonunique-winner model. The formal definitions of the problems studied in this chapter are as follows.

Unweighted Borda Manipulation (UM-Borda)

Input: An election $(\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}, \mathcal{V})$ where p is not the unique winner, and a set $\mathcal{V}'$ of t manipulators.

Question: Can the manipulators cast their votes in a way so that p becomes the unique winner in the election $(\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}} \uplus \Pi_{\mathcal{V}'})$, where $\Pi_{\mathcal{V}'}$ with $|\Pi_{\mathcal{V}'}| = t$ is the multiset of votes cast by the manipulators?

Weighted Borda Manipulation (WM-Borda)

Input: A weighted election $(\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}, \mathcal{V}, f_1 : \mathcal{V} \to \mathbb{N})$ where p is not the unique winner, a set $\mathcal{V}'$ of t manipulators and a weight function $f_2 : \mathcal{V}' \to \mathbb{N}$.

Question: Can the manipulators cast their votes $\Pi_{\mathcal{V}'}$ in a way so that p is the unique winner in the weighted election $(\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}} \uplus \Pi_{\mathcal{V}'}, \mathcal{V} \cup \mathcal{V}', f : \mathcal{V} \cup \mathcal{V}' \to \mathbb{N})$, where $f(v) = f_1(v)$ if $v \in \mathcal{V}$ and $f(v) = f_2(v)$ otherwise, and $\Pi_{\mathcal{V}'}$ with $|\Pi_{\mathcal{V}'}| = t$ is the multiset of votes cast by the manipulators?

6.1.2 Related Works

As one of the most prominent voting systems, complexity of strategic behavior for Borda has been intensively studied. It is known that many types of bribery and control behavior for Borda are $\mathcal{NP}$-hard [54, 98, 99, 223]. For manipulation, WM-Borda is $\mathcal{NP}$-hard even when the election contains only three candidates [68]. Bartholdi, Tovey and Trick [156] showed that both UM-Borda and WM-Borda are polynomial-time solvable if there is only one manipulator. The complexity of UM-Borda in the case of more than one manipulator remained open for many years, until very recently it was proved $\mathcal{NP}$-hard even when there are only two manipulators and three non-manipulators [26, 75]. Heuristic and approximation algorithms for UM-Borda have been studied in the literature [73, 75, 258]. It is worth mentioning that Zuckerman, Procaccia and Rosenschein [258] proposed an approximation algorithm for UW-Borda which can output a success manipulation with $t + 1$ manipulators whenever the given instance has a success manipulation with t manipulators. By applying the integer linear programming (ILP) technique, UM-Borda can be solved exactly with a very high computational complexity $O^*(m!^{O(m!)})$ [26], where m is the number of candidates. Prior to the work of this thesis, no purely combinatorial exact algorithm is known for UM-Borda and WM-Borda (except the very brute force one which checks all possibilities). In particular, Betzler, Niedermeier and Woeginger [26] posed as an open problem whether UM-Borda can be solved exactly with a running time single-exponentially depending on m in the case of two manipulators.

We propose two algorithms solving WM-Borda and UM-Borda in $O^*((m \cdot 2^m)^{t+1})$ time and $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$ time, respectively, where t is the number of manipulators and m is the number of candidates. Both algorithms rely on dynamic programming techniques. Our results imply that both WM-Borda and UM-Borda can be solved in time single exponentially on m in the case of constant number of manipulators. In particular, for $t = 2$, we have an algorithm with running time $O^*(3^m)$ for UM-Borda, affirmatively answering the open question in [26]. In fact, when either m or t is a constant, our algorithms are single-exponential algorithms. See Table 6.1 for a summary of our results concerning combinatorial algorithms for UM-Borda and WM-Borda. In addition to combinatorial algorithms, we improve the running time of the ILP-based algorithm for UM-Borda to $O^*(2^{9m^2 \log m})$.

	WM-Borda	UM-Borda
m and t are not constants	$O^*((m \cdot 2^m)^{t+1})$	$O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$
m is a constant	$O^*(a^t)$	Poly(t)
t is a constant	$O^*(b^m)$	$O^*(c^m)$

Table 6.1: Running time of the combinatorial algorithms for the weighted and unweighted Borda manipulation problems. Here, m and t are the number of candidates and the number of manipulators, respectively. Moreover, a, b, c are constants, and Poly(t) is a polynomial function in t.

6.2 Algorithm for Weighted Borda Manipulation

In this section, we present an exact combinatorial algorithm for WM-Borda. The following observation is clearly true.

Observation 6.1. *Every yes-instance of WM-Borda has a solution where the distinguished candidate p is ranked in the top in every manipulative vote.*

Let $((\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}, \mathcal{V}, f_1), \mathcal{V}', f_2, t)$ be the given instance. Due to Observation 6.1, there must be a solution $\Pi_{\mathcal{V}'}$ with $BSC_{\mathcal{V}\cup\mathcal{V}'}(p) = BSC_{\mathcal{V}}(p) + \sum_{v' \in \mathcal{V}'} f(v') \cdot |\mathcal{C}|$ if the given instance is a yes-instance. Therefore, to make p the unique winner, $BSC_{\mathcal{V}'}(c) \leq$

$g(c)$ should hold for all $c \in \mathcal{C}$, where $g(c) = BSC_{\mathcal{V}}(p) + \sum_{v' \in \mathcal{V}'} f(v') \cdot |\mathcal{C}| - BSC_{\mathcal{V}}(c) - 1$. The value of $g(c)$ is called the *capacity* of c. Meanwhile, if in the given instance there is a candidate c with $g(c) < 0$, then the given instance must be a no-instance. Therefore, in the following, we assume that the given instance contains no candidate c with $g(c) < 0$. Based on these, we can reformulate WM-Borda as follows:

Reformulation of WM-Borda

Input: A set $\mathcal{C}$ of candidates, a capacity function $g : \mathcal{C} \to \mathbb{N}$, and a multiset $F = \{f_1, f_2, ..., f_t\}$ of non-negative integers.

Question: Is there a multiset $\Pi = \{\pi_1, \pi_2, ..., \pi_t\}$ of bijections mapping from $\mathcal{C}$ to $[|\mathcal{C}|]$ such that $\sum_{i=1}^{t} f_i \cdot (\pi_i(c) - 1) \leq g(c)$ holds for all $c \in \mathcal{C}$?

Here, the bijection π_i corresponds to the vote cast by the i-th manipulator and $f_i \in_+ F$ corresponds to the weight of the i-th manipulator (suppose that a fixed order over the manipulators is given). Notice that in the Reformulation of WM-Borda, we do not have the distinguished candidate p. But we have taken the final Borda score of the distinguished candidate p into account in the above reformulation. This is reflected by the capacity function g.

Our algorithm is based on a dynamic programming method which is associated with a boolean dynamic table defined as $DT(C, Z_1, Z_2, ..., Z_t)$, where $C \subseteq \mathcal{C}$ is a subset of candidates, $Z_i \subseteq [|\mathcal{C}|]$ and $|C| = |Z_i|$ for all $i \in [t]$. Here, each Z_i encodes the positions that are occupied by the candidates of C in the vote cast by the i-th manipulator. In particular, a $z \in Z_i$ corresponds to the $(|\mathcal{C}| - z + 1)$-th position of the i-th manipulative vote. The entry $DT(C, Z_1, Z_2, ..., Z_t) = 1$ means that there is a multiset $\Pi = \{\pi_1, \pi_2, ..., \pi_t\}$ of bijections mapping from $\mathcal{C}$ to $[|\mathcal{C}|]$ such that for each $i \in [t]$, $\bigcup_{c \in C} \{\pi_i(c)\} = Z_i$, and moreover, for every candidate $c \in C$, c is "safe" under Π. Here, we say a candidate c is safe under Π, if $\sum_{i=1}^{t} f_i \cdot (\pi_i(c) - 1) \leq g(c)$. Intuitively, $DT(C, Z_1, Z_2, ..., Z_t) = 1$ means that we can place all candidates of C in the positions encoded by Z_i for all $i \in [t]$ without exceeding the capacity of any $c \in C$. Clearly, a given instance is a yes-instance if and only if $DT(\mathcal{C}, Z_1 = [|\mathcal{C}|], Z_2 = [|\mathcal{C}|], ..., Z_t = [|\mathcal{C}|]) = 1$. A formal description of the algorithm is shown in Algorithm 6.1.

Theorem 6.1. *WM-Borda is solvable in $O^*((m \cdot 2^m)^{t+1})$ time, where m is the number of candidates.*

Proof. We consider Algorithm 6.1 for WM-Borda. Let $\mathcal{C}$ be the set of candidates and $m = |\mathcal{C}|$ be the number of candidates. In the *Initialization*, we check whether $\sum_{i=1}^{t} f_i \cdot (z_i - 1) \leq g(c)$ for each candidate $c \in \mathcal{C}$ and each encoded position $z_i \in [m]$ for each $i \in [t]$. Since there are m many candidates and m many positions to be considered for each z_i, the running time of the Initialization is bounded by $O^*(m^{t+1})$. In the *Updating*, we compute $DT(C, Z_1, Z_2, ..., Z_t)$ for all $C \subseteq \mathcal{C}$ and

Algorithm 6.1: An exact combinatorial algorithm for WM-Borda.

Input : An instance $(\mathcal{C}, g, F)$ of the Reformulation of WM-Borda.
Output : "Yes" if the given instance a yes-instance, and "No" otherwise.

/* Initialization */
forall the $c \in \mathcal{C}$ *and* $z_1, z_2, ..., z_t \in [|\mathcal{C}|]$ **do**
  if $\sum_{i=1}^{t} f_i \cdot (z_i - 1) \leq g(c)$ **then**
    $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\}) := 1;$
  else
    $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\}) := 0;$
  end
end
/* Updating $DT(C, Z_1, Z_2, ..., Z_t)$ */
forall the $\ell = 2$ *to* $|\mathcal{C}|$ **do**
  forall the $C \subseteq \mathcal{C}$ *and all* $Z_i \subseteq [|\mathcal{C}|]$ *for every* $i = 1, 2, ..., |\mathcal{C}|$ *with* $|C| = |Z_i| = \ell$ **do**
    if $\exists c \in C$ *and* $\exists z_i \in Z_i$ *for all* $i \in [t]$ *such that* $DT(C \setminus \{c\}, Z_1 \setminus \{z_1\}, Z_2 \setminus \{z_2\}, ..., Z_t \setminus \{z_t\}) = 1$ *and* $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\}) = 1$ **then**
      $DT(C, Z_1, Z_2, ..., Z_t) := 1;$
    else
      $DT(C, Z_1, Z_2, ..., Z_t) := 0;$
    end
  end
end
if $DT(\mathcal{C}, Z_1 = [|\mathcal{C}|], Z_2 = [|\mathcal{C}|], ..., Z_t = [|\mathcal{C}|]) = 1$ **then**
  Return "Yes";
else
  Return "No";
end

all $Z_1 \subseteq [m], Z_2 \subseteq [m], ..., Z_t \subseteq [m]$ with $|C| = |Z_1| = |Z_2| = ... = |Z_t| = \ell$, where $2 \leq \ell \leq m$. To compute each of them, we consider all possibilities of $c \in C$ and $z_1 \in Z_1, z_2 \in Z_2, ..., z_t \in Z_t$. For each possibility, we further check whether $DT(C \setminus \{c\}, Z_1 \setminus \{z_1\}, Z_2 \setminus \{z_2\}, ..., Z_t \setminus \{z_t\}) = 1$ and $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\}) = 1$. Since there are at most m^{t+1} such possibilities, and there are at most $2^{(t+1)m}$ entries needed to be computed, we arrive at the total running time of $O^*((m \cdot 2^m)^{t+1})$.

To check the correctness, recall that $DT(C, Z_1, Z_2, ..., Z_t) = 1$ means we can place all candidates of C in the positions encoded by Z_i for all $i \in [t]$ without exceeding the capacity of any $c \in C$. According to this, $DT(C, Z_1, Z_2, ..., Z_t)$ is equal to 1 whenever there exist $c \in C$ and $z_i \in Z_i$ for all $i \in [t]$ such that $DT(C \setminus \{c\}, Z_1 \setminus \{z_1\}, Z_2 \setminus \{z_2\}, ..., Z_t \setminus \{z_t\}) = 1$ and $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\}) = 1$. This corresponds exactly to the recurrence for updating the dynamic table. In the Initialization, we set $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\})$ to 1 if $\sum_{i=1}^{t} f_i \cdot (z_i - 1) \leq g(c)$ which means that we can place c in the positions encoded by $z_1, z_2, ..., z_t$ in $\pi_1, \pi_2, ..., \pi_t$, respectively, without exceeding the capacity of c. Thus, each value of $DT(\{c\}, \{z_1\}, \{z_2\}, ..., \{z_t\})$ follows the meaning of what we defined for the dynamic table. Finally, it is obvious that a given WM-Borda instance is a yes-instance if and only if $DT(\mathcal{C}, Z_1 = [m], Z_2 = [m], ..., Z_t = [m]) = 1$. □

Algorithm 6.1 applies to the nonunique-winner model. However, in the nonunique-winner model, we require that $\sum_{i=1}^{t} f_i \cdot (\pi_i(c) - 1) \leq g(c) + 1$ holds for all $c \in \mathcal{C}$. Therefore, we need to replace Line 2 in Algorithm 6.1 with the following line.

if $\sum_{i=1}^{t} f_i \cdot (z_i - 1) \leq g(c) + 1$ **then**

Betzler, Niedermeier and Woeginger [26] posed as an open question whether UM-Borda in case of two manipulators can be solved in single-exponential time with respect to the number of candidates. By Theorem 6.1, we can answer this question affirmatively.

Corollary 6.1. *WM-Borda (UM-Borda is a special case of WM-Borda) in case of two manipulators can be solved in $O^*(8^m)$ time, where m is the number of candidates.*

In fact, Theorem 6.1 implies a more general result: WM-Borda is solvable in single-exponential time with respect to m if t is a constant, and with respect to t if m is a constant, where m and t are the number of candidates and the number of manipulators, respectively.

6.3 Algorithm for Unweighted Borda Manipulation

In this section, we study the UM-Borda problem. Recall that UM-Borda is a special case of WM-Borda where all voters have the same unit weight. The specialization offers us an simper way to calculate Borda scores of candidates. In particular, in the unweighted Borda system, when compute $BSC_{\mathcal{V}'}(c)$ for a candidate c, it is irrelevant which manipulators placed c in the j-th positions. The decisive factor is the number of manipulators placing c in the j-th positions. This leads to the following approach where we firstly reduce UM-Borda to a matrix problem and then solve this matrix problem by a dynamic programming algorithm, resulting in a better running time than that in Section 6.2. Firstly, the matrix problem is defined as follows.

Filling Magic Matrix (FMM)

Input: A multiset $g = \{g_1, g_2, ..., g_m\}$ of non-negative integers and an integer $t > 0$.

Question: Is there an $m \times m$ matrix M with non-negative integers such that:

(1) $\forall i \in [m],\ \sum_{j=1}^{m} (j-1) \cdot M[i][j] \leq g_i$;

(2) $\forall i \in [m],\ \sum_{j=1}^{m} M[i][j] = t$; and

(3) $\forall j \in [m], \sum_{i=1}^{m} M[i][j] = t$?

Using matrix to solve the manipulation problem has also been considered by Davies et al. [75]. In this paper, the authors used an n by m relaxed manipulation matrix to devise several heuristic algorithms for the unweighted manipulation problem under Borda, Baldwin's and Nanson's voting correspondences, where n denotes the number of voters and m the number of candidates. The entry $A[i,j]$ defined in their matrix A is the score that the i-th voter gives to the j-th candidate (the scores might need to be adjusted to get a final solution to the manipulation instance. See [75] for further details). Therefore, each entry is an integer between 0 to $m-1$. Our mechanism differs from theirs in several aspects. First, our matrix has both m rows and m columns. Moreover, each row corresponds to a candidate and each column corresponds to a position. The entry $M[i,j]$ is defined as the number of manipulators that rank the i-th candidate in the $(m-j+1)$-th position. Therefore, each entry is an integer between 0 and t, where t is the number of manipulators. Second, their algorithms are heuristic algorithms, while ours are exact algorithms. Third, our method can be easily adapted to reduce Borda manipulation instances to ILP instances with m^2 variables (we discuss in detail later), however their method seems difficult to reduce Borda manipulation instances to ILP instances with the number of variables bounded by a function of m.

In the following, we present an algorithm for FMM. The algorithm is based on a dynamic programming method associated with a boolean dynamic table $DT(\ell, T)$, where $\ell \in [m]$ and $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$ is a multiset of non-negative integers. The entry $DT(\ell, T) = 1$ means that there is an $m \times m$ matrix M such that:

(1) $\sum_{j=1}^{m} M[i][j] = t$ for all $i \in [\ell]$;

(2) $\sum_{j=1}^{m}(j-1) \cdot M[i][j] \leq g_i$ for all $i \in [\ell]$; and

(3) $\sum_{i=1}^{l} M[i][j] = T_j$ for all $j \in [m]$.

It is clear that a given instance of FMM is a yes-instance if and only if $DT(m, T_{[m]}) = 1$, where $T_{[m]}$ is the multiset containing m copies of t. The algorithm for solving FMM is described in Algorithm 6.2.

Lemma 6.1. *FMM is solvable in* $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$ *time.*

Algorithm 6.2: A dynamic algorithm for FMM.

Input : An instance $(g = \{g_1, g_2, ..., g_m\}, t)$ of FMM.
Output : "Yes" if the given instance is a yes-instance, and "No" otherwise.

/* Initialization */
forall the *possible multisets* $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$ *with* $\sum_{j=1}^{m} T_j = t$ **do**
    if $\sum_{j=1}^{m} (j-1) \cdot T_j \leq g_1$ **then**
        $DT(1, T) = 1$
    else
        $DT(1, T) = 0$;
    end
end
/* Updating $DT(\ell, T)$ */
forall the $\ell = 2$ *to* m **do**
    forall the *possible multisets* $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$ **do**
        forall the *possible multisets* $T' = \{T'_j \in \mathbb{N} \mid j \in [m], T'_j \leq T_j\}$ *with* $\sum_{j=1}^{m} T'_j = t$ **do**
            Let $T - T' = \{T_1 - T'_1, T_2 - T'_2, ..., T_m - T'_m\}$;
            if $DT(\ell - 1, T - T') = 1$ *and* $\sum_{j=1}^{m} (j-1) \cdot T'_j \leq g_\ell$ **then**
                $DT(\ell, T) := 1$;
            else
                $DT(\ell, T) := 0$;
            end
        end
    end
end
if $DT(m, T_{[m]}) = 1$ **then**
    Return "Yes";
else
    Return "No";
end

Proof. In the initialization, we consider all possible multisets $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$ with $\sum_{j=1}^{m} T_j = t$. Since T has at most $\binom{t+m-1}{t}$ possibilities (according to $\sum_{j=1}^{m} T_j = t$), the running time of the initialization is bounded by $O^*(\binom{t+m-1}{t})$. In the updating procedure, we use a loop indicated by a variable ℓ with $2 \leq \ell \leq m$ to update $DT(\ell, T)$. In each loop we compute the values of the entries $DT(\ell, T)$ for all multisets $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$. To compute each of the entries, we check whether there is a multiset $T' = \{T'_1, T'_2, ..., T'_m\}$ with $\sum_{j=1}^{j=m} T'_j = t$ and $T'_j \leq T_j$ for every $j \in [m]$, such that $DT(\ell - 1, T - T') = 1$ and $\sum_{j=1}^{m} (j-1) \cdot T'_j \leq g_\ell$. Since there are at most $\binom{t+m-1}{t}$ possible multisets T', the time to compute each $DT(\ell, T)$ is bounded by $O^*(\binom{t+m-1}{t})$. Since T has at most $(t+1)^m$ possibilities, there are at most $(t+1)^m$ entries needed to be computed in each loop, implying a whole running time of $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$ for the updating procedure. In conclusion, the whole running time of the algorithm is $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$.

To check the correctness, recall that for each $\ell \in [m]$ and $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$, $DT(\ell, T) = 1$ means that there is an $m \times m$ matrix M such that:

(1) $\sum_{j=1}^{m} M[i][j] = t$ for all $i \in [\ell]$;

(2) $\sum_{j=1}^{m} (j-1) \cdot M[i][j] \leq g_i$ for all $i \in [\ell]$; and

(3) $\sum_{i=1}^{\ell} M[i][j] = T_j$ for all $j \in [m]$.

Thus, $DT(\ell, T) = 1$ if and only if there is at least one $T' = \{T'_j \in \mathbb{N} \mid j \in [m], T'_j \leq T_j\}$ with $\sum_{j=1}^{m} T'_j = t$ such that $DT(\ell-1, T-T') = 1$ and $\sum_{j=1}^{m} (j-1) \cdot T'_j \leq g_\ell$. This corresponds to the updating procedure. In the initialization, we compute the value of $DT(1, T)$ for all possible multisets $T = \{T_j \in \mathbb{N} \mid j \in [m], T_j \leq t\}$. We set $DT(1, T)$ to 1 whenever $\sum_{j=1}^{m} T_j = t$ and $\sum_{j=1}^{m} (j-1) \cdot T_j \leq g_1$. Thus, we can set $M[1][j] = T_j$ to make sure that the three required conditions in our definition for the dynamic table hold. Finally, it is obvious that the given instance of FMM is a yes-instance if and only if $DT(m, T_{[m]}) = 1$ where $T_{[m]}$ is the multiset containing m copies of t. The theorem follows. □

We now come to show how to solve UM-Borda via FMM. A *partial vote* is a partial injection $\pi : \mathcal{C} \cup \{p\} \to [|\mathcal{C} \cup \{p\}|]$ which maps a subset $C \subseteq \mathcal{C} \cup \{p\}$ to $[|\mathcal{C} \cup \{p\}|]$ such that for any two distinct $a_1, a_2 \in C$, $\pi(a_1) \neq \pi(a_2)$. Here, C is the *domain* and $\{\pi(a) \mid a \in C\}$ is the *codomain* of π. A position not in the codomain is called a *free position*. For simplicity, we define $\pi(c) = -1$ for $c \notin C$.

Lemma 6.2. *UM-Borda can be reduced to FMM in polynomial time.*

Proof. Let $\mathcal{F} = ((\mathcal{C} \cup \{p\}, \Pi_\mathcal{V}, \mathcal{V}), \mathcal{V}', t)$ be an instance of UM-Borda. By Observation 6.1, we know that if $\mathcal{F}$ is a yes-instance there must be a solution $\Pi_{\mathcal{V}'}$ such that each manipulator ranks p in the top. We assume that $BSC_\mathcal{V}(p) + t \cdot |\mathcal{C}| - BSC_\mathcal{V}(c) - 1 \geq 0$ for all $c \in \mathcal{C}$ as discussed in Section 6.2. Let $(c_1, c_2, ..., c_{|\mathcal{C}|})$ be any arbitrary order of $\mathcal{C}$. We construct an instance $\mathcal{F}' = (t, g)$ of FMM, where $g = \{g_1, g_2, ..., g_{|\mathcal{C}|}\}$ such that $g_i = BSC_\mathcal{V}(p) + t \cdot |\mathcal{C}| - BSC_\mathcal{V}(c_i) - 1$ for all $i \in [|\mathcal{C}|]$. It is clear that the construction takes polynomial time. In the following, we prove that $\mathcal{F}$ is a yes-instance if and only if $\mathcal{F}'$ is a yes-instance.

($\Rightarrow$:) Given a solution $\Pi_{\mathcal{V}'}$ of $\mathcal{F}$, we can get a solution for $\mathcal{F}'$ by setting $M[i][j] = |\{\pi \in_+ \Pi_{\mathcal{V}'} \mid \pi(c_i) = j\}|$, where $\{\pi \in_+ \Pi_{\mathcal{V}'} \mid \pi(c_i) = j\}$ is the multiset containing all votes $\pi \in_+ \Pi_{\mathcal{V}'}$ with $\pi(c_i) = j$. By the above construction, the correctness of M is easy to verify.

($\Leftarrow$:) Let a $|\mathcal{C}| \times |\mathcal{C}|$ matrix M be a solution of $\mathcal{F}' = (t, g = \{g_1, g_2, ..., g_{|\mathcal{C}|}\})$. Then, a solution for $\mathcal{F}$, where there are exactly $M[i][j]$ manipulators who rank c_i in the $(|\mathcal{C}| - j + 2)$-th positions (notice that we have in total $|\mathcal{C}| + 1$ candidates), can be constructed by the polynomial-time algorithm described in Algorithm 6.3. For simplicity, for a candidate c_i and an integer j with $1 \leq j \leq |\mathcal{C}|$, $c_i \rightsquigarrow j$ means that there are less than $M[i][j]$ manipulators who have already placed c_i in the $(|\mathcal{C}| - j + 2)$-th positions. For two partial votes π and π' and two candidates c and c', $(\pi, c) \leftrightarrow (\pi', c')$ means to switch the position of c in π and the position of c' in π', that is, if $\pi(c) = j, \pi'(c') = j'$, then, after $(\pi, c) \leftrightarrow (\pi', c')$, we get $\pi(c') = j, \pi'(c) = j'$.

Algorithm 6.3: Algorithm for reducing from UM-Borda to FMM.

Input : A solution M (an m by m matrix) of an instance $\mathcal{F}' = (g = \{g_1, g_2, ..., g_m\}, t)$ of FMM, where $\mathcal{F}'$ is constructed from a given instance $\mathcal{F} = ((\mathcal{C} \cup \{p\}, \Pi_{\mathcal{V}}, \mathcal{V}), \mathcal{V}', t)$ of UM-Borda as described in the beginning of the proof to Lemma 6.2.

Output: A solution $\Pi_{\mathcal{V}'}$ of $\mathcal{F}$.

1 Initialize $\Pi_{\mathcal{V}'} = \{\pi_1, \pi_2, ..., \pi_t\}$ of partial votes such that each partial vote has empty domain;
2 **forall the** $z \in [t]$ **do**
3     $\pi_z(p) := |\mathcal{C}| + 1$;
4 **end**
5 **forall the** $\bar{j} = |\mathcal{C}|$ *to* 1 **do**
6   **while** $\exists \pi_z$ *where the* $(|\mathcal{C}| - \bar{j} + 2)$*-th position is free* **do**
7   Let c_i be any candidate with $c_i \rightsquigarrow \bar{j}$;
8   **if** $\pi_z(c_i) = -1$ **then**
9             $\pi_z(c_i) := \bar{j}$;
10   **else**
11   Let $j' = \pi_z(c_i)$ and let $\pi_{z'}$ be a vote with $\pi_{z'}(c_i) = -1$;
12   **if** *the* $(|\mathcal{C}| - \bar{j} + 2)$*-th position of* $\pi_{z'}$ *is free* **then**
13                 $\pi_{z'}(c_i) := \bar{j}$
14   **else**
15   **while** $\exists j'' > \bar{j}$ *with* $\pi_z^{-1}(j'') = \pi_{z'}^{-1}(j')$ **do**
16                     $(\pi_z, \pi_z^{-1}(j')) \leftrightarrow (\pi_{z'}, \pi_{z'}^{-1}(j'))$;
17                     $j' := j''$;
18   **end**
19                 $(\pi_z, \pi_z^{-1}(j')) \leftrightarrow (\pi_{z'}, \pi_{z'}^{-1}(j'))$;
20                 $\pi_z(c_i) := \bar{j}$;
21   **end**
22   **end**
23   **end**
24 **end**
25 Return $\Pi_{\mathcal{V}'}$;

Since $\sum_{i=1}^{|\mathcal{C}|} M[i][\bar{j}] = t$ and there are exactly t manipulators, there must be a candidate c_i with $c_i \rightsquigarrow \bar{j}$ whenever there is a vote whose $(|\mathcal{C}| - \bar{j} + 2)$-th position is free, which guarantees the soundness of Line 7 in Algorithm 6.3. Similarly, there must be a $\pi_{z'}$ with $\pi_{z'}(c_i) = -1$ in Line 11, since, otherwise, $\sum_{j=1}^{|\mathcal{C}|} M[i][j] > t$, contradicting that the given instance of FMM is a yes-instance. After the switches in the **while loop** in Lines 15-18, and Line 19, both π_z and $\pi_{z'}$ must fulfill the following property: no candidate is placed in two different positions in either vote. See Figure 6.1 for an example of the **while loop** in Lines 15-18.

Obviously, such a constructed $\Pi_{\mathcal{V}'}$ is a solution for UM-Borda: for each candidate $c_i \in \mathcal{C}$, we have

$$BSC_{\mathcal{V}'}(c_i) = \sum_{j=1}^{|\mathcal{C}|} (j-1) \cdot M[i][j] \leq g_i = BSC_{\mathcal{V}}(p) + t \cdot |\mathcal{C}| - BSC_{\mathcal{V}}(c_i) - 1$$

To analyze the running time of the algorithm, we need the following lemma.

Lemma 6.3. *The* ***while loop*** *in Lines 15-18 in Algorithm 6.3 takes polynomial time.*

Proof. To prove the lemma, we construct an auxiliary bipartite graph B with $\mathcal{C}_{z'}$ as the left-hand vertices and $\mathcal{C}_z$ as the right-hand vertices, where $\mathcal{C}_{z'}$ and $\mathcal{C}_z$ are the sets of candidates which have been ranked in $\pi_{z'}$ and π_z in some k-th positions for $k \leq |\mathcal{C}| - \bar{j} + 2$, respectively. Two vertices are adjacent if and only if they represent the same candidate (as the vertices linked by a gray line in Figure 6.1) or they were placed in the same (but not identical) positions (as the vertices linked by a dark line in Figure 6.1). We observe that the constructed auxiliary graph has maximum degree two. Since $\mathcal{C}_{z'} \setminus \mathcal{C}_z$ is not empty, there is a simple path $P = (c_{a_1}, c_{a_2}, ..., c_{a_x})$ with $c_{a_1} = c_i$ and $c_{a_x} \in \mathcal{C}_{z'} \setminus \mathcal{C}_z$. It is clear that each execution of the **while loop** corresponds to the following switch procedure: switch the positions of a_k and a_{k+1} for a $k \in \{1, 3, ..., x-1\}$ (since $c_{a_1} = c_i \in \mathcal{C}_z$ and $c_{a_x} \in \mathcal{C}_{z'}$, we have that x is even). The lemma follows from the fact that the length of the simple path is bounded by $2|\mathcal{C}|$. □

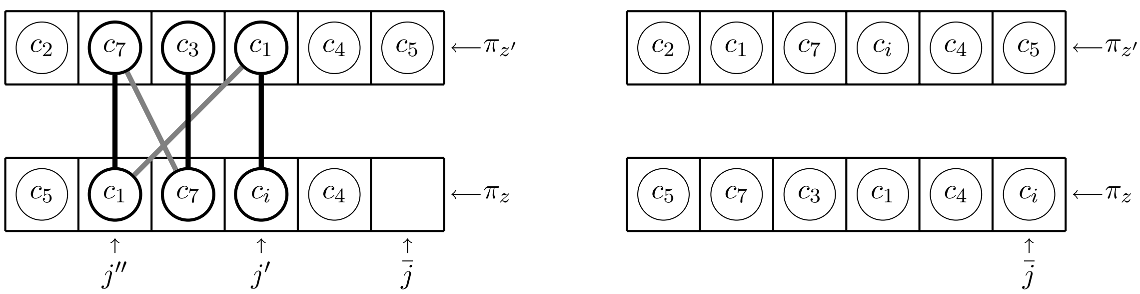


Figure 6.1: An illustration of the proof of Lemma 6.3. The left-hand shows the status of π_z and $\pi_{z'}$ before the switches. Due to the algorithm, the positions of every pair of candidates linked by the dark lines are switched. The gray lines here are to show that $\pi_z^{-1}(j'') = \pi_{z'}^{-1}(j')$, as in the precondition of the **while loop** in Lines 15-18 in Algorithm 6.3. The right-hand shows the status after these switches.

We now analyze the whole running time of Algorithm 6.3. The algorithm has four loops in total. The **for loop** in Lines 2-4 clearly takes polynomial time. The following **for loop** in Lines 5-24 loops exactly $|\mathcal{C}|$ times. Moreover, the **for loop** contains the **while loop** in Lines 6-23. The **while loop** in Lines 6-23 loops at most t times since each execute of a loop fixes a $(|\mathcal{C}| - \bar{j} + 2)$-th free position for some π_z where $\bar{j}$ is the loop indicator for the second **for loop**, and we have at most t different π_z. Therefore, to show that the algorithm takes polynomial time, it remains to show that each execute of the **while loop** in Lines 6-23 takes polynomial time. This is true since the most time-consuming step in this **while loop** is the **while loop** in Lines 15-18, which, due to Lemma 6.3, takes polynomial time. Summery all above, the running time of the algorithm is polynomially in t and $|\mathcal{C}|$. □

Due to Lemmas 6.1 and 6.2, we have the following theorem.

Theorem 6.2. *UM-Borda can be solved in $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$ time, where m is the number of candidates and t is the number of manipulators.*

Proof. Given an instance of UM-Borda, we reduce it to an instance of FMM, as described in Lemma 6.2. Then, we solve the instance of FMM with Algorithm 6.2. Finally, we construct a solution of the given instance of UM-Borda from the solution returned from Algorithm 6.2, as described in Lemma 6.2. According to Lemma 6.1, Algorithm 6.2 runs in $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$. Since it takes polynomial time to reduce from UM-Borda to FMM, the theorem follows. □

We remark that if the number of manipulators is bounded by a constant, the algorithm described in the proof of Theorem 6.2 runs in $O^*((t+1)^m)$ time. In particular, for $t = 2$, the algorithm runs in $O^*(3^m)$ time.

Next we show that FMM can be solved by an integer linear programming (ILP) based algorithm. The ILP contains m^2 variables x_{ij} for $i, j \in [m]$, and is subject to the following restrictions

$$
\begin{cases}
\sum_{i=1}^{m} x_{ij} = t \text{ for all } j \in [m] \\[2ex]
\sum_{j=1}^{m} x_{ij} = t \text{ for all } i \in [m] \\[2ex]
\sum_{j=1}^{m} (j-1) \cdot x_{ij} \leq g_i \text{ for all } i \in [m] \\[2ex]
x_{ij} \geq 0 \text{ for all } i, j \in [m]
\end{cases}
$$

where $t \in \mathbb{N}$, the number of the manipulators, and $g = \{g_1, g_2, \ldots, g_m\}$ with $g_i \in \mathbb{N}$ for all $i \in [m]$, the multiset of the capacities of the candidates, are input.

H. W. Lenstra [177] proposed an $O^*(\zeta^{O(\zeta)})$-time algorithm for solving ILP with ζ variables. The running time was then improved by R. Kannan [161], and Frank and Tardos [126] (see Lemma 1.1).

Due to Lemmas 6.2 and 1.1, we have the following theorem.

Theorem 6.3. *UM-Borda admits an algorithm with running time $O^*(2^{5(m^2+o(m^2))\log m})$, where m is the number of candidates.*

6.4 Conclusion

We have studied exact combinatorial algorithms for Borda manipulation problems. In particular, we proposed two exact combinatorial algorithms with running times $O^*((m\cdot 2^m)^{t+1})$ and $O^*(\binom{t+m-1}{t}\cdot(t+1)^m)$ for weighted Borda manipulation (WM-Borda) and unweighted Borda manipulation (UM-Borda), respectively, where t is the number of manipulators and m is the number of candidates in the given election. Observe that if t is bounded by a constant, UM-Borda can be solved in $O^*((t+1)^m)$ time. Our results answer an open question posed by Betzler, Niedermeier and Woeginger [26] affirmatively. In addition, we presented an integer linear programming based $\mathcal{FPT}$-algorithm with running time $O^*(2^{5(m^2+o(m^2))\log m})$ for UM-Borda. We remark that all our algorithms can be adapted to solve the weighted and unweighted manipulation problems for all (positional) scoring voting systems.

One future direction would be to improve the presented combinatorial algorithms. As showed in this chapter and in [26], UM-Borda is $\mathcal{FPT}$ with respect to the number m of candidates. In this paper, we proposed an algorithm for UM-Borda with running time $O^*(\binom{t+m-1}{t}\cdot(t+1)^m)$. In particular, if the number of the manipulators is a constant, the algorithm is single-exponential in the number of candidates. A challenging task is to investigate whether there is a single-exponential algorithm for UM-Borda when the number of manipulators is not a constant.

7

Conclusion and Outlook

This thesis investigated the (parameterized) complexity of strategic voting problems in restricted settings. In the following two sections, we first summarize our results and then discuss some directions for future research.

7.1 Summary of Results

This thesis mainly investigated the (parameterized) complexity of control, bribery and manipulation in elections under natural restrictions. In addition, this thesis explored the parameterized complexity of a number of possible winner(s) problems on partial tournaments with respect to several natural parameters.

In Chapters 2 and 3, we studied the (parameterized) complexity of control problems in generalized single-peaked elections. In particular, we studied control problems for r-Approval, Condorcet, Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ in k-peaked elections in Chapter 2. We proved that all the $\mathcal{NP}$-hardness of these control problems in the general case still hold even in $3,4$-peaked elections. However, in 2-peaked elections, several $\mathcal{NP}$-hardness in general turned out to be polynomial-time solvable. See Tables 2.1 and 2.2 for summaries of our results regarding these problems. In Chapter 3, we studied the (parameterized) complexity of control problems in elections with bounded single-peaked width for Condorcet, Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. We proved that for the constructive control by adding/deleting votes, all the $\mathcal{NP}$-hardness in general still hold even in elections with single-peaked width 3. However, for the destructive case, all the $\mathcal{NP}$-hardness results turned out to be $\mathcal{FPT}$ with respect to single-peaked width, implying the polynomial-time solvability of these problems in elections with constant single-peaked width. Furthermore, we derived a framework for identifying $\mathcal{FPT}$ control problems with respect to the single-peaked width. See Table 3.1 for a summary of our results regarding this topic. Many of our $\mathcal{NP}$-hardness reductions apply to other restricted elections, such as elections with bounded single-crossing width and d-Euclidean elections. See Section 3.6 for detailed discussions.

In Chapter 4, we studied the distance restricted bribery problem for Condorcet, Maximin and Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$. In the bribery problem, each voter may be bribed to recast his vote in any arbitrary way [108]. In the distance restricted bribery problem, each voter can recast a new vote which, however, has to be as close as to his original vote. We adopted the Hamming distance and the Kendall-Tau distance to measure the similarity of two votes. Our results show that the distance restricted bribery problem is generally $\mathcal{NP}$-hard even when the distance is bounded by a small constant. See Table 4.1 for a summary of our results regarding this topic.

In Chapter 5, we studied several possible winner(s) problems on partial tournaments. In this scenario the candidates are represented by vertices. Moreover, there is an arc from a vertex a to a vertex b if a beats b. Here, a "beats" b means that there are more voters who prefer a to b. The winners are selected according to some well-defined tournament solutions, e.g, Uncovered set, Banks set, etc. In particular, we focused on the possible winner problems with respect to Uncovered set and Banks set. For Uncovered set, the question is whether a given subset of vertices (candidates) can be included in the Uncovered set by adding/reversing some arcs to the given partial tournament. In particular, we studied three parameters: the size of the given subset of vertices, the number of arcs that are allowed to be reversed and the number of arcs that are allowed to add. For Banks set, the question is whether a given distinguished vertex (candidate) is a 0-indegree vertex in some maximal transitive subtournament. In particular, we studied three parameters: the size of the maximal transitive subtournament, the number of the in-neighbors of the distinguished candidate, and the number of the out-neighbors of the distinguished candidate. For problems considered in this chapter, we proposed $\mathcal{FPT}$ results, $\mathcal{W}$-hardness results and $\mathcal{XP}$ results. See Table 5.1 for a summary of our results regarding this topic.

In Chapter 6, we studied exact combinatorial algorithms for both weighted and unweighted Borda manipulation problems. In particular, we proposed two exact combinatorial algorithms with running times $O^*((m \cdot 2^m)^{t+1})$ and $O^*(\binom{t+m-1}{t} \cdot (t+1)^m)$ for weighted Borda manipulation (WM-Borda) and unweighted Borda manipulation (UM-Borda), respectively, where t is the number of manipulators and m is the number of candidates. Observe that if t is bounded by a constant, UM-Borda can be solved in $O^*((t+1)^m)$ time. Our results answer an open problem posed by Betzler, Niedermeier and Woeginger [26] affirmatively: UM-Borda with two manipulators can be solved in single-exponential time with respect to the number of candidates. Moreover, we proposed an integer linear program formulation for UM-Borda with m^2 variables. As a consequence of our formulation and the $\mathcal{FPT}$ algorithm for ILP devised by Frank and Tardos [126], UM-Borda can be solved in $O^*(2^{5(m^2+o(m^2))\log m})$ time. Our results are summarized in Table 6.1 and Theorem 6.3.

7.2 Further Research Directions

For open questions and remarks concerning the specific problems investigated in this thesis, we refer to the respective conclusion section of each chapter. More specifically, questions regarding control problems in k-peaked elections can be found in Section 2.4, regarding control problems in elections with bounded single-peaked width, bounded single-crossing width, Euclidean elections can be found in Section 3.6, regarding the distance restricted bribery problem can be found in Section 4.4, regarding the possible winners problems can be found in Section 5.4, and regarding manipulation problems can be found in Section 6.4. In the following, we discuss more prominent directions for future research.

7.2.1 Practical $\mathcal{FPT}$ Algorithms

Parameterized complexity of voting problems has been widely studied in COMSOC, and many voting problems have been proved to be $\mathcal{FPT}$ (see e.g., [23, 25, 84, 154, 246, 249]). Nevertheless, most of the $\mathcal{FPT}$-algorithms are based on ILP formulations, and thus are far from practical. In this thesis, we derived such $\mathcal{FPT}$-algorithms for destructive control problems in elections with bounded single-peaked width for Condorcet, Copeland$^\alpha$ for every $0 \leq \alpha \leq 1$ and Maximin, with respect to single-peaked width. It is intriguing to investigate practically efficient $\mathcal{FPT}$-algorithms for these problems. Furthermore, deriving explicit kernels for these problems is also another challenging task.

7.2.2 Experimental Studies

To date, most of the work in COMSOC focused on the worst-case analysis of voting problems. Recently, this purely worst-case analysis, which ignores real-world settings, was criticized by researchers. See [67, 114, 191, 214, 239] for detailed discussions.

In this direction, two things are expected to be done. The first thing is to examine the hard voting problems ($\mathcal{NP}$-hard or $\mathcal{W}$-hard) with algorithms that run on real-world data. Some representative work can be found in [73, 189, 219, 238]. Concerning hard problems studied in this thesis, it is interesting to study heuristic algorithms for these problems and examine the performance of these algorithms with real-world data, in order to investigate how hard it is to solve these problems in practice. We refer to [191] for information on a site that gathers real-world preference data that is open to researchers. Another method to examine whether a certain voting problem is hard to solve in practice is to encode the voting problem into constraint satisfactory problems (CSPs for short). There has been many advanced CSP solvers for researchers to use such as CPLEX. See [136] for further discussions on CSP solvers.

The second thing is to examine the feasibility of strategic behavior in elections that are subject to some prominent distributions, through the lens of probability theory. For example, Procaccia and Rosenschein [214] introduced the concept of junta distributions (generally speaking, these are distributions over the elections that satisfy several constraints) and proved that if a (heuristic) algorithm often solve the manipulation problem when the instances are distributed according to a junta distribution, it would also often solve the manipulation problem when the instances are distributed according to many other plausible distributions. Another recent related work can be found in [137]. It is interesting to investigate the feasibility of strategic behavior in elections with bounded single-peaked width or bounded single-crossing width where the subelections restricted to the intervals are subject to some distributions, such as junta distributions.

7.2.3 Approximation Algorithms

Approximation algorithms lie in the central of computer science. Designing approximation algorithms for voting problems has long been studied (see e.g., [23, 53, 73, 143, 176, 181, 240, 241]). However, approximation algorithms for voting problems in restricted elections have been less investigated so far. It is interesting to study approximation algorithms for problems studied in this thesis, such as control problems in 2,3-peaked elections or in elections with constant single-peaked width or constant single-crossing width.

7.2.4 Surveys to Read

Finally, we refer to several representative surveys on computational social choice for more open problems and research directions: [22, 49, 66, 182, 220].

Bibliography

[1] F. N. Abu-Khzam, R. L. Collins, M. R. Fellows, M. A. Langston, W. H. Suters, and C. T. Symons. Kernelization algorithms for the vertex cover problem: Theory and experiments. In *ALENEX/ANALC*, pages 62–69, 2004. Cited on page 18

[2] N. Alon, R. Yuster, and U. Zwick. Color-coding. *J. ACM.*, 42(4):844–856, 1995. Cited on page 20

[3] F. A. Aloul, A. Ramani, I. L. Markov, and K. A. Sakallah. Generic ILP versus specialized 0-1 ILP: An update. In *ICCAD*, pages 450–457, 2002. Cited on page 19

[4] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. Cited on pages 15 and 16

[5] K. J. Arrow. A difficulty in the concept of social welfare. *J. Polit. Econ.*, 58(4):328–346, 1950. Cited on page 24

[6] O. L. Astrachan. Bubble sort: An archaeological algorithmic analysis. In *SIGCSE*, pages 1–5, 2003. Cited on page 98

[7] S. Athey. Single crossing properties and the existence of pure strategy equilibria in games of incomplete information. *Econometrica.*, 69(4):861–889, 2001. Cited on page 89

[8] S. Athey. Monotone comparative statics under uncertainty. *Q. J. Econ.*, 117(1):187–223, 2 2002. Cited on page 89

[9] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. In *AAAI*, pages 784–790, 2015. Cited on page 5

[10] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *AAMAS*, pages 107–115, 2015. Cited on page 5

[11] H. Aziz, S. Gaspers, N. Mattei, S. Mackenzie, N. Narodytska, and T. Walsh. Manipulating the probabilistic serial rule. In *AAMAS*, pages 1451–1459, 2015. Cited on page 5

[12] H. Aziz, P. Harrenstein, M. Brill, J. Lang, F. A. Fischer, and H. G. Seedig. Possible and necessary winners of partial tournaments. In *AAMAS*, pages 585–592, 2012. Cited on pages 131 and 134

[13] H. Aziz, S. Mackenzie, L. Xia, and C. Ye. Ex post efficiency of random assignments. In *AAMAS*, pages 1639–1640, 2015. Cited on page 5

[14] H. Aziz and P. Stursberg. A generalization of probabilistic serial to randomized social choice. In *AAAI*, pages 559–565, 2014. Cited on page 5

[15] J. Bang-Jensen and G. Gutin. *Digraphs – Theory, Algorithms and Applications.* Springer-Verlag, London, 2008. 2nd Edition. Cited on page 130

[16] J. S. Banks. Sophisticated voting outcomes and agenda control. *Soc. Choice. Welfare.*, 1(4):295–306, 1985. Cited on page 130

[17] M. S. Bansal and D. Fernàndez-Baca. Computing distances between partial rankings. *Inf. Process. Lett.*, 109(4):238–241, 2009. Cited on page 96

[18] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. *Computational Aspects of Approval Voting*, chapter 10, pages 199–251. Handbook on Approval Voting. Springer Berlin Heidelberg, 2010. Cited on page 27

[19] R. Berghammer, A. Rusinowska, and M. de Swart. Computing tournament solutions using relation algebra and relview. *Eur. J. Oper. Res.*, 226(3):636–645, 2013. Cited on page 130

[20] R. Berghammer and H. Schnoor. Relation-algebraic and tool-supported control of Condorcet voting. *CoRR*, abs/1304.7244, 2013. Cited on pages 100, 108, 110, and 113

[21] R. Berghammer and H. Schnoor. Control of Condorcet voting: Complexity and a relation-algebraic approach. In *AAMAS*, pages 1365–1366, 2014. Cited on pages 100, 108, 110, and 113

[22] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting – a parameterized complexity perspective. In *The Multivariate Algorithmic Revolution and Beyond*, pages 318–363, 2012. Cited on pages 9, 10, 98, and 165

[23] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. Fixed-parameter algorithms for Kemeny rankings. *Theor. Comput. Sci.*, 410(45):4554–4570, 2009. Cited on pages 5, 96, 164, and 165

[24] N. Betzler, M. R. Fellows, J. Guo, R. Niedermeier, and F. A. Rosamond. How similarity helps to efficiently compute Kemeny rankings. In *AAMAS (1)*, pages 657–664, 2009. Cited on pages 96 and 146

[25] N. Betzler, J. Guo, and R. Niedermeier. Parameterized computational complexity of Dodgson and Young elections. *Inf. Comput.*, 208(2):165–177, 2010. Cited on pages 20 and 164

[26] N. Betzler, R. Niedermeier, and G. J. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *IJCAI*, pages 55–60, 2011. Cited on pages 4, 146, 147, 148, 149, 152, 159, and 163

[27] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009. Cited on page 29

[28] D. Black. On the rationable of group decition-making. *J. Polit. Econ.*, 56:23–34, 1948. Cited on pages 24 and 26

[29] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, pages 17–37, 2009. Cited on page 18

[30] H. L. Bodlaender. Lower bounds for kernelization. In *IPEC*, pages 1–14, 2014. Cited on page 19

[31] H. L. Bodlaender, M. Cygan, S. Kratsch, and J. Nederlof. Deterministic single exponential time algorithms for connectivity problems parameterized by treewidth. In *ICALP*, pages 196–207, 2013. Cited on page 146

[32] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014. Cited on page 19

[33] H. L. Bodlaender, S. Thomassé, and A. Yeo. Analysis of data reduction: Transformations give evidence for non-existence of polynomial kernels. Technical report, Institute of Information and Computing Sciences, Utrecht University, Netherlands, 2008. Cited on pages 18 and 19

[34] A. G. Bonifacio. Bribe-proof reallocation with single-peaked preferences. *Soc. Choice. Welfare.*, 44(3):617–638, 2015. Cited on page 7

[35] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: Algorithms, theory, and experiments. *ACM Trans. Internet Techn.*, 5(1):231–297, 2005. Cited on page 98

[36] S. J. Brams and P. C. Fishburn. Approval voting. *AM Polit. Sci. Rev.*, 72(3):831–847, 1978. Cited on page 27

[37] S. J. Brams and D. M. Kilgour. Satisfaction Approval voting. In Rudolf Fara, Dennis Leech, and Maurice Salles, editors, *Voting Power and Procedures*, Studies in Choice and Welfare, pages 323–346. Springer International Publishing, 2014. Cited on page 5

[38] S. J. Brams, D. M. Kilgour, and M. R. Sanver. A minimax procedure for electing committees. *Public. Choice.*, 132(3-4):401–420, 2007. Cited on page 96

[39] S. J. Brams and M. R. Sanver. Critical strategies under Approval voting: Who gets ruled in and ruled out. *Elect. Stud.*, 25(2):287–305, 2006. Cited on page 27

[40] F. J. Brandenburg, A. Gleißner, and A. Hofmeier. Comparing and aggregating partial orders with Kendall-Tau distances. *Discrete Math., Alg. and Appl.*, 5(2), 2013. Cited on page 96

[41] F. Brandt. Some remarks on Dodgson's voting rule. *Math. Log. Q.*, 55(4):460–463, 2009. Cited on page 5

[42] F. Brandt, M. Brill, F. A. Fischer, and P. Harrenstein. Minimal retentive sets in tournaments. *Soc. Choice. Welfare.*, 42(3):551–574, 2014. Cited on page 130

[43] F. Brandt, M. Brill, and P. Harrenstein. Extending tournament solutions. In *AAAI*, pages 580–586, 2014. Cited on page 130

[44] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *AAAI*, pages 715–722, 2010. Cited on pages 7, 24, 43, 61, 62, 64, 71, 78, 86, and 91

[45] F. Brandt, M. Brill, and H. G. Seedig. On the fixed-parameter tractability of composition-consistent tournament solutions. In *IJCAI*, pages 85–90, 2011. Cited on page 130

[46] F. Brandt, M. Chudnovsky, I. Kim, G. Liu, S. Norin, A. Scott, P. Seymour, and S. Thomassé. A counterexample to a conjecture of Schwartz. *Soc. Choice. Welfare.*, 40(3):739–743, 2013. Cited on page 130

[47] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2015. Cited on page 130

[48] F. Brandt, F. A. Fischer, and P. Harrenstein. The computational complexity of choice sets. *Math. Log. Q.*, 55(4):444–459, 2009. Cited on page 130

[49] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. Woeginger. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, 2014. Cited on page 165

[50] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. In *AAAI*, pages 1398–1404, 2014. Cited on pages 6 and 97

[51] R. Bredereck, J. Chen, and G. J. Woeginger. Are there any nicely structured preference profiles nearby? In *IJCAI*, pages 62–68, 2013. Cited on page 24

[52] R. Bredereck, J. Chen, and G. J. Woeginger. A characterization of the single-crossing domain. *Soc. Choice. Welfare.*, 41(4):989–997, 2013. Cited on page 89

[53] E. Brelsford. Approximation and elections. Master's thesis, Rochester Institute of Technology, Rochester, USA, 2007. Cited on page 165

[54] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Approximability of manipulating elections. In *AAAI*, pages 44–49, 2008. Cited on pages 5, 101, and 148

[55] L. Bulteau, F. Hüffner, C. Komusiewicz, and R. Niedermeier. Multivariate algorithmics for NP-hard string problems. *Bulletin of the EATCS*, 114, 2014. Cited on page 20

[56] J. F. Buss and J. Goldsmith. Nondeterminism within P. *SIAM J. Comput.*, 22(3):560–572, 1993. Cited on page 146

[57] J. Byrka and K. Sornat. PTAS for Minimax Approval voting. In *WINE*, pages 203–217, 2014. Cited on page 5

[58] I. Caragiannis, J. A. Covey, M. Feldman, C. M. Homan, C. Kaklamanis, N. Karanikolas, A. D. Procaccia, and J. S. Rosenschein. On the approximability of Dodgson and Young elections. *Artif. Intell.*, 187:31–51, 2012. Cited on page 5

[59] M. Cesati. Compendium of parameterized problems. http://cesati.sprg.uniroma2.it/research/compendium/. Cited on page 18

[60] S. Chatterji, A. Sen, and H. Zeng. A characterization of single-peaked preferences via random social choice functions. *Theoretical Economics*, 2015. to appear. Cited on page 7

[61] E. Chávez, K. Figueroa, and G. Navarro. Effective proximity retrieval by ordering permutations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(9):1647–1658, 2008. Cited on page 98

[62] J. Cheetham, F. Dehne, A. Rau-Chaplin, U. Stege, and P. J. Taillon. Solving large FPT problems on coarse-grained parallel machines. *J. Comput. Syst. Sci.*, 67(4):691–706, 2003. Cited on pages 18 and 146

[63] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Combinatorial voter control in elections. In *MFCS*, pages 153–164, 2014. Cited on page 5

[64] J. Chen, I. A. Kanj, and G. Xia. Improved upper bounds for vertex cover. *Theor. Comput. Sci.*, 411(40-42):3736–3756, 2010. Cited on page 146

[65] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. In *STOC*, pages 177–186, 2008. Cited on page 135

[66] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *SOFSEM (1)*, pages 51–69, 2007. Cited on page 165

[67] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *AAAI*, pages 627–634, 2006. Cited on page 164

[68] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *J. ACM.*, 54(3):1–33, 2007. Cited on pages 3, 14, 146, and 148

[69] R. D. Cooter. *The Strategic Constitution.* Princeton University Press, 2002. Cited on page 25

[70] D. Cornaz, L. Galand, and O. Spanjaard. Bounded single-peaked width and proportional representation. In *ECAI*, pages 270–275, 2012. Cited on pages 60, 61, and 62

[71] D. Cornaz, L. Galand, and O. Spanjaard. Kemeny elections with bounded single-peaked or single-crossing width. In *IJCAI*, pages 76–82, 2013. Cited on pages 24, 60, 62, 63, and 90

[72] A. J. Davenport and J. Kalagnanam. A computational study of the Kemeny rule for preference aggregation. In *AAAI*, pages 697–702, 2004. Cited on page 5

[73] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. An empirical study of Borda manipulation. In *COMSOC*, pages 91–102, 2010. http://arxiv.org/abs/1007.5104. Cited on pages 4, 5, 148, 164, and 165

[74] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for Borda manipulation. In *AAAI*, pages 657–662, 2011. Cited on page 5

[75] J. Davies, G. Katsirelos, N. Narodytska, T. Walsh, and L. Xia. Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artif. Intell.*, 217:20–42, 2014. Cited on pages 4, 146, 148, and 153

[76] R. de Haan and S. Szeider. Compendium of parameterized problems at higher levels of the polynomial hierarchy. *ECCC.*, 21:143, 2014. Cited on page 18

[77] C. Dellarocas. Strategic manipulation of internet opinion forums: Implications for consumers and firms. *Manage. Sci.*, 52(10):1577–1593, 10 2006. Cited on page 89

[78] G. Demange. Single-peaked orders on a tree. *Math. Soc. Sci.*, 3(3):389–396, 1983. Cited on page 24

[79] P. Dey, N. Misra, and Y. Narahari. Kernelization complexity of possible winner and coalitional manipulation problems in voting. In *AAMAS*, pages 87–96, 2015. Cited on page 5

[80] C. L. Dodgson. *A Method for Taking Votes on More than Two Issues.* Clarendon Press, 1876. Cited on page 108

[81] J. P. Doignon and J. C. Falmagne. A polynomial time algorithm for unidimensional unfolding representations. *J. Algorithms.*, 16(2):218–233, March 1994. Cited on page 58

[82] M. Dom, J. Guo, F. Hüffner, R. Niedermeier, and A. Truß. Fixed-parameter tractability results for feedback set problems in tournaments. *J. Discrete Algorithms.*, 8(1):76–86, 2010. Cited on page 135

[83] M. Dom, D. Lokshtanov, and S. Saurabh. Incompressibility through colors and IDs. In *ICALP (1)*, pages 378–389, 2009. Cited on pages 19 and 143

[84] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica.*, 64(1):126–151, 2012. Cited on pages 5, 97, and 164

[85] F. Dorn, F. V. Fomin, D. Lokshtanov, V. Raman, and S. Saurabh. Beyond bidimensionality: Parameterized subexponential algorithms on directed graphs. *Inf. Comput.*, 233:60–70, 2013. Cited on page 146

[86] R. G. Downey and M. R. Fellows. Parameterized computational feasibility. In *Feasible Mathematics II*, pages 219–244, 1995. Cited on page 139

[87] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999. Cited on pages 17 and 41

[88] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013. Cited on pages 17 and 19

[89] R. G. Downey, M. R. Fellows, and U. Stege. Parameterized complexity: A framework for systematically confronting computational intractability. In *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future*, pages 49–99. Springer, 1999. Cited on page 17

[90] S. Dreyfus. Richard Bellman on the birth of dynamic programming. *Oper. Res.*, 50(1), 2002. Cited on page 20

[91] J. Duggan. Uncovered sets. *Soc. Choice. Welfare.*, 41(3):489–535, 2013. Cited on page 130

[92] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *WWW*, pages 613–622, 2001. Cited on pages 24 and 127

[93] P. J. Egan. Do something politics and double-peaked policy preferences. *J. Polit.*, 76(2):333–349, 2014. Cited on pages 24 and 25

[94] E. Elkind, P. Faliszewski, M. Lackner, and S. Obraztsova. The complexity of recognizing incomplete single-crossing preferences. In *AAAI*, pages 865–871, 2015. Cited on page 89

[95] E. Elkind, P. Faliszewski, and P. Skowron. A characterization of the single-peaked single-crossing domain. In *AAAI*, pages 654–660, 2014. Cited on page 91

[96] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In *AAMAS*, pages 53–60, 2014. Cited on pages 5 and 6

[97] E. Elkind, P. Faliszewski, and A. Slinko. Distance rationalization of voting rules. *Soc. Choice. Welfare.*, 45(2):345–377, 2015. Cited on pages 96 and 127

[98] E. Elkind, P. Faliszewski, and A. M. Slinko. Swap bribery. In *SAGT*, pages 299–310, 2009. Cited on pages 3, 6, 97, and 148

[99] E. Elkind, P. Faliszewski, and A. M. Slinko. Cloning in elections: Finding the possible winners. *J. Artif. Intell. Res. (JAIR)*, 42:529–573, 2011. Cited on pages 5 and 148

[100] G. Erdélyi, E. Hemaspaandra, and L. A. Hemaspaandra. More natural models of electoral control by partition. In *Arxiv*, 2014. http://arxiv.org/pdf/1410.2652.pdf. Cited on page 5

[101] G. Erdélyi, M. Lackner, and A. Pfandler. Computational aspects of nearly single-peaked electorates. In *AAAI*, pages 283–289, 2013. Cited on page 24

[102] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based Approval voting fully resists constructive control and broadly resists destructive control. *Math. Log. Q.*, 55(4):425–443, 2009. Cited on page 27

[103] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *ECAI*, pages 366–370, 2008. Cited on pages 7, 24, and 58

[104] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and aggregating rankings with ties. In *PODS*, pages 47–58, 2004. Cited on page 98

[105] R. Fagin, R. Kumar, and D. Sivakumar. Comparing top k lists. In *SODA*, pages 28–36, 2003. Cited on page 98

[106] P. Faliszewski. Nonuniform bribery. In *AAMAS (3)*, pages 1569–1572, 2008. Cited on pages 6 and 97

[107] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of bribery in elections. In *AAAI*, pages 641–646, 2006. Cited on pages 6, 13, 96, and 101

[108] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *J. Artif. Intell. Res. (JAIR)*, 35:485–532, 2009. Cited on pages 5, 6, and 162

[109] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM.*, 53(11):74–82, 2010. Cited on page 5

[110] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Multimode control attacks on elections. *J. Artif. Intell. Res. (JAIR)*, 40:305–351, 2011. Cited on pages 5, 6, 42, 50, 60, 61, 78, 101, 108, and 118

[111] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. *Artif. Intell.*, 207:69–99, 2014. Cited on pages 5, 6, 7, 24, 27, 28, 40, and 91

[112] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res. (JAIR)*, 35:275–341, 2009. Cited on pages 5, 6, 10, 19, 42, 46, 54, 60, 61, 67, 75, 88, 97, 101, 108, 110, and 113

[113] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Inf. Comput.*, 209(2):89–107, 2011. Cited on pages 6, 7, 24, 27, 28, 30, 33, 36, 86, and 91

[114] P. Faliszewski and A. D. Procaccia. AI's war on manipulation: Are we winning? *AI. MAG.*, 31(4):53–64, 2010. Cited on page 164

[115] S. L. Feld, B. Grofman, R. Hartly, M. Kilgour, and N. Miller. The uncovered set in spatial voting games. *Theor. Decis.*, 23(2):129–155, 1987. Cited on page 131

[116] M. R. Fellows. Parameterized complexity: The main ideas and connections to practical computing. *Electr. Notes Theor. Comput. Sci.*, 61:1–19, 2002. Cited on page 146

[117] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009. Cited on pages 44, 46, 48, 51, 54, 141, and 142

[118] M. R. Fellows, B. M. P. Jansen, D. Lokshtanov, F. A. Rosamond, and S. Saurabh. Determining the winner of a Dodgson election is hard. In *FSTTCS*, pages 459–468, 2010. Cited on pages 5 and 6

[119] M. R. Fellows, D. Lokshtanov, N. Misra, F. A. Rosamond, and S. Saurabh. Graph layout problems parameterized by vertex cover. In *ISSAC*, pages 294–305, 2008. Cited on page 19

[120] Q. Feng, Q. Zhou, W. Li, and J. Wang. Kernelization in parameterized computation: A survey. *Tsinghua Science and Technology*, 19(4):338–345, 2014. Cited on page 18

[121] H. Fernau, F. V. Fomin, D. Lokshtanov, M. Mnich, G. Philip, and S. Saurabh. Social choice meets graph drawing: How to get subexponential time algorithms for ranking and drawing problems. *Tsinghua Science and Technology*, 19(4):374–386, 8 2014. Cited on page 5

[122] J. Fiala, P. A. Golovach, and J. Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theor. Comput. Sci.*, 412(23):2513–2523, 2011. Cited on page 19

[123] A. Filos-Ratsikas, M. Li, J. Zhang, and Q. Zhang. Facility location with double-peaked preferences. In *AAAI*, pages 893–899, 2015. Cited on page 25

[124] P. C. Fishburn. Condorcet social choice functions. *SIAM J. Appl Math.*, 33(3):469–489, 1977. Cited on pages 5 and 131

[125] J. Flum and M. Grohe. *Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006. Cited on pages 17 and 18

[126] A. Frank and É. Tardos. An application of simultaneous diophantine approximaiton in combinatorial optimazation. *Combinatorica.*, 7(1):49–65, 1987. Cited on pages 19, 159, and 163

[127] R. Freeman, M. Brill, and V. Conitzer. On the axiomatic characterization of runoff voting rules. In *AAAI*, pages 675–681, 2014. Cited on page 5

[128] T. Gagie and S. Puglisi. Searching and indexing genomic databases via kernelization. *Frontiers in Bioengineering and Biotechnology*, 3(12):1–4, 2015. Cited on page 18

[129] S. Gailmard, J. W. Patty, and E. M. Penn. Arrow's theorem on single-peaked domains. In E. Aragonès, C. Beviá, H. Llavador, and N. Schofield, editors, *The Political Economy of Democoracy*, pages 235–342. Fundación BBVA, 2009. Cited on pages 7 and 24

[130] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is NP-complete. *SIAM J. Appl Math.*, 32(4):826–834, 1977. Cited on pages 36 and 39

[131] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, New York, 1979. Cited on pages 15, 19, 63, and 124

[132] S. Gaspers and M. Mnich. Feedback vertex sets in tournaments. *J. Graph Theor.*, 72(1):72–89, 2013. Cited on page 130

[133] J. Gathen and M. Sieveking. A bound on solutions of linear integer equahties and inequahties. *Pro Amer. Math. Soc.*, 72(1):155–158, 1978. Cited on page 19

[134] J. Geanakoplos. Three brief proofs of Arrow's impossibility theorem. *Econ. Theory.*, 26(1):211–215, 2005. Cited on page 3

[135] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica.*, 41(4):587–601, 1973. Cited on page 3

[136] E. Goldberg and Y. Novikov. Berkmin: A fast and robust sat-solver. *Discrete. Appl. Math.*, 155(12):1549 – 1561, 2007. Cited on page 164

[137] J. Goldsmith, J. Lang, N. Mattei, and P. Perny. Voting with rank dependent scoring rules. In *AAAI*, pages 698–704, 2014. Cited on page 165

[138] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theor. Comput. Sci.*, 38:293–306, 1985. Cited on pages 99, 108, and 121

[139] M. T. Goodrich, R. Tamassia, and M. H. Goldwasser. *Data Structures and Algorithms in Java.* Wiley, 6 edition, 2014. Cited on page 119

[140] J. M. Grandmont. Intermediate preferences and the Majority rule. *Econometrica.*, 46(2):317–30, March 1978. Cited on page 91

[141] B. Gulko and S. Leung. Maximin safety: When failing to lose is preferable to trying to win. *Arxiv*, 2015. Cited on page 5

[142] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007. Cited on page 18

[143] F. Gurski and M. Roos. Binary linear programming solutions and non-approximability for control problems in voting systems. *Discrete. Appl. Math.*, 162:391–398, 2014. Cited on pages 5 and 165

[144] R. W. Hamming. Error detecting and error correcting codes. *AT & T. Tech. J.*, 26(2):147–160, 1950. Cited on page 98

[145] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *J. ACM.*, 44(6):806–825, 1997. Cited on pages 6 and 108

[146] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.*, 171(5-6):255–285, 2007. Cited on pages 3, 6, 11, and 27

[147] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theor. Comput. Sci.*, 349(3):382–391, 2005. Cited on page 6

[148] L. A. Hemaspaandra, R. Lavaee, and C. Menton. Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. In *AAMAS*, pages 1345–1346, 2013. Cited on page 88

[149] D. Hermelin, S. Kratsch, K. Soltys, M. Wahlström, and X. Wu. A completeness theory for polynomial (Turing) kernelization. In *IPEC*, pages 202–215, 2013. Cited on page 19

[150] D. Hermelin and X. Wu. Weak compositions and their applications to polynomial lower bounds for kernelization. In *SODA*, pages 104–113, 2012. Cited on page 19

[151] L. Hoots and R. C. Powers. Anonymous and positively responsive aggregation rules. *Math. Soc. Sci.*, 77:9–14, 2015. Cited on page 5

[152] O. Hudry. A note on "Banks winners in tournaments are difficult to recognize" by G. J. Woeginger. *Soc. Choice. Welfare.*, 23(1):113–114, 2004. Cited on page 135

[153] O. Hudry. A survey on the complexity of tournament solutions. *Math. Soc. Sci.*, 57(3):292–303, 2009. Cited on pages 130, 131, 134, and 135

[154] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. Voting schemes for which it can be difficult to tell who won the election. *Soc. Choice. Welfare.*, 6(2):157–165, 1989. Cited on pages 3, 6, and 164

[155] J. J. Bartholdi III and J. B. Orlin. Single transferable vote resists strategic voting. *Soc. Choice. Welfare.*, 8(4):341–354, 1991. Cited on pages 3 and 6

[156] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Soc. Choice. Welfare.*, 6(3):227–241, 1989. Cited on pages 3, 6, and 148

[157] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. How hard is it to control an election? *Math. Comput. Model.*, 16(8-9):27–40, 1992. Cited on pages 3, 6, 11, 12, 27, 28, 42, 43, 50, 58, 60, 61, and 64

[158] J. J. Bartholdi III and M. A. Trick. Stable matching with preferences derived from a psychological model. *Oper. Res. Lett.*, 5(4):165–169, 1986. Cited on page 58

[159] R. Jacob, T. Lieber, and M. Mnich. Treewidth computation and kernelization in the parallel external memory model. In *TCS*, pages 78–89, 2014. Cited on page 18

[160] M. Kalech, S. Kraus, G. A. Kaminka, and C. V. Goldman. Practical voting rules with partial information. *Auton. Agent. Multi-AG.*, 22:151–182, 2011. Cited on page 24

[161] R. Kannan. Minkowski's convex body theory and integer programming. *Math. Oper. Res.*, 12:415–440, 1987. Cited on pages 19 and 159

[162] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972. Cited on page 19

[163] M. Karpinski and W. Schudy. Faster algorithms for feedback arc set tournament, Kemeny rank aggregation and betweenness tournament. In *ISAAC (1)*, pages 3–14, 2010. Cited on page 5

[164] J. M. Keil. On the complexity of scheduling tasks with discrete starting times. *Oper. Res. Lett.*, 12(5):293–295, 1992. Cited on page 33

[165] M. G. Kendall. A new measure of rank correlation. *Biometrika.*, 30(1/2):pp. 81–93, 1938. Cited on pages 98 and 127

[166] D. M. Kilgour and E. Marshall. Approval balloting for fixed-size committees. In D. S. Felsenthal and M. Machover, editors, *Electoral Systems*, Studies in Choice and Welfare, pages 305–326. Springer Berlin Heidelberg, 2012. Cited on page 5

[167] E. J. Kim, A. Langer, C. Paul, F. Reidl, P. Rossmanith, I. Sau, and S. Sikdar. Linear kernels and single-exponential algorithms via protrusion decompositions. In *ICALP*, pages 613–624, 2013. Cited on page 146

[168] E. J. Kim, C. Paul, and G. Philip. A single-exponential FPT algorithm for the k_4-minor cover problem. *J. Comput. Syst. Sci.*, 81(1):186–207, 2015. Cited on page 146

[169] C. L. Kingsford, B. Chazelle, and M. Singh. Solving and analyzing side-chain positioning problems using linear and integer programming. *Bioinformatics.*, 21(7):1028–1039, 2005. Cited on page 19

[170] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *M-PREF*, 2005. Cited on pages 6 and 130

[171] S. Konieczny, J. Lang, and P. Marquis. DA^2 merging operators. *Artif. Intell.*, 157(1-2):49–79, 2004. Cited on page 96

[172] S. Kratsch. Recent developments in kernelization: A survey. *Bulletin of the EATCS*, 113:58–97, 2014. Cited on page 19

[173] G. Laffond, J. Lain, and J. F. Laslier. Composition-consistent tournament solutions and social choice functions. *Soc. Choice. Welfare.*, 13(1):75–93, 1996. Cited on page 62

[174] H. Landau. On dominance relations and the structure of animal societies III. the condition for score structure. *B. Math. Biophys.*, 15(2):143–148, 1953. Cited on page 131

[175] J. Lang, G. Pigozzi, M. Slavkovik, and L. van der Torre. Judgment aggregation rules based on minimization. In *TARK*, pages 238–246, 2011. Cited on page 96

[176] R. LeGrand, E. Markakis, and A. Mehta. Some results on approximating the minimax solution in Approval voting. In *AAMAS*, pages 1193–1195, 2007. Cited on pages 5 and 165

[177] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res*, 8(4):538–548, 1983. Cited on pages 19, 65, and 159

[178] D. Lepelley. Condorcet efficiency of positional voting rules with single-peaked preferences. *Economic Design*, 1(1):289–299, 1994. Cited on page 7

[179] J. Levin and B. Nalebuff. An introduction to vote-counting schemes. *J. Econ. Perspect.*, 9(1):3–26, 1995. Cited on page 10

[180] A. P. Lin. The complexity of manipulating k-Approval elections. In *ICAART (2)*, pages 212–218, 2011. http://arxiv.org/abs/1005.4159. Cited on pages 27, 28, 102, and 103

[181] A. P. Lin. *Solving Hard Problems in Election Systems*. PhD thesis, Rochester Institute of Technology, 2012. Cited on pages 5 and 165

[182] C. Lindner and J. Rothe. Fixed-parameter tractability and parameterized complexity, applied to problems from computational social choice. In *Mathematical Programming Glossary*, 2008. Cited on page 165

[183] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theor. Comput. Sci.*, 410(27-29):2746–2753, 2009. Cited on pages 29 and 42

[184] H. Liu and D. Zhu. Parameterized complexity of control problems in Maximin election. *Inf. Process. Lett.*, 110(10):383–388, 2010. Cited on pages 5, 42, and 78

[185] H. Liu and D. Zhu. Parameterized complexity of control by voter selection in Maximin, Copeland, Borda, Bucklin, and Approval election systems. *Theor. Comput. Sci.*, 498:115–123, 2013. Cited on page 5

[186] D. Lokshtanov. *New Methods in Parameterized Algorithms and Complexity*. PhD thesis, University of Bergen, 2009. Cited on page 19

[187] A. Lumini and L. Nanni. Detector of image orientation based on Borda count. *Pattern. Recongn. Lett.*, 27(3):180–186, 2006. Cited on page 24

[188] K. Magiera and P. Faliszewski. How hard is control in single-crossing elections? In *ECAI*, pages 579–584, 2014. Cited on pages 6 and 90

[189] A. Mao, A. D. Procaccia, and Y. Chen. Better human computation through principled voting. In *AAAI*, 2013. Cited on page 164

[190] N. Mattei, M. Silvia Pini, F. Rossi, and K. B. Venable. Bribery in voting with cp-nets. *Ann. Math. Artif. Intell.*, 68(1-3):135–160, 2013. Cited on pages 6 and 97

[191] N. Mattei and T. Walsh. Preflib: A library for preferences http://www.preflib.org. In *ADT*, pages 259–270, 2013. Cited on page 164

[192] K. Mehlhorn and P. Sanders. *Algorithms and Data Structures: The Basic Toolbox*. Springer, 2008. Cited on page 119

[193] P. Merz and B. Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE. T. Evolut. Comput.*, 4(4):337–352, 2000. Cited on page 98

[194] D. Micciancio and P. Voulgaris. A deterministic single exponential time algorithm for most lattice problems based on voronoi cell computations. *SIAM J. Comput.*, 42(3):1364–1391, 2013. Cited on page 146

[195] M. K. Miller and D. N. Osherson. Methods for distance-based judgment aggregation. *Soc. Choice. Welfare.*, pages 575–601, 2009. Cited on page 96

[196] N. R. Miller. A new solution set for tournaments and majority voting: Further graph-theoretical approaches to the theory of voting. *AM. J. Polit. Sci.*, 24(1):68–96, 1980. Cited on page 131

[197] N. R. Miller, B. Grofman, and S. L. Feld. The structure of the Banks set. *Public. Choice.*, 66:243–251, 1990. Cited on page 130

[198] J. A. Mirrlees. An exploration in the theory of optimum income taxation. *Rev. Econ. Stud.*, 38(114):175–208, 1971. Cited on page 89

[199] N. Misra, A. Nabeel, and H. Singh. On the parameterized complexity of Minimax Approval voting. In *AAMAS*, pages 97–105, 2015. Cited on pages 5 and 27

[200] M. Mnich, Y. R. Shrestha, and Y. Yang. When does Schwartz Conjecture hold? In *IJCAI*, pages 603–609, 2015. Cited on page 130

[201] H. Moulin. On strategy-proofness and single peakedness. *Public. Choice.*, 35(4):437–455, 1980. Cited on page 86

[202] E. Muller and M. Satterthwaite. The equivalence of strong positive association and strategy-proofness. *J. Econ. Theory.*, 14:412–418, 1977. Cited on page 3

[203] R. Niedermeier. *Invitation to Fixed-parameter Algorithms*. Oxford University Press Inc, 2006. Cited on pages 17, 18, 19, and 137

[204] S. Nitzan and A. Rubinstein. A further characterization of Borda ranking method. *Public. Choice.*, 36(1):183–158, 1981. Cited on page 5

[205] S. Obraztsova and E. Elkind. Optimal manipulation of voting rules. In *AAMAS*, pages 619–626, 2012. Cited on page 97

[206] Y. Ohta and T. Kanade. Stereo by intra- and inter-scanline search using dynamic programming. *IEEE T. Pattern. Anal.*, 7(2):139–154, 1985. Cited on page 20

[207] T. Perek, P. Faliszewski, M. Silvia Pini, and F. Rossi. The complexity of losing voters. In *AAMAS*, pages 407–414, 2013. Cited on page 5

[208] M. S. Pini, F. Rossi, and K. B. Venable. Bribery in voting with soft constraints. In *AAAI*, 2013. Cited on page 97

[209] J. Pitt, L. Kamara, M. J. Sergot, and A. Artikis. Voting in multi-agent systems. *Comput. J.*, 49(2):156–170, 2006. Cited on page 3

[210] S. Polyakovskiy, R. Berghammer, and F. Neumann. Solving hard control problems in voting systems via integer programming. *CoRR*, abs/1408.5987, 2014. Cited on page 19

[211] G. Popescu. Adaptive voting algorithms for group and social recommender systems. In *HCI*, pages 378–382, 2013. Cited on page 3

[212] G. Popescu. Designing a voting mechanism in the GroupFun music recommender system. In *HCI*, pages 383–386, 2013. Cited on page 3

[213] G. Popescu. Group recommender systems as a voting problem. In *HCI (26)*, pages 412–421, 2013. Cited on page 3

[214] A. D. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *J. Artif. Intell. Res. (JAIR)*, 28:157–181, 2007. Cited on page 164

[215] V. Raman and S. Saurabh. Parameterized algorithms for feedback set problems and their duals in tournaments. *Theor. Comput. Sci.*, 351(3):446–458, 2006. Cited on page 135

[216] V. Raman and S. Sikdar. Parameterized complexity of the induced subgraph problem in directed graphs. *Inf. Process. Lett.*, 104(3):79–85, 2007. Cited on page 135

[217] R. L. Rivest and E. Shen. An optimal single-winner preferential voting system based on game theory. In *COMSOC*, pages 399–410, 2010. Cited on page 5

[218] K. W. S. Roberts. Voting over income tax schedules. *J. Public. Econ.*, 8(3):329–340, 1977. Cited on page 89

[219] J. Rothe and L. Schend. Control complexity in Bucklin, Fallback, and Plurality voting: An experimental approach. In *SEA*, pages 356–368, 2012. http://arxiv.org/pdf/1203.3967.pdf. Cited on page 164

[220] J. Rothe and L. Schend. Challenges to complexity shields that are supposed to protect elections against manipulation and control: A survey. *Ann. Math. Artif. Intell.*, 68(1-3):161–193, 2013. Cited on page 165

[221] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory. Comput. Syst.*, 36(4):375–386, 2003. Cited on pages 6 and 67

[222] A. Rubinstein. Ranking the participants in a tournament. *J. Soc. Ind. Appl. Math*, 38(1):108–111, 1980. Cited on page 5

[223] N. F. Russell. Complexity of control of Borda count elections. Master's thesis, Rochester Institute of Technology, 2007. Cited on page 148

[224] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoustics, Speech and Signal Processing.*, 26(1):43–49, Feb 1978. Cited on page 20

[225] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *J. Econ. Theory.*, 10(2):187–216, 1975. Cited on page 3

[226] T. Schiavinotto and T. Stützle. A review of metrics on permutations for search landscape analysis. *Computers & OR*, 34(10):3143–3153, 2007. Cited on page 98

[227] M. Schulze. A new monotonic, clone-independent, reversal symmetric, and Condorcet-consistent single-winner election method. *Soc. Choice. Welfare.*, 36(2):267–303, 2011. http://mschulze.9mail.de/schulze1.pdf. Cited on pages 5 and 88

[228] A. Scott and M. Fey. The minimal covering set in large tournaments. *Soc. Choice. Welfare.*, 38(1):1–9, 2012. Cited on page 130

[229] Y. Shoham and K. Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2009. Cited on page 3

[230] S. S. Skiena. Dynamic programming. In *The Algorithm Design Manual*, pages 273–315. Springer London, 2008. Cited on page 20

[231] W. D. Smith. Descriptions of single-winner voting systems. http://m-schulze.9mail.de/votedesc.pdf, 2006. Cited on pages 5 and 10

[232] Y. Sprumon. The division problem with single-peaked preferences: A characterization of the uniform allocation rule. *Econometrica*, 59(2):509–519, 1991. Cited on page 7

[233] A. D. Taylor and A. M. Pacelli. *Mathematics and Politics.* Springer New York, 2008. Cited on page 86

[234] P. A. Taylor. Election systems 101. http://home.earthlink.net/ peter.a.taylor/swuusi.pdf, 2004. Cited on page 10

[235] T. N. Tideman. Independence of clones as a criterion for voting rules. *Soc. Choice. Welfare.*, 4(3):185–206, 1987. Cited on page 62

[236] C. A. Tovey. Tutorial on computational complexity. *Interfaces.*, 32(3):30–61, 2002. Cited on page 15

[237] T. Walsh. Uncertainty in preference elicitation and aggregation. In *AAAI*, pages 3–8, 2007. Cited on page 24

[238] T. Walsh. An empirical study of the manipulability of single transferable voting. In *ECAI*, pages 257–262, 2010. Cited on page 164

[239] T. Walsh. Is computational complexity a barrier to manipulation? *Ann. Math. Artif. Intell.*, 62(1-2):7–26, 2011. Cited on page 164

[240] T. Walsh. Where are the hard manipulation problems? *J. Artif. Intell. Res. (JAIR)*, 42:1–29, 2011. Cited on pages 5 and 165

[241] J. Wang, M. Yang, J. Guo, Q. Feng, and J. Chen. Parameterized complexity of control and bribery for d-Approval elections. In *COCOA*, pages 260–271, 2013. Cited on pages 5 and 165

[242] G. J. Woeginger. Banks winners in tournaments are difficult to recognize. *Soc. Choice. Welfare.*, 20(3):523–528, 2003. Cited on pages 131 and 135

[243] G. J. Woeginger. Exact algorithms for NP-hard problems: A survey. In G. Rinaldi M. Jünger, G. Reinelt, editor, *Combinatorial Optimization–Eureka, You Shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–207. Springer Berlin Heidelberg, 2003. Cited on page 146

[244] T. H. Wu, A. Davoodi, and J. T. Linderoth. GRIP: Global routing via integer programming. *IEEE. T. Comput. AID. D.*, 30(1):72–84, 2011. Cited on page 19

[245] L. Xia. Designing social choice mechanisms using machine learning. In *AAMAS*, pages 471–474, 2013. Cited on page 3

[246] L. Xia. Fixed-parameter tractability of integer generalized scoring rules. In *AAMAS*, pages 1599–1600, 2014. Cited on pages 88 and 164

[247] L. Xia and V. Conitzer. Finite local consistency characterizes generalized scoring rules. In Craig Boutilier, editor, *IJCAI*, pages 336–341, 2009. Cited on page 5

[248] L. Xia, V. Conitzer, and A. D. Procaccia. A scheduling approach to coalitional manipulation. In *EC*, pages 275–284, 2010. Cited on page 5

[249] Y. Yang. Election attacks with few candidates. In *ECAI*, pages 1131–1132, 2014. Cited on pages 88 and 164

[250] Y. Yang and J. Guo. Exact algorithms for weighted and unweighted Borda manipulation problems. In *AAMAS*, pages 1327–1328, 2013. http://arxiv.org/abs/1304.3145. Cited on pages 21 and 22

[251] Y. Yang and J. Guo. Possible winner problems on partial tournaments: A parameterized study. In *ADT*, pages 425–439, 2013. Cited on pages 21 and 22

[252] Y. Yang and J. Guo. The control complexity of r-Approval: from the single-peaked case to the general case. In *AAMAS*, pages 621–628, 2014. Cited on pages 5, 20, 21, and 37

[253] Y. Yang and J. Guo. Controlling elections with bounded single-peaked width. In *AAMAS*, pages 629–636, 2014. Cited on pages 5, 20, and 21

[254] Y. Yang and J. Guo. How hard is control in multi-peaked elections: A parameterized study. In *AAMAS*, pages 1729–1730, 2015. Cited on pages 20 and 21

[255] H. P. Young and A. Levenglick. A consistent extension of Condorcet's election principle. *SIAM. J. Appl. Math.*, 35(2):285–300, 9 1978. Cited on page 5

[256] K. Zhang, M. Deng, T. Chen, M. S. Waterman, and F. Sun. A dynamic programming algorithm for haplotype block partitioning. *Proceedings of the National Academy of Sciences*, 99(11):7335–7339, 2002. Cited on page 20

[257] M. Zuckerman, O. Lev, and J. S. Rosenschein. An algorithm for the coalitional manipulation problem under Maximin. In *AAMAS*, pages 845–852, 2011. Cited on page 5

[258] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. *Artif. Intell.*, 173(2):392–412, 2009. Cited on pages 5 and 148