Foundations of computing Volume 6

Jiehua Chen

Exploiting Structure in Computationally Hard Voting Problems

Universitätsverlag der TU Berlin

Technische Universität Berlin

Jiehua Chen
陈洁花

Exploiting Structure in Computationally Hard Voting Problems

Die Schriftenreihe *Foundations of Computing* der Technischen Universität Berlin wird herausgegeben von:
Prof. Dr. Stephan Kreutzer,
Prof. Dr. Uwe Nestmann,
Prof. Dr. Rolf Niedermeier.

Foundations of Computing | 06

Jiehua Chen
陈洁花

Exploiting Structure in Computationally Hard Voting Problems

Universitätsverlag der TU Berlin

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.dnb.de abrufbar.

Universitätsverlag der TU Berlin, 2016
http://www.verlag.tu-berlin.de

Fasanenstr. 88, 10623 Berlin
Tel.: +49 (0)30 314 76131 / Fax: -76133
E-Mail: publikationen@ub.tu-berlin.de

Zugl.: Berlin, Techn. Univ., Diss., 2015
1. Gutachter: Prof. Dr. Rolf Niedermeier
2. Gutachter: Prof. Dr. Jérôme Lang
3. Gutachterin: Dr. Edith Elkind
Die Arbeit wurde am 18. Dezember 2015 an der Fakultät IV unter Vorsitz von Prof. Dr. Stefan Tai erfolgreich verteidigt.

Druck: docupoint GmbH
Satz/Layout: Jiehua Chen

Umschlagfoto:
Jiehua Chen, 2016

ISBN 978-3-7983-2825-9 (print)
ISBN 978-3-7983-2826-6 (online)

ISSN 2199-5249 (print)
ISSN 2199-5257 (online)

Zugleich online veröffentlicht auf dem institutionellen Repositorium der Technischen Universität Berlin:
DOI 10.14279/depositonce-5113
http://dx.doi.org/10.14279/depositonce-5113

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit Wahlproblemen und den darin auftretenden Strukturen. Einige dieser Strukturen finden sich in den Wählerpräferenzen, wie zum Beispiel die in der Sozialwahltheorie (engl. *social choice theory*) intensiv erforschten *domain restrictions* [ASS02, ASS10], wo die Wählerpräferenzen eine bestimmte eingeschränkte Struktur haben. Andere Strukturen lassen sich wiederum mittels Problemparametern quantitativ ausdrücken, was sie einer parametrisierten Komplexitätsanalyse zugänglich macht [Cyg+15, DF13, FG06, Nie06].

Dieser Zweiteilung folgend ist die Arbeit in zwei Themengebiete untergliedert. Das erste Gebiet beinhaltet Betrachtungen zu Strukturen in Wählerpräferenzen, wie z.B. Single-Crossing-Strukturen oder eindimensionale euklidische Strukturen. Es wird in den Kapiteln 3 bis 5 abgehandelt.

Das zweite Themengebiet umfasst die parametrisierte Komplexitätsanalyse zweier NP-schwerer Wahlprobleme, wobei die neu gewonnenen Erkenntnisse zu den im ersten Teil der Arbeit untersuchten Strukturen verwendet werden. Es beschäftigt sich außerdem mit Fragen sowohl zur klassischen als auch zur parametrisierten Komplexität mehrerer Wahlprobleme für zwei in der Praxis weit verbreitete parlamentarische Wahlverfahren. Dieser Teil der Arbeit erstreckt sich über die Kapitel 6 bis 8.

Kapitel 3 untersucht die Single-Crossing-Eigenschaft. Diese beschreibt eine Anordnung der Wähler, bei der es für jedes Paar von Alternativen höchstens zwei aufeinanderfolgende Wähler gibt, die unterschiedlicher Meinung über die Reihenfolge dieser beiden Alternativen sind. Wie sich herausstellt, lässt sich diese Eigenschaft durch eine endliche Anzahl von verbotenen Strukturen charakterisieren. Ein Wählerprofil ist genau dann *single-crossing*, wenn es keine dieser Strukturen beinhaltet. Es wird außerdem ein Algorithmus vorgestellt, der die Single-Crossing-Eigenschaft unter Verwendung von *PQ trees* [BL76] in $O(n \cdot m^2)$ Schritten erkennt, wobei n die Anzahl der Wähler und m die Anzahl der Alternativen ist.

Kapitel 4 behandelt Wählerprofile, die eindimensional-euklidisch sind, d.h. für die sich die Alternativen und Wähler so auf die reelle Achse abbilden lassen, dass für jeden Wähler und je zwei Alternativen diejenige näher zum Wähler abgebildet wird, die er der anderen vorzieht. Es stellt sich heraus, dass es im Gegensatz zur Single-Crossing-Eigenschaft *nicht* möglich ist, eindimensionale euklidische Profile durch endlich viele verbotene Strukturen zu charakterisieren.

Kapitel 5 beschäftigt sich mit der Frage, wie berechnungsschwer es ist, eine bestimmte strukturelle Eigenschaft wie z.B. die Single-Crossing-Eigenschaft zu erreichen, indem man eine möglichst kleine Anzahl von Wählern oder Kandidaten aus einem Profil entfernt. Es zeigt sich, dass dieses Problem für die Single-Crossing-Eigenschaft durch das Löschen von Wählern zwar in polynomieller Zeit gelöst werden kann, es durch das Löschen von Kandidaten jedoch NP-schwer ist. Für alle anderen Eigenschaften sind beide Varianten des Löschens ebenfalls NP-schwer. Allerdings lässt sich für jedes der Probleme auf triviale Weise mittels des Parameters „Anzahl der zu löschenden Wähler bzw. Alternativen" *fixed-parameter tractability* zeigen. Das bedeutet, dass sie effizient lösbar sind, wenn der Parameter klein ist. Der Grund dafür ist, dass sich alle hier betrachteten Eigenschaften durch eine endliche Anzahl verbotener Strukturen charakterisieren lassen, deren Zerstörung die gewünschte Eigenschaft herstellt.

Kapitel 6 führt die kombinatorische Variante des bekannten Problems CONTROL BY ADDING VOTERS ein, das erstmals durch Bartholdi III, Tovey und Trick [BTT92] beschrieben wurde. In der klassischen Problemstellung gibt es eine Menge von nichtregistrierten Wählern mit bekannten Präferenzen, und es wird eine kleinste Teilmenge von nichtregistrierten Wählern gesucht, sodass deren Hinzufügen zu einem gegebenen Profil einen bestimmten Kandidaten zum Gewinner macht. In der hier beschriebenen Variante wird zusätzlich angenommen, dass für jeden hinzugefügten Wähler auch eine Menge von weiteren Wählern „kostenlos" hinzugefügt werden kann. Dieses Problem wird für die beiden bekannten Wahlregeln Condorcet-Wahl und Mehrheitswahl untersucht. Wie sich herausstellt, ist die Problemstellung schon für zwei Alternativen NP-schwer. Desweiteren werden Parameter identifiziert, die sich aus den kombinatorischen Eigenschaften dieses Problems ergeben. Für diese lässt sich eine beinahe erschöpfende Beschreibung der parametrisierten Komplexität des Problems erstellen. In einem Fall, bleibt unser Problem für sogenannte Single-Peaked-Präferenzen berechnungsschwer, während es für Single-Crossing-Präferenzen in polynomieller Zeit lösbar ist.

Kapitel 7 untersucht, wie verschiedene natürliche Parameter und Preisfunktionen die Berechnungskomplexität des SHIFT BRIBERY-Problems [EFS09] beeinflussen. Darin fragt man, ob eine gegebene Alternative zum Gewinner gemacht werden kann, indem sie in den Präferenzen einiger Wähler nach vorne verschoben wird. Jede Verschiebung hat einen Preis, und das Ziel ist es, ein gegebenes Budget nicht zu überschreiten. Die Ergebnisse sind gemischt: einige Parameter erlauben effiziente Algorithmen, während für andere das Problem schwer bleibt, z.B. für den Parameter „Anzahl der beeinflussten Wähler" ist das Problem sogar W[2]-schwer. Für die Optimierungsvariante von SHIFT BRIBERY, bei der das verwendete Budget minimiert wird, erzielen wir einen Approximationsalgorithmus mit einem Approximationsfaktor von $(1+\varepsilon)$, dessen Laufzeit in ihrem nicht-polynomiellen Anteil nur von ε und der Anzahl der Wähler abhängt.

Kapitel 8 konzentriert sich auf zwei weitverbreitete parlamentarische Wahlregeln: die *successive rule* und die *amendment rule*. Beide Regeln verwenden eine lineare Ordnung der Alternativen, auch *Agenda* genannt. Es werden drei Probleme untersucht: COALITIONAL MANIPULATION fragt nach der kleinstmöglichen Anzahl von Wählern mit beliebigen Präferenzen, deren Hinzufügung einen bestimmten Kandidaten zum Gewinner macht; AGENDA CONTROL fragt, ob es möglich ist, eine Agenda derart festzulegen, dass ein bestimmter Kandidat gewinnt; POSSIBLE/NECESSARY WINNER fragt für unvollständige Wählerpräferenzen und/oder eine nur teilweise festgelegte Agenda, ob eine bestimmte Alternative überhaupt bzw. sicher zum Sieger machen kann. Es stellt sich heraus, dass sowohl COALITIONAL MANIPULATION als auch AGENDA CONTROL für beide Wahlregeln in polynomieller Zeit lösbar sind. Allerdings deuten die Ergebnisse einer auf realem Wählerverhalten basierenden, experimentellen Studie darauf hin, dass die meisten Profile nicht durch einige wenige Wähler manipuliert werden können, und dass eine erfolgreiche Kontrolle mittels Agenda typischerweise nicht möglich ist. POSSIBLE WINNER ist für beide Regeln NP-schwer, während NECESSARY WINNER für die *amendment rule* coNP-schwer und für die *successive rule* in polynomieller Zeit lösbar ist.

Abstract

This thesis explores and exploits structure inherent in voting problems. Some of these structures are found in the preferences of the voters, such as the domain restrictions, which have been widely studied in social choice theory [ASS02, ASS10]. Others can be expressed as quantifiable measures (or *parameters*) of the input, which make them accessible to a parameterized complexity analysis [Cyg+15, DF13, FG06, Nie06].

Accordingly, the thesis deals with two major topics. The first topic revolves around preference structures, for instance, the single-crossing or the one-dimensional Euclidean structures. It is covered in Chapters 3 to 5. The second topic includes the parameterized complexity analysis of two computationally hard voting problems, making use of some of the structural properties studied in the first part of the thesis. It also investigates questions on the computational complexity, both classical and parameterized, of several voting problems for two widely used parliamentary voting rules. It is covered in Chapters 6 to 8.

In Chapter 3, we study the single-crossing property which describes a natural order of the voters such that for each pair of alternatives, there are at most two consecutive voters along this order which differ in their relative ordering of the two alternatives. We find finitely many forbidden subprofiles whose absence from a profile is necessary and sufficient for the existence of single-crossingness. Using this result, we can detect single-crossingness without probing every possible order of the voters. We also present an algorithm for the detection of single-crossingness in $O(n \cdot m^2)$ time via PQ trees [BL76], where n denotes the number of voters and m the number of alternatives.

In Chapter 4, we study the one-dimensional Euclidean property which describes an embedding of the alternatives and voters into the real numbers such that every voter prefers alternatives that are embedded closer to him to those which are embedded farther away. We show that, contrary to our results for the single-crossing property, finitely many forbidden subprofiles are *not* sufficient to characterize the one-dimensional Euclidean property.

In Chapter 5, we study the computational question of achieving a certain property, as for instance single-crossingness, by deleting the fewest number of either alternatives or voters. We show that while achieving single-crossingness by deleting the fewest number of voters can be done in polynomial time, it is NP-hard to achieve this if we delete alternatives instead. Both problem variants are NP-hard for the

remaining popular properties, such as single-crossingness or value-restriction. All these problems are trivially fixed-parameter tractable for the parameter "number of alternatives to delete" (resp. "number of voters to delete") because for each studied property there are finitely many forbidden subprofiles whose removal makes a profile possess this property.

In Chapter 6, we introduce a combinatorial variant of CONTROL BY ADDING VOTERS. In CONTROL BY ADDING VOTERS as introduced by Bartholdi III, Tovey, and Trick [BTT92], there is a set of unregistered voters (with known preference orders), and the goal is to add the fewest number of unregistered voters to a given profile such that a specific alternative wins. In our new model, we additionally assume that adding a voter means also adding a bundle (that is, a subset) of other voters for free. We focus on two prominent voting rules, the plurality rule and the Condorcet rule. Our problem turns out to be extremely hard; it is NP-hard for even two alternatives. We identify different parameters arising from the combinatorial model and obtain an almost complete picture of the parameterized complexity landscape. For the case where the bundles of voters have a certain structure, our problem remains hard for single-peaked preferences, while it is polynomial-time solvable for single-crossing preferences.

In Chapter 7, we investigate how different natural parameters and price function families influence the computational complexity of SHIFT BRIBERY [EFS09], which asks whether it is possible to make a specific alternative win by shifting it higher in the preference orders of some voters. Each shift has a price, and the goal is not to exceed the budget. We obtain both fixed-parameter tractability and parameterized intractability results. We also study the optimization variant of SHIFT BRIBERY which seeks to minimize the budget spent, and present an approximation algorithm which approximates the budget within a factor of $(1+\varepsilon)$ and has a running time whose super-polynomial part depends only on the approximation parameter ε and the parameter "number of voters".

In Chapter 8, we turn our focus to two prominent parliamentary voting rules, the successive rule and the amendment rule. Both rules proceed according to a linear order of the alternatives, called the *agenda*. We investigate COALITIONAL MANIPULATION (which asks to add the fewest number of voters with arbitrary preference orders to make a specific alternative win), AGENDA CONTROL (which asks to design an appropriate agenda for a specific alternative to win), and POSSIBLE/NECESSARY WINNER (which asks whether a specific alternative wins in a/every completion of the profile and the agenda). We show that while COALITIONAL MANIPULATION and AGENDA CONTROL are polynomial-time solvable for both rules, our real-world experimental results indicate that most profiles cannot be manipulated by only few voters, and that a successful agenda control is typically impossible. POSSIBLE WINNER is NP-hard for both rules. While NECESSARY WINNER is coNP-hard for the amendment rule, it is polynomial-time solvable for the successive rule. All computationally hard problems are fixed-parameter tractable for the parameter "number of alternatives".

Preface

This thesis summarizes a large part of my research work from September 2011 to August 2015. During this time, I have been a member of the research group of Prof. Rolf Niedermeier at TU Berlin. I am very grateful to the Studienstiftung des Deutschen Volkes (2011-10-01 to 2014-03-31) and TU Berlin (since 2014-04-01) for financial support.

As the title suggests, my intention in pursuing a Ph.D. was to exploit structures arising in computationally hard voting problems. I attack this goal in two steps. In the first step, I look into voter preferences and their structures. This is a field of active research both in social choice and political sciences [ASS02, ASS10, Bla48, Gae09]. I study structured (Chapters 3 and 4) and nearly structured preferences (Chapter 5). In the second step, I study two computationally hard voting problems (Chapters 6 and 7) and two sequential voting rules together with some computational questions surrounding them (Chapter 8).

Obviously, my research was not done in isolation. Most of the results I present in this thesis have already been published in conference proceedings and journals, resulting from the cooperation with my co-authors. In the following, I will go through every chapter of the main part of my thesis, partly telling some of the history behind the research projects, partly pointing out the main differences between published articles and the results presented in this thesis, and most of all explaining my specific contributions to these results. I do this in the chronological order of when the projects behind the chapters were started, instead of the order of the chapters in this thesis.

Chapter 3: Single-crossing preferences. In the 2012 Dagstuhl seminar on "Computation and Incentives in Social Choice", I gave my first scientific talk. At this seminar, there was a talk on clone structures in elections [EFS12] which was given by Piotr Faliszewski (AGH University of Science & Technology, Krakow). In his talk, Piotr mentioned single-peaked preferences and single-crossing preferences. Right after the talk, knowing that my colleague Robert Bredereck (TU Berlin) and I had recently worked on nearly single-peaked preferences, Gerhard Woeginger (TU Eindhoven) approached us and asked whether we knew the characterization of single-crossing preferences; this was inspired by the work of Ballester and Haeringer [BH11] who characterized single-peaked preferences through two forbidden substructures. I was always fascinated by the idea of Ballester and Haeringer and was of course very curious about whether this also holds for single-crossing preferences. In the same evening, we started searching for small subprofiles precluding which makes a profile single-crossing. We were successful and could characterize the single-crossing preference by finitely many forbidden substructures. We also found that a single-crossing profile with m alternatives can have at most $(m \cdot (m-1)/2+1)$ different preference orders. But, we did not know how to construct one. I thought about this question overnight and found an approach which turned out to be part of the Bruhat order construction approach (Example 3.2). Back in Berlin, I thought more about the proof of the characterization and I discovered that we could directly use the PQ tree algorithm to detect the single-crossing property (Section 3.7). All these results are covered in the journal *Social Choice and Welfare* [BCW13a].

Chapter 5: Nearly structured preferences. Before I started pursuing my Ph.D, Nadja Betzler (who also was my supervisor Rolf Niedermeier's student, but has since January 2011 quit academics) and Robert Bredereck investigated the problem of achieving single-peakedness. They found that deleting the fewest number of voters to achieve single-peakedness is NP-hard, but it is fixed-parameter tractable and admits a polynomial-time kernel for the parameter "number n of voters" to delete. After successfully characterizing single-crossing preferences (Chapter 3), Gerhard Woeginger asked in an email about the computational complexity of achieving the single-crossing property by deleting either the fewest number of voters or the fewest number of alternatives. For the case of voter deletion, his idea was to reduce our problem to finding a maximum-weight path in an directed acyclic graph, which is polynomial-time solvable. When Gerhard visited our research group as a Humboldt award winner, starting from October 2012, we—Robert, Gerhard, and me—continued our research on nearly structured preferences. Besides single-peakedness and single-crossingness, we also considered other preference structures, as for instance worst-restriction, medium-restriction, best-restriction, and group-separability. Later on, I discovered a reduction for β-restricted preferences which is similar to the one given for value-restricted preferences; the β-restriction is necessary for group-separability. The results of this research are presented at the 23rd International Joint Conference on Artificial Intelligence 2013 (IJCAI '13) [BCW13b], and appear in the journal *Mathematical Social Sciences* [BCW16].

Chapter 4: One-dimensional Euclidean preferences. At about the time when we were discussing the voter deletion problem for the single-crossing property (Theorem 5.12), Gerhard Woeginger mentioned the one-dimensional Euclidean structure and asked whether it is possible to characterize it as we did for single-crossingness. I found that the one-dimensional Euclidean structure is a restriction of single-peakedness and single-crossingness, and conjectured that it is equivalent to the single-peaked and single-crossing structure. Soon after I announced my naive conjecture, I discovered (with the help of a program written by Robert) a single-peaked and single-crossing profile with six alternatives and fifteen voters that is *non-1D-Euclidean*; I later shrunk this profile to only three voters and six alternatives, which is the smallest single-peaked and single-crossing profile known to us that is non-1D-Euclidean.

In June 2013, Kirk Pruhs (University of Pittsburgh) visited us in Berlin. After one week of intense research, we—Kirk, Gerhard, and me—came up with an idea of what prevents a single-peaked and single-crossing profile with an arbitrary but even number of alternatives from being 1D-Euclidean. Thus, we could construct single-peaked and single-crossing profiles which are non-1D-Euclidean, but we could not prove that they are minimal with respect to voter deletion. As it turned out, indeed they are not. I found another way of constructing a non-1D-Euclidean profile for an arbitrary number $4k$ of alternatives (Section 4.6), and I could show that deleting any voter makes these profiles 1D-Euclidean. I later managed to generalize the idea of my construction to other non-1D-Euclidean profiles, some of which I could show are minimal with respect to voter deletion, which implies that profiles constructed in Section 4.6 are not sufficient to characterize Euclideanness. Thus, the question of a set of forbidden substructures to characterize the 1D-Euclideanness remains open. The journal version of Chapter 4 appears in the journal *Social Choice and Welfare* [CPW16].

Chapter 7: Shift bribery problems. Piotr Faliszewski visited us in Berlin as a DFG Mercator fellow from fall 2013 to spring 2014. He is an expert in computational social choice including the study of various bribery and control problems. Knowing that our group's research background is parameterized complexity theory, Piotr suggested to have a more fine-grained study of the shift bribery problem, which is computationally hard for many popular voting rules [EFS09]. We—Piotr, Robert, Rolf, André Nichterlein (TU Berlin), and me—studied two aspects of this problem: parameterization and price functions. After intense discussions, we came up with many interesting parameters and price function families for the voters. Among our numerous results I was chiefly responsible for the fixed-parameter tractability result for the parameter "number t of total unit shifts", and the corresponding partial kernelizations. The results of this research are presented at the 28th AAAI Conference on Artificial Intelligence 2014 (AAAI '14) [Bre+14b]. The results of this research are published in the journal *Information and Computation* [Bre+16a].

Chapter 6: Combinatorial voter control problems. After a successful collaboration with Piotr in the study the parameterized shift bribery problem, Nimrod Talmon joined our group in Berlin and received some reading material from Rolf Niedermeier, one of which is about combinational auctions [RPH98]. Nimrod suggested to consider voting problems with combinatorial flavor. Piotr found that this idea suited control problems, such as control by adding voters, very well. We soon focused on combinatorial voter control, where adding a voter also means adding a subset of other voters for free; the preference orders of the added voters are not changeable. We studied this problem for the plurality rule and the Condorcet rule. In March 2014, when I visited Piotr in Krakow, we identified several interesting parameters for our combinatorial voter control problem. We also considered the case with single-peaked preferences and with single-crossing preferences. Among our many results, one I obtained in cooperation with Laurent Bulteau (CNRS, Marne-la-Vallée) is particularly noteworthy. We could show that the problems remain NP-complete even when each added voter may result in at most one additional voter to also join the voting. After we presented our results at the 39th International Symposium on Mathematical Foundations of Computer Science 2014 (MFCS '14) [Che+14], we managed to further show that for the same setting where each added voter results in at most one more additional added voter, the problem is fixed-parameter tractable for the parameter "number of subsets of voters allowed to add". The results of this research are published in the journal *Theoretical Computer Science* [Bul+15].

Chapter 8: Problems around two parliamentary voting rules. While staying in Berlin as a Humboldt award winner in the summer of 2014, Toby Walsh (NICTA and the University of New South Wales) proposed to work on some problems for two parliamentary voting rules, the successive rule and the amendment rule. He already had found out that for both voting rules, coalitional manipulation is polynomial-time solvable while deciding whether an alternative may possibly win with weighted voters is weakly NP-hard. He wanted to know the computational complexity of agenda control and possible/necessary winner. I could show that agenda control is also polynomial-time solvable for both rules. Later I found out that one of my results was already covered in a paper by Miller [Mil80]. Robert Bredereck also joined our research efforts. I am mostly responsible for the polynomial-time solvability results. I also conducted an empirical study for the manipulation and agenda control problems. The results were presented at the 24th International Joint Conference on Artificial Intelligence (IJCAI '15) [Bre+15b].

Besides the work presented in this thesis, I have also worked on several different problems in computational social choice [Alo+13, Alo+15, Bev+14b, Bev+15a, Bre+12, Bre+14c, Che+15a] and several computationally hard problems regarding graphs and hypergraphs [Bev+14a, Bev+16, Che+13, Che+15b], set covering [Bev+15b], team formation [Bre+16b], and matrix explanation [Bre+13, Bre+15a].

Dedicated to my mother 张秀群, *to my father* 陈海涛, *to my sister* 陈洁蓉,
and to my best friend Sven Grottke

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction and Overview

I'm the kind of person that believes there's a part of your voting that has to be purely on principle, and there's a part that has to be on strategy.

Michael Moore

Whenever faced with more than one choice (or *alternative*), different *individuals* may have different opinions upon which choice is better than another. This is true even in daily activities. As a working example, suppose that every day at noon, the employees of a company have to choose where to have lunch. There are three places to go, the nearby (U)niversity's cafeteria, the (I)talian restaurant at the corner, or the (S)taff canteen upstairs. As a new employee, Alex prefers to go to the cheap cafeteria, followed by the staff canteen, and to the Italian restaurant only as a last resort. Ben, being a big fan of fine Italian cuisine, loves dining at the Italian restaurant, followed by the cafeteria because it is outside. Chris is a workaholic who prefers to go upstairs and eat at the staff canteen over going to the other two places. The preferences can be expressed as follows,

$$\begin{array}{llllll} \text{Alex:} & \text{(U)inversity} & \succ & \text{(S)taff} & \succ & \text{(I)talian,} \\ \text{Ben:} & \text{(I)talian} & \succ & \text{(U)inversity} & \succ & \text{(S)taff,} \\ \text{Chris:} & \text{(S)taff} & \succ & \big\{\text{(I)talian} & , & \text{(U)inversity}\big\}, \end{array}$$

where $\succ$ is used to express that the alternative in front of $\succ$ is strictly preferred to the alternative behind it, and alternatives are grouped together inside $\{\}$ when neither is preferred to the other.

To decide on a dining place, a "winner", we need a *voting rule* to aggregate all these preferences. For example, a very intuitive choice of a winner is the alternative which is preferred to every other alternative by a majority of the voters. This rule was originally proposed by de Condorcet [Con85], and it remains one of the most widely studied voting rules in social choice and political sciences. Applying it to our example requires Chris to decide between I and U as his second and third choice. If he prefers U to I, then U wins: Alex and Chris prefer it to I, and Alex and Ben prefer it to S. If, however, Chris prefers I to U, then we run into a problem known as a majority cycle, also called the Condorcet paradox: Alex and Ben prefer U to S, Alex and Chris prefer S to I, and Ben and Chris prefer I to U. Thus, no alternative is preferred to every other alternative by a majority of the voters, and there is no Condorcet winner. This kind of paradox was already observed by Condorcet [Con85].

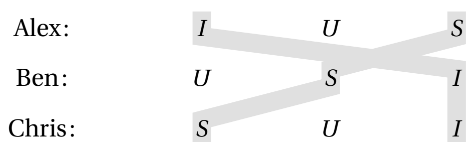
Not being able to choose a winner is problematic in many scenarios other than just deciding on a dining place. One cause of this issue is that we assume no restrictions on the individuals' preferences. There are, however, several well known preference restrictions/properties which rule out the existence of a majority cycle, thus ensuring the existence of a Condorcet winner [Bla48, Pat71, Rob77, Sen70]. For instance, Black [Bla48] observed the so-called *single-peaked* structure in political electorates: the alternatives can be ordered from left to right such that each individual's preferences along this order are either always increasing (that is, he always prefers an alternative to those further to the left), always decreasing (that is, he always prefers an alternative to those further to the right), or first increasing and then decreasing. In our example, if Chris prefers U over I, then the preferences are indeed single-peaked regarding the following left-to-right order:

$$(S, U, I).$$

While dealing with income taxation, Mirrlees [Mir71] and Roberts [Rob77] observed that individuals' preferences are likely to be *single-crossing*. This requires a permutation of the individuals such that for each pair of tax rates (alternatives), there are at most two consecutive individuals along this permutation which differ in their relative ordering of the two rates. Our example, again with Chris having preferences $S \succ U \succ I$, is single-crossing regarding the following permutation:

$$(\text{Alex, Ben, Chris}).$$

For instance, Alex and Ben are the only consecutive individuals that differ in the relative ordering of I and S, as shown in the following figure: the thick gray lines intersect at a point between Alex and Ben, illustrating that the permutation (Alex, Ben, Chris) is single-crossing with respect to $\{I, S\}$.



Individuals and alternatives may even display some kind of spatial structure [BL07, EH08, Poo89, Sto63]. For instance, suppose that the inhabitants in a district need to decide on the location of a new hospital. Typically, for each two options, an inhabitant will prefer the one that is closer to his home. This is an example of so-called *two-dimensional Euclidean preferences*.

Of course, there are other voting rules besides the Condorcet rule, so why do we choose one which may not have a winner? Unfortunately, many classical impossibility results [Arr50, Gib73, Sat75] from social choice theory indicate that designing a "perfect" voting rule is actually impossible: these results show that voting rules cannot satisfy several natural requirements simultaneously, with one of the requirements being that every preference order is allowed.

Even when leaving this difficult issue aside, after having decided on a voting rule, we may encounter questions on the computational cost of problems such as how to determine the outcome, whether it is possible for some individual to obtain an outcome that he prefers more (or even most, in the best case) by voting strategically, or whether the outcome can be manipulated by altering, for instance, the set of alternatives.

For example, in many real-world scenarios, having an efficient algorithm for winner determination or rank aggregation is very important. Consider a meta search engine that computes a consensus ranking by aggregating the results of several independent search engines. Few people would be willing to use the meta search engine if they need to wait several minutes to see a result.

As another example, suppose an online marketplace where customers can rate products, and products are ranked according to an aggregation of these ratings based on a voting rule. A company may try to improve its sales by creating fake customers who give the company's products favorable ratings. Even though Gibbard [Gib73] and Satterthwaite [Sat75] did show that similar manipulations even by just one voter are possible for any reasonable voting rule, showing NP-hardness for the problem of deciding whether a manipulation is successful may still prevent it from being carried out in practice.

Many of such computational problems turn out to have a very high computational complexity. For instance, Bartholdi III, Tovey, and Trick [BTT89] showed that it is NP-hard to decide whether a specific alternative can win under each of several popular voting rules, including a well-known one designed by Charles Lutwidge Dodgson.[1] Bartholdi III and Orlin [BO91] showed that strategic voting for the single transferable voting (STV) rule, which is also known as the Hare system and whose

[1] He is better known as Lewis Carroll, the author of the novel *"Alice's Adventures in Wonderland"*.

variants are used in political elections in several countries is NP-hard as well.

All these intractability results typically assume that the individuals may have arbitrary preferences, and that both the number of individuals and the number of alternatives are unbounded. This beckons the question whether these assumptions are reasonable. How do we characterize preferences which possess structural properties such as single-peakedness? How do we model situations where preferences are close to having a certain structure? Do computationally hard voting problems remain hard for these kinds of preferences? Or can we tackle their apparent intractability by exploiting their structure? We provide some answers to these questions in the remainder of this thesis.

1.1 Thesis overview

Chapter 2 presents relevant notations and concepts that will be used throughout the whole thesis. Chapters 3 to 8 constitute the core of this work and are built around two main topic. The first topic, which is discussed in Part I (Chapters 3 to 5), revolves around structured and nearly structured preferences, including single-peakedness and single-crossingness as introduced at the beginning of this chapter. The second topic, which is covered in Part II (Chapters 6 to 8), deals with computationally hard voting problems that mainly concern manipulative attacks. Chapter 9 concludes this thesis with some future research directions.

In the remainder of this section, we briefly summarize the highlights of Chapters 3 to 8. We remark that throughout this work, except for Chapter 8, we assume that each individual, whom we refer to as a voter, ranks the alternatives according to a linear order; we call this order a *preference order*.

1.1.1 Part I: Structured and nearly structured preferences

The first main topic of this thesis, presented in Part I (Chapters 3 to 5), is about the study of specific preference structures, including single-peakedness, single-crossingness, and one-dimensional Euclideanness. These three structures were described informally at the beginning of the introduction.

Single-crossing preferences. In Chapter 3, we characterize the single-crossing property. In particular, we provide finitely many forbidden substructures whose absence from a profile is sufficient and necessary to make this profile single-crossing. We also present a polynomial-time algorithm for detecting the single-crossing property which is based on the PQ tree algorithm [LB62]. Notably, the single-peaked property can also be decided in polynomial time using a similar approach [BT86].

One-dimensional Euclidean preferences. In Chapter 4, we study the one-dimensional Euclidean property that models the spatial perception of voters' preferences: Voters and alternatives can be placed on the real line such that each voter has a stronger preference for an alternative that is closer to him than for another one that is further away. This property is a restriction of the single-peaked and the single-crossing properties. We show that, contrary to the single-crossing property from Chapter 3, the one-dimensional Euclidean property *cannot* be characterized by *finitely* many forbidden substructures. In our proof, we construct an infinite sequence of sets of preferences such that (a) each of these sets is *not* one-dimensional Euclidean, but (b) any strict subset of it is one-dimensional Euclidean. This construction uses a so-called cyclic relation technique. Interestingly, we can use the same technique to construct other infinite sequences which fulfill Condition (a). We can show Condition (b) for some constructed sets of preferences. This implies that the absence of the constructed sequence is necessary, but not sufficient for a profile to be one-dimensional Euclidean.

Nearly structured preferences. In Chapter 5, we study computational questions arising from the problem of deciding the distance from general preferences to the nearest structured ones, such as single-peaked preferences or single-crossing preferences. We look into two distance measures: the minimum number of voters, and the minimum number of alternatives to delete from a profile to obtain a given property. We show that all considered problems, except for deciding the distance to the nearest single-crossing profile by deleting the minimum number of voters, are NP-hard.

1.1.2 Part II: Computationally hard voting problems

The second main topic of this thesis, presented in Part II (Chapters 6 to 8), deals with computational questions around manipulative attacks, as for instance altering the outcome by adding some unregistered voters, or by shifting some specific alternative higher in the preference orders of some voters.

Combinatorial voter control problems. The voter control problem was introduced by Bartholdi III, Tovey, and Trick, who ask whether it is possible to make a specific alternative win by adding (resp. deleting) a given number of voters from a specific pre-described set. In Chapter 6, we introduce a combinatorial variant of voter control. Specifically, we focus on the problem of making an alternative win by adding voters, where adding a voter also means adding a bundle of other voters at unit cost. Because of its combinatorial nature, this problem turns out to be computationally hard even for only two alternatives and for the *plurality* rule under which an alternative is a winner if and only if it is ranked in first position by most of the voters. This result is widely applicable since for two alternatives, any reasonable voting rule resembles the plurality rule. Nevertheless, we study the structure of this highly intractable problem and identify several tractable special cases.

Shift bribery problems. While the combinatorial voter control problem is about changing the profile by adding voters (with pre-determined preferences), the shift bribery problem [EFS09] is about changing the profile by changing the preferences of the voters. More specifically, shift bribery asks whether it is possible to make a specific alternative win by shifting it higher in the preference order of some voters. Each such shift has a price and the goal is not to exceed a given budget. In Chapter 7, we explore restricted special cases where the prices of the voters or the input may have common structures. For example, each single shift may come at a unit price, or the number of alternatives may be a small constant.

Problems around two parliamentary voting rules. After examining manipulative attacks on arbitrary voting rules, in Chapter 8 we focus on two sequential voting rules which are used in the parliaments of various countries. Both rules use a so-called *agenda* which is a linear order over the alternatives. We study three types of computational problems for both rules. The first type asks whether it is possible to make a specific alternative win by adding a given number of manipulators (with arbitrary preference orders). The second type asks whether it is possible to make a specific alternative win under an appropriately designed agenda. The last type deals with the situation where voters may have incomplete preferences, and asks whether a specific alternative can possibly or necessarily win despite uncertainty in the preferences. We complement our theoretical findings with an empirical study on the first two types of problems using real-world data.[2]

1.2 Published papers

This thesis is based on the following peer-reviewed publications and technical reports.

[2] See Section 9.2 for an explanation why we did not carry out such studies for the problems in Chapters 6 and 7.

Chapter 3: Single-crossing preferences

[BCW13a] R. Bredereck, J. Chen, and G. J. Woeginger. "A Characterization of the Single-Crossing Domain". In: *Social Choice and Welfare* 41(4), 2013, pp. 989–998.

Chapter 4: One-dimensional Euclidean preferences

[CPW15] J. Chen, K. Pruhs, and G. J. Woeginger. "*The one-dimensional Euclidean domain: Finitely many obstructions are not enough*". Technical Report: arXiv:1506.03838v1 [cs.GT], 2015.

[CPW16] J. Chen, K. Pruhs, and G. J. Woeginger. "The one-dimensional Euclidean domain: Finitely many obstructions are not enough". In: *Social Choice and Welfare*, 2016, to appear.

Chapter 5: Nearly structured preferences

[BCW13b] R. Bredereck, J. Chen, and G. J. Woeginger. "Are There Any Nicely Structured Preference Profiles Nearby?" In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*, 2013, pp. 62–68.

[BCW16] R. Bredereck, J. Chen, and G. J. Woeginger. "Are There Any Nicely Structured Preference Profiles Nearby?" In: *Mathematical Social Sciences* 79, 2016, pp. 61–73.

Chapter 6: Combinatorial voter control problems

[Che+14] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. "Combinatorial Voter Control in Elections". In: *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14)*, 2014, pp. 153–164.

[Bul+15] L. Bulteau, J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. "Combinatorial Voter Control in Elections". In: *Theoretical Computer Science* 589, 2015, pp. 99–120.

Chapter 7: Shift bribery problems

[Bre+14b] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. "Prices Matter for the Parameterized Complexity of Shift Bribery". In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '14)*, 2014, pp. 1398–1404.

[Bre+16a] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. "Prices Matter for the Parameterized Complexity of Shift Bribery". In: *Information and Computation*, 2016, to appear.

Chapter 8: Problems around two parliamentary voting rules

[Bre+15b] R. Bredereck, J. Chen, R. Niedermeier, and T. Walsh. "Parliamentary Voting Procedures: Agenda Control, Manipulation, and Uncertainty". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI '15)*, 2015, pp. 164–170.

I also contributed to the following peer-reviewed publications which are not covered in this thesis.

[Alo+13] N. Alon, R. Bredereck, J. Chen, S. Kratsch, R. Niedermeier, and G. J. Woeginger. "How to put through your agenda in collective binary decisions". In: *Proceedings of the 3rd International Conference on Algorithmic Decision Theory*. Vol. 8176. Lecture Notes in Computer Science. 2013, pp. 30–44.

[Alo+15] N. Alon, R. Bredereck, J. Chen, S. Kratsch, R. Niedermeier, and G. J. Woeginger. "How to put through your agenda in collective binary decisions". In: *ACM Transactions on Economics and Computation* 4(1) (2015), pp. 1–28.

[Bev+14a] R. van Bevern, R. Bredereck, L. Bulteau, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. "Star partitions of perfect graphs". In: *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP '14)*. Vol. 8572. Lecture Notes in Computer Science. Springer, 2014, pp. 174–185.

[Bev+14b] R. van Bevern, R. Bredereck, J. Chen, V. Froese, R. Niedermeier, and G. J. J. Woeginger. "Network-based dissolution". In: *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14)*. Vol. 8635. LNCS. Springer, 2014, pp. 69–80.

[Bev+15a] R. van Bevern, R. Bredereck, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. "Network-based vertex dissolution". In: *SIAM Journal on Discrete Mathematics* 29(2) (2015), pp. 888–914.

[Bev+15b] R. van Bevern, J. Chen, F. Hüffner, S. Kratsch, N. Talmon, and G. J. Woeginger. "Approximability and parameterized complexity of multicover by c-intervals". In: *Information Processing Letters* 115(10) (2015), pp. 744–749.

[Bev+16] R. van Bevern, R. Bredereck, L. Bulteau, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. "Star partitions of perfect graphs". In: *Journal of Graph Theory* (2016). Accepted.

[Bre+12] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, and O. Suchý. "A multivariate complexity analysis of lobbying in multiple referenda". In: *Proceedings of the 26th Conference on Artificial Intelligence*. AAAI Press, 2012, pp. 1292–1298.

[Bre+13] R. Bredereck, J. Chen, S. Hartung, C. Komusiewicz, R. Niedermeier, and O. Suchý. "On explaining integer vectors by few homogenous segments". In: *Proceedings of the 13th International Workshop on Algorithms and Data Structures (WADS '13)*. Vol. 8037. Lecture Notes in Computer Science. Springer, 2013, pp. 207–218.

[Bre+14c] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, O. Suchý, and G. J. Woeginger. “A multivariate complexity analysis of lobbying in multiple referenda”. In: *Journal of Artificial Intelligence Research* 50 (2014), pp. 409–446.

[Bre+15a] R. Bredereck, J. Chen, S. Hartung, C. Komusiewicz, R. Niedermeier, and O. Suchý. “On explaining integer vectors by few homogeneous segments”. In: *Journal of Computer and System Sciences* 81(4) (2015), pp. 766–782.

[Bre+16b] R. Bredereck, J. Chen, F. Hüffner, and S. Kratsch. “Parameterized complexity of team formation in social networks”. In: *Proceedings of the 11th International Conference on Algorithmic Aspects in Information and Management (AAIM ’16)*. Vol. 9778. Lecture Notes in Computer Science. Springer, 2016, pp. 137–149.

[Che+13] J. Chen, C. Komusiewicz, R. Niedermeier, M. Sorge, O. Suchý, and M. Weller. “Effective and efficient data reduction for the subset interconnection design problem”. In: *Proceedings of the 24th International Symposium on Algorithms and Computation (ISAAC ’07)*. Vol. 8283. Lecture Notes in Computer Science. 2013, pp. 361–371.

[Che+15a] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. “Elections with few voters: Candidate control can be easy”. In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI ’15)*. AAAI Press, 2015, pp. 2045–2051.

[Che+15b] J. Chen, C. Komusiewicz, R. Niedermeier, M. Sorge, O. Suchý, and M. Weller. “Polynomial-time data reduction for the subset interconnection design problem”. In: *SIAM Journal on Discrete Mathematics* 29(1) (2015), pp. 1–25.

CHAPTER 2

Preliminaries and Notation

We could, of course, use any notation we want; do not laugh at notations; invent them, they are powerful. In fact, mathematics is, to a large extent, invention of better notations.

Richard P. Feynman, 1963

In this section, we introduce concepts and basic notation that will be used throughout the thesis.

2.1 Relations and orders

We recall some basic definitions from set theory. Consider a finite set X of elements. A *relation* R, short for *binary relation*, over X is a set of ordered pairs of elements from X. We sometimes omit the underlying set X for the relation R if it is clear from the context.

A binary relation R on X may satisfy one or more of the following properties.

Reflexivity	$\forall x \in X\colon (x,x) \in R$.
Antisymmetry	$\forall x \neq y \in X\colon (x,y) \in R \Rightarrow (y,x) \notin R$.
Transitivity	$\forall x,y,z \in R\colon (x,y),(y,z) \in R \Rightarrow (x,z) \in R$.
Totality	$\forall x,y \in R\colon (x,y) \notin R \Rightarrow (y,x) \in R$.

A *partial order* is a reflexive, antisymmetric, and transitive relation. A *linear (or total) order* is a total partial order. For the same set of elements, there are many more partial orders than linear orders. For instance, for a set of 3 elements, by routine calculation, we see that the number of different partial orders is 19 while the number of different linear orders is 6.

In Section 2.3, we introduce the concept of preference orders, which are basically partial orders, over alternatives. There, we use infix notation to indicate the relation of two distinct alternatives.

2.2 Graphs

We recall some basic definitions from graph theory. We consider two types of graphs: one with undirected edges and the other with directed edges (or arcs).

2.2.1 Undirected graphs

A *graph* G, short for *undirected graph*, is a tuple (V,E) consisting of a set V of *vertices* and a set $E \subseteq \binom{V}{2}$ of *edges* (that is, unordered pairs of vertices) where each edge *connects* two distinct vertices, and there is at most one edge between two vertices. We also denote the vertex set of G by $V(G) := V$ and the edge set by $E(G) := E$. Two vertices $u, v \in V(G)$ are *adjacent*, or *neighbors*, if $\{u, v\}$ is an edge in $E(G)$. We also say that vertices u and v are *incident* with edge $\{u, v\}$.

A *path* L in a graph G is a sequence of vertices $\langle u_1, u_2, \ldots, u_t\rangle$, $t \geq 1$, where for each $i \in \{1,2,\ldots,t-1\}$, it holds that $\{u_i, u_{i+1}\} \in E(G)$. We also say that L is a *path between vertices* u_1 *and* u_t. Graph G is *connected* if it contains a path between each pair of vertices $u, v \in V(G)$.

By deleting some of the vertices and edges of graph G, we can obtain a subgraph of G. Formally, if $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$, then graph G' is a *subgraph* of G, denoted $G' \subseteq G$, and G is a *supergraph* of G'.

We consider two special subgraphs G' of G. If G' contains all edges $e \in E(G)$ with $e \subseteq V(G')$, then G' is an *induced subgraph* of G; we also say that G' is *induced* by $V(G')$ in G. A subset $V' \subseteq V(G)$ of r vertices is an *r-clique* if it induces a subgraph such that any two vertices $u, v \in V'$ are adjacent. We sometimes also call the corresponding induced subgraph a *clique of order r*. If G is a $|V(G)|$-clique, then G is *complete*.

2.2.2 Directed graph

A *directed graph* F is a tuple consisting of a set of vertices and a set of *directed edges*, called *arcs*, where each arc is an ordered pair of two distinct vertices. Just as for undirected graphs, we denote the vertex set of F by $V(F)$ and the arc set of F by $E(F)$. *Subgraphs* of a directed graph are defined in the same way as those of an undirected graph. Given an arc $(u, v) \in E(F)$, we say that u is v's *in-neighbor* and that v is u's *out-neighbor*. The *in-degree* (resp. *out-degree*) of a vertex u is the number of its in-neighbors (resp. out-neighbors).

If F contains exactly one arc between any two distinct vertices, then F is a *tournament*, short for *tournament graph*.

We sometimes label each edge (or arc) with a non-negative integer which we call a *weight*. In this case, we call the corresponding graph a *weighted* graph.

The underlying undirected graph of a directed graph F is an undirected graph G'

(a) **(b)**

Figure 2.1.: (a) An undirected graph on vertex set $\{a, b, c\}$. The circles with labels denote the vertices. The lines between two circles denote the edges. It is a complete graph. (b) A weighted directed graph for the same vertex set. It is a tournament.

with the same vertex set $V(G') := V(F)$ and edge set $E(G') := \{\{a, b\} \mid (a, b) \in E(F)\}$.

Example 2.1. Figure 2.1(a) depicts a complete graph G with three vertices, a, b, c, and Figure 2.1(b) a weighted directed graph F with the same vertices. Note that graph G is complete and graph F is a tournament. Graph G can also be seen as the underlying undirected graph of graph F.

We use the standard definitions of *connectivity* and *connected components* for directed graphs: A directed graph F is *connected*, short for *weakly connected*, if its underlying undirected graph is connected. A *strongly connected component* of F is a subgraph of F, maximal with respect to inclusion of vertex sets, such that for each two vertices $a, b \in V(F)$, there is a directed path from a to b.

2.3 Preference profiles

Let $A = \{a_1, a_2, \ldots, a_m\}$ be a set of m alternatives and let $V = \{v_1, v_2, \ldots, v_n\}$ be a set of n voters, $m, n \geq 2$. A *preference order* over A, denoted $\succ$, is a partial order of A. We call a preference order $\succ$ *complete* or *linear* if it is a linear order. Otherwise, we call it *incomplete*.

Definition 2.1 (Preference profiles)**.** A *preference profile* $\mathscr{P}$ over a set V of voters and a set A of alternatives specifies the preferences of each of the voters, that is, each voter $v_i \in V$ *ranks* the alternatives A according to a partial order $\succ_i$ over A. We say that $\succ_i$ is the *preference order* of voter v_i. We write $\mathscr{A}(\mathscr{P})$ to denote the alternative set A, $\mathscr{V}(\mathscr{P})$ to denote the voter set V, and $\mathscr{R}(\mathscr{P})$ to denote the corresponding collection $(\succ_1, \succ_2, \ldots, \succ_n)$ of preference orders. We sometimes simply write $\succ$ instead of $\succ_i$ if it is clear whose preference order we refer to. We call a preference profile with n voters and m alternatives an *$n \times m$ profile.*

For two alternatives $b, c \in A$, the relation $b \succ_i c$ means that voter v_i strictly *prefers* b over c. The relation $b \sim_i c$ means that voter v_i regards b and c are incomparable to each other. We say that a voter *ranks* an alternative in the *first (resp. second, third, etc.) position* if this alternative is his most (resp. second most, third most, etc.) preferred alternative. Given two disjoint subsets $B, C \subseteq A$ of alternatives, we write $B \succ_i C$ to express that voter v_i prefers each alternative $b \in B$ to each alternative $c \in C$, that is, $b \succ_i c$, and all alternatives in B (resp. C) are *incomparable* to each other. We write $B \succ_i c$ as shorthand for $B \succ_i \{c\}$, and $c \succ_i B$ for $\{c\} \succ_i B$. If not stated explicitly, we write $\langle B \rangle$ to denote an arbitrary but fixed preference order over B. We write $\overleftarrow{\langle B \rangle}$ to denote the corresponding reverse order.

We illustrate the concept of preference profiles in the following example.

Example 2.2. Consider three alternatives a, b, c and two voters v_1 and v_2 with the following preference orders:

$$\begin{aligned}&\text{voter } v_1\colon\ a \succ_1 b \succ_1 c,\\&\text{voter } v_2\colon\ \{c, b\} \succ_2 a.\end{aligned}$$

The first voter's preference order is complete while the second voter's is not. A preference profile $\mathscr{P}$ with alternative set $\mathscr{A}(\mathscr{P}) := \{a, b, c\}$, voter set $\mathscr{V}(\mathscr{P}) := \{v_1, v_2\}$, and preference order collection $\mathscr{R}(\mathscr{P}) := (\succ_1, \succ_2)$ is a 2×3 profile.

2.4 Voting rules and their properties

In the previous section, we have defined how voters may express their opinions on the ranking of the alternatives. Still, we require some rule to aggregate the voter preferences to decide on an overall *winner*. Formally, a *voting rule* is a function that takes as input a preference profile $\mathscr{P}$ with complete preference orders and outputs a subset $A' \subseteq \mathscr{A}(\mathscr{P})$ of alternatives as co-winners. Note that we implicitly assume that all voting rules that we consider are *anonymous*. That is, their outcomes depend only on the numbers of voters with particular preference orders and not on the identities of the particular voters that cast them.

In this section, we describe three popular voting rules. Note that they all assume complete preference orders.

Definition 2.2 (Plurality and majority rules)**.** Given a profile $\mathscr{P}$ with complete preference orders, let the *score* of an alternative $a \in \mathscr{A}(\mathscr{P})$, written $\mathsf{score}_{\mathscr{P}}(a)$, be the number of voters that rank this alternative in the first position. A *plurality winner* is an alternative with maximum score. A *majority winner* is an alternative whose score is more than the half of the number of voters.

It is easy to see that a majority winner is also a plurality winner. While a profile may not have a majority winner, if there is one, then she is unique.

Example 2.3. Consider the 3×4 profile depicted as follows.

voter v_1: $a \succ_1 d \succ_1 b \succ_1 c$,
voter v_2: $c \succ_2 d \succ_2 a \succ_2 b$,
voter v_3: $b \succ_3 d \succ_3 c \succ_3 a$.

It has three plurality winners, a, b, and c, but no majority winner.

The next voting rule is based on pairwise comparisons of any two alternatives.

Definition 2.3 (Condorcet rule [Con85])**.** Given a profile $\mathscr{P}$ with complete preference orders, we say that an alternative $a \in \mathscr{A}(\mathscr{P})$ *beats* another alternative $b \in \mathscr{A}(\mathscr{P})$ if a majority of voters prefers a to b. Accordingly, we say b loses to a. A *Condorcet winner* is an alternative that beats all other alternatives.

Just as for the majority winner, a profile may not have a Condorcet winner. If it has one, then she is unique. Alternative d is a (and the only) *Condorcet winner* of the 3×4 profile depicted in Example 2.3. A majority winner is necessarily a Condorcet winner.

We may want our voting rules to possess some nice properties which are so natural that some of them even called axioms in the literature [ASS02, Gae09].

Condorcet consistency The voting rule selects the Condorcet winner as a winner if she exist.

Non-dictatorship No voter can determine the winner alone.

Weak Pareto efficiency An alternative is not a winner if all voters prefer another one to it.

Strategy-proofness (or non-manipulability) No voter can obtain a better outcome by altering his preference order.

Surjectivity Every alternative can be a winner for some preference profile.

While these properties are desirable, the well-known Gibbard-Satterthwaite theorem [Gib73, Sat75] showed that basically any surjective, non-dictatorial voting rule is manipulable. One possibility to circumvent this situation is through computational complexity. The goal is to show that it is computationally hard to manipulate a given voting rule. In the next section, we recall some basic definitions from the theory of computational complexity.

We close this section by mentioning that in this thesis, we mainly focus on *choice functions*, that is, voting rules which decide upon a subset of co-winners; the plurality rule decides at least one co-winner while the majority rule and the Condorcet rule may have empty co-winners. Yet, sometimes, we need *a welfare function* to aggregate preferences into a joint preference order. We can extend an arbitrary choice function to construct such an order by repeatedly computing a set of co-winners, and removing them from the profile; co-winners are ranked according to some tie-breaking rule. Similarly to the choice functions, welfare functions also face the fate that basically the only welfare function to satisfy some natural properties, including for instance, a preference order variant of the weak Pareto efficiency, is dictatorial [Arr50].

2.5 Computational complexity

Given a preference profile and a voting rule, it is interesting to know the computational complexity of determining a winner, of manipulating this voting rule, or of determining a possible (or necessary) winner when the profile has incomplete preferences. To understand this, we need a framework to formalize the computational complexity [AB09, GJ79, Pap94]. We recall some basic notions from computational complexity theory.

Consider a decision problem L over a finite alphabet Σ. We say that problem L is polynomial-time solvable, and hence, belongs to the complexity class P, if there is an algorithm that *decides* whether a given instance $x \in \Sigma^*$ is a yes-instance of L in $|x|^{O(1)}$ time. In this case, we say that the algorithm is *efficient*, and L is *tractable*.

We say that L belongs to the complexity class NP if there is a polynomial-time algorithm that, given an instance and a polynomial-size information (a certificate) about this instance, can verify whether this instance is a yes-instance of L. We say that problem L is NP*-hard* if for each problem $L' \subseteq \Sigma^*$ in NP, there is a *polynomial reduction* from L' to L:

Definition 2.4 (Polynomial reductions)**.** Given two finite alphabets Γ and Σ, a *polynomial reduction* from problem $L' \subseteq \Gamma^*$ to problem $L \subseteq \Sigma$ is a mapping f from Γ^* to Σ^* which maps each instance $x \in \Gamma^*$ to an instance $f(x) \in \Sigma^*$ such that x is a yes-instance of L' if and only if $f(x)$ is a yes-instance of L with f being a polynomial-time computable algorithm. We sometimes abbreviate polynomial reductions to just *reductions*.

Bartholdi III, Tovey, and Trick [BTT89] show that it is NP-hard to determine whether an alternative is a winner under a Condorcet-consistent voting rule designed by Lewis Carroll. NP-complete problems are problems that are NP-hard and belong to NP. Strictly speaking, the complexity class P only concerns decision problems. Sometimes, we also say that some search problem belongs to P, meaning that there is a polynomial-time algorithm finding a solution to the problem. For instance, finding a plurality winner can be done in linear-time. Accordingly, we say that some search problem is NP-hard if its decision variant is NP-hard, that is deciding "is there a solution?" or "is there a solution of a certain quality?" is NP-hard.

2.6 Parameterized complexity

Every known algorithm solving an NP-hard problem requires superpolynomial running time. Nevertheless, a refined analysis of the problem may yield an algorithm that runs in polynomial time if some problem-specific integer parameter is a constant, where the degree of the polynomial in the running time is independent of the parameter. Parameterized complexity theory deals with such refined analysis.

Definition 2.5 (Parameterized problems and fixed-parameter tractability)**.** Given a finite alphabet Σ, a *parameterized problem* L is a subset $L \subset \Sigma^* \times \mathbb{N}$, and the second entry of an instance in $\Sigma^* \times \mathbb{N}$ is called *parameter*. We say that L is *fixed-parameter tractable* if there is an algorithm deciding whether a given instance $(x, k) \in \Sigma^* \times \mathbb{N}$ belongs to L in $f(k) \cdot |x|^{O(1)}$ time, where f is a computable function depending on the parameter k only. We also call such algorithms *fixed-parameter algorithms*. The class FPT contains all parameterized problems that are fixed-parameter tractable.

To illustrate the concept of fixed-parameter tractability, let us consider the following NP-complete problem, which is perhaps the most and the best studied problem in the field of fixed-parameter algorithms.

> VERTEX COVER
> **Input:** An undirected graph $G = (U, E)$ and an integer $k \leq |U|$.
> **Question:** Is there a *vertex cover* $U' \subseteq U$ of at most k vertices, that is, $|U'| \leq k$ and $\forall e \in E \colon e \cap U' \neq \emptyset$?

It is easy to design a recursive algorithm that runs in $O(2^k \cdot |G|)$ time, where k is the size of the vertex cover and $|G|$ is the size of the input graph: Begin with an empty set U' of vertices. Then, recursively pick an arbitrary edge $\{u, v\}$ that is not covered by U', that is, $\{u, v\} \cap U' = \emptyset$, and branch into two cases: add either u or v to the set U'. Stop when the size of U' exceeds k. Thus, VERTEX COVER parameterized by the vertex cover size k is fixed-parameter tractable.

We emphasize that a fixed-parameter algorithm is in general more efficient than an algorithm that runs in polynomial time for constant parameters and that a running time of the form $|x|^{f(k)}$ does not show fixed-parameter tractability. Problems which can be solved by the second kind of algorithms form the complexity class XP.

2.6.1 Fixed-parameter algorithms and polynomial kernel lower bounds

Systematic approaches to finding fixed-parameter algorithms include *problem kernelization*, *bounded search tree*, and *integer linear programming*, to name but a few. We describe these three techniques briefly.

Problem kernelization. The basic idea behind *problem kernelization* is to "shrink" in polynomial time the input instance to an equivalent instance whose size is upper-bounded by a function f of the parameter.

Definition 2.6 (Kernelization algorithms)**.** A parameterized problem $L \subset \Sigma^* \times \mathbb{N}$ for a given finite alphabet Σ admits a *kernelization algorithm*, or simply *kernelization*, if there is an algorithm that, given an instance (I, k) of L runs in polynomial time in the instance size $|(I, k)|$ and returns an instance (I', k') of L such that

- $|(I', k')| \leq f(k)$ for some computable function f and
- $(I, k) \in L$ if and only if $(I', k') \in L$.

We will see an application of kernelization in Section 7.5.1.

We call the shrunk (equivalent) instances (I', k') *problem kernels*. It has been shown that being fixed-parameter tractable is equivalent to admitting a problem kernel (see [FG06]). Often, the problem kernels we obtain are of super-polynomial size. It is, however, interesting to search for kernelization algorithms that produce problem kernels with size polynomial in the parameter, that is, the function f in Definition 2.6 is a polynomial. For example, using the Buss kernelization [BG93], we can obtain a problem kernel of size $O(k^2)$ for VERTEX COVER, where k is the size of the vertex cover; see, for instance, the work of Niedermeier [Nie06, Chapter 7] for a detailed analysis of the problem kernel. A polynomial kernel reflects in a sense the effectiveness of the kernelization, or preprocessing. However, sometimes we cannot find a polynomial kernel. In fact, there are parameterized problems that do not admit polynomial kernels under some complexity-theoretic assumptions (that is, $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$) [BJK14, Bod+09, FS11]. It is known that $\mathsf{NP} \subseteq \mathsf{coNP/poly}$ would imply a collapse of the polynomial hierarchy to its third level [Yap83]. Bodlaender et al. [Bod+09] introduced the so-called *composition technique*. Later, Bodlaender, Thomassé, and Yeo [BTY11] introduced the concept of *polynomial-parameter transformation*, which describes a reduction from one parameterized problem to another one that produces polynomial problem kernels.

Definition 2.7 (Polynomial-parameter transformation)**.** Let L and $L' \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems for some finite alphabet Σ. A *polynomial-parameter* *transformation* from L to L' is an algorithm that, given an instance $(I,k) \in \Sigma^* \times \mathbb{N}$, computes an instance $(I',k') \in \Sigma^* \times \mathbb{N}$ in time polynomial in $|(I,k)|$ such that

- k' is polynomially bounded in k, and
- $(I,k) \in L$ if and only if $(I',k') \in L'$.

We can show the absence of polynomial kernels by providing a polynomial-parameter transformation from a parameterized problem that is known not to possess a polynomial kernel:

Proposition 2.1 ([BTY11])**.** *Let L and $L' \subseteq \Sigma^* \times \mathbb{N}$ be two parameterized problems for some finite alphabet Σ such that the unparameterized versions of both L and L' are* NP*-complete. If there is a polynomial-parameter transformation from L to L', then L admits a polynomial kernel if L' admits a polynomial kernel.*

We discuss the issue of whether there is a polynomial kernel for some parameterized problems in Chapters 6 to 8. We refer to the work of Cygan et al. [Cyg+15, Chapter 15] for more information on this issue.

Search tree. The search tree technique is of recursive nature. In a nutshell, a search tree algorithm maintains "partial solutions" and recursively calls itself on several subproblems, extending the current partial solution by at least one "element". In this way, the algorithm traverses a tree where the root represents an initial empty solution, each node represents a partial solution and each edge represents the extension from a corresponding partial solution to another. If (1) extending a partial solution can be done in polynomial time, and (2) the number of children of each node and the height of the tree is upper-bounded in a function that only depends on the parameter, then we can easily verify that our algorithm runs in "FPT time". We call such algorithm a *bounded search tree*. We have already seen a simple search tree algorithm for the VERTEX COVER problem parameterized by the vertex cover size at the beginning of Section 2.6.

Integer-linear programming approach. Integer-linear programming centers on the following NP-complete problem [GJ79].

Definition 2.8. INTEGER LINEAR PROGRAMMING FEASIBILITY
Input: An integer $m' \times n'$-matrix $A \in \mathbb{Z}^{m' \times n'}$ and an integer vector $b \in \mathbb{Z}^{m'}$.
Question: Is there an integer vector $x \in \mathbb{Z}^{n'}$ with $A \cdot x \leq b$?

The search variant of INTEGER LINEAR PROGRAMMING FEASIBILITY aims at finding an integer vector of n' variables fulfilling the constraints indicated by the *constraint matrix* A and the *goal vector* b. Despite the NP-hardness, a famous algorithm by Lenstra [Len83] shows that any INTEGER LINEAR PROGRAMMING FEASIBILITY instance with an $m' \times n'$-matrix A and goal vector b can be solved in $f(n') \cdot |(A,b)|^{O(1)}$ time, where f is a function solely dependent on n', and $|(A,b)|$ denotes the number of bits in the binary representation of A and b. In other words, Lenstra's result directly implies fixed-parameter tractability for the parameter "number n' of variables". Frank and Tardos [FT87] and Kannan [Kan87] improve the FPT running time. Together, we obtain the following, which contributes to the best known FPT running time for this setting.

Theorem 2.2 ([FT87, Kan87, Len83])**.** INTEGER LINEAR PROGRAMMING FEASIBILITY *can be solved in* $O((n')^{2.5 \cdot n' + o(n')} \cdot |(A,b)|)$ *time, where* A *denotes the input matrix with* n' *columns,* b *denotes the goal vector, and* $|(A,b)|$ *denotes the number of bits in the binary representation of* A *and* b*.*

Throughout this thesis, we use the function $\mathrm{ilp}(\rho_1, \rho_2, \rho_3)$ to denote the running time of an INTEGER LINEAR PROGRAMMING FEASIBILITY instance with A being a constraint matrix with ρ_1 columns and ρ_2 rows, and b being a goal vector, such that the absolute value of each coefficient in A and of each entry in b is at most ρ_3.

If we can reduce a given parameterized problem to INTEGER LINEAR PROGRAMMING FEASIBILITY such that the number ρ_1 of variables depends only on our parameter (and ρ_2, ρ_3 are not excessively large), then we immediately obtain fixed-parameter tractability for the desired parameter. We utilize the integer-linear programming technique to obtain fixed-parameter tractability results, as stated for instance in Theorem 6.10, and Corollaries 8.17 and 8.23. Obviously, the running time of our relevant problem depends heavily on the running time of INTEGER LINEAR PROGRAMMING FEASIBILITY (with the appropriate instance setting). Thus, we call the corresponding fixed-parameter tractability class ILP-FPT.

2.6.2 Parameterized intractability

Parameterized complexity theory also provides a hierarchy of hardness classes, starting with W[1], such that

$$\mathsf{FPT} \subseteq \mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \ldots \subseteq \mathsf{XP}.$$

One can show that a parameterized problem L is (presumably) not fixed-parameter tractable by devising a *parameterized reduction* from a W[1]-hard (or a W[2]-hard problem) to L.

Definition 2.9 (Parameterized reduction)**.** A *parameterized reduction* from a parameterized problem L to another parameterized problem L' is a function that, given an instance (x,k), computes an instance (x',k') in $f(k)\cdot|x|^{O(1)}$ time such that

- $k' \le g(k)$ and
- $(x,k) \in L$ if and only if $(x',k') \in L'$,

where f and g are two computable functions.

The class W[1] contains all parameterized problems L such that there is a parameterized reduction from L to CLIQUE parameterized by the clique size h, which is complete for W[1].

CLIQUE
Input: An undirected graph $G=(V(G),E(G))$ and a non-negative integer $h \in \mathbb{N}$.
Question: Does G admit a size-h *clique*, that is, a size-h vertex subset $U \subseteq V(G)$ such that $G[U]$ is complete?

The class W[2] contains all parameterized problems L' such that there is a parameterized reduction from L' to SET COVER parameterized by the set cover size h, which is complete for W[2].

SET COVER
Input: A family $\mathcal{F} = (S_1,\dots,S_r)$ of sets over a universe $\mathcal{U} = \{u_1,\dots,u_s\}$ of elements and a non-negative integer $h \ge 0$.
Question: Is there a size-at-most-h *set cover*, that is, a collection $\mathcal{F}'$ of h sets in $\mathcal{F}$ whose union is $\mathcal{U}$?

Unless W[1] = FPT (resp. W[2] = FPT), no fixed-parameter algorithms exist for W[1]-complete (resp. W[2]-complete) problems.

We refer the reader to the textbooks of Cygan et al. [Cyg+15], Downey and Fellows [DF13], Flum and Grohe [FG06], and Niedermeier [Nie06] for more information on parameterized complexity and algorithms. Besides this, Betzler et al. [Bet+12] and Bredereck et al. [Bre+14a] survey voting problems and research challenges involving parameterized algorithms. We use parameterized complexity analysis for voting problems in Chapters 6 to 8.

Finally, we refer the readers to the books of Brandt et al. [Bra+16] and Rothe [Rot15] for general accounts on computational social choice.

Part I

Structured and Nearly Structured Preferences

CHAPTER 3

Single-Crossing Preferences

It may be noted that the assumption requires that there be a 'natural' ordering of individuals, whereas single-peakedness requires that there be a 'natural' ordering of options.

Kevin W.S. Roberts, 1977

We characterize single-crossing profiles in terms of two forbidden substructures, one containing three voters and six alternatives, and the other one containing four voters and four alternatives. We also provide an efficient way to decide whether a preference profile is single-crossing.

3.1 Introduction

Single-peaked and single-crossing preferences have become standard domain restrictions in many political, economic, and psychological models[Bla48, Coo64, Mir71, Rob77]. A preference profile is *single-peaked* if there is a linear order of the alternatives such that every voter's preferences over the alternatives along this order is either always strictly increasing, always strictly decreasing, or first strictly increasing and then strictly decreasing. A preference profile is *single-crossing* if there is a linear order of the voters such that each pair of alternatives separates this order into two sub-orders where in each sub-order, all voters agree on the relative order of this pair (see Figure 3.1 for an illustration of this concept). In many situations, these assumptions guarantee the existence of a strategy-proof voting rule, or the existence of a Condorcet winner.

Single-peaked preferences go back to the work of Black [Bla48] and have been studied extensively over the years. Single-peakedness implies a number of nice properties, as for instance strategy-proofness of a family of voting rules [Mou80] and transitivity of the majority relation [Ina69] (also known as Condorcet principle; we

This chapter is based on "A Characterization of the Single-Crossing Domain" by R. Bredereck, J. Chen, and G. J. Woeginger, *Social Choice and Welfare* [BCW13a] .

discuss this in Chapter 5). Single-crossing preferences go back to the work of Karlin [Kar68] in applied mathematics and the papers of Mirrlees [Mir71] and Roberts [Rob77] on income taxation. Grandmont [Gra78], Rothstein [Rot90], and Gans and Smart [GS96] analyzed various aspects of the majority rule under single-crossing preferences. Furthermore, single-crossing preferences play a role in the areas of income redistribution [MR81], coalition formation [Dem94, Kun06], local public goods distribution and stratification [EP98, Wes77], and in the choice of constitutional voting rules [BJ04]. Saporiti and Tohmé [ST06] studied single-crossing preferences in the context of strategic voting and the median choice rule, and Saporiti [Sap09] investigated them in the context of strategy proof social choice functions. Barberà and Moreno [BM11] developed the concept of top monotonicity as a common generalization of single-peakedness and single-crossingness (and of several other domain restrictions).

Forbidden substructures. Sometimes mathematical structures allow characterizations through forbidden substructures. For example, Kuratowski's theorem [Kur30] characterizes planar graphs in terms of forbidden subgraphs: a graph is planar if and only if it does not contain a subdivision of K_5 or $K_{3,3}$. In a similar spirit, Lekkerkerker and Boland [LB62] characterized interval graphs through five (infinite) families of forbidden induced subgraphs, and Földes and Hammer [FH77] characterized split graphs in terms of three forbidden induced subgraphs. Hoffman, Kolen, and Sakarovitch [HKS85] characterized totally-balanced 0-1-matrices in terms of certain forbidden submatrices. The characterizations of split graphs and totally-balanced 0-1-matrices use a finite number of obstructions, while the characterizations of planar graphs and interval graphs both involve infinitely many obstructions.

In the area of social choice, Ballester and Haeringer [BH11] characterize single-peaked preference profiles in terms of two forbidden substructures. The first forbidden substructure consists of three voters and three alternatives, where no two voters rank the same alternative worst. The second forbidden substructure consists of two voters and four alternatives, where (informally speaking) both voters rank the first three alternatives in opposite ways with the second alternative in the middle, but prefer the fourth alternative to the second one. We refer the interested reader to Chapters 4 and 5 for concrete definitions of these two substructures.

3.2 Results

Inspired by the approach and the results of Ballester and Haeringer [BH11], we present a characterization of single-crossing preference profiles by two forbidden substructures. One of our forbidden substructures consists of three voters and six alternatives (as described in Example 3.4) and the other one consists of four voters and four alternatives (as described in Example 3.5). We point out that the six (resp. four) alternatives in the first (resp. second) forbidden substructure are not necessarily distinct: The substructures only partially specify the preference orders of the involved voters; hence by identifying and collapsing some of the alternatives involved we can easily generate a number of smaller forbidden substructures (which of course are just special cases of our larger forbidden substructures). Finally, we will discuss the close relationship between single-crossing preference profiles and *consecutive ones matrices*. A 0-1-matrix has the consecutive ones property if its columns can be permuted such that the 1-values in each row are consecutive.

3.3 Chapter outline

In Section 3.4 we summarize basic definitions and provide some examples. In Section 3.5 we formulate and prove our main result (Theorem 3.2). In Section 3.6 we discuss the tightness of our characterization, and we argue that there is no characterization that works with smaller forbidden substructures. In Section 3.7 we briefly discuss some approaches to finding a single-crossing order of the voters in polynomial time. We conclude in Section 3.8.

3.4 Definitions, notations, and examples

Consider a preference profile with complete preference orders. An unordered pair of two distinct alternatives is called a *couple*. A subset $V \subseteq \mathcal{V}(\mathscr{P})$ of the voters is *mixed* with respect to couple $\{a, b\}$ if V contains two voters, one preferring a to b, and the other one preferring b to a. If V is not mixed with respect to $\{a, b\}$, then it is said to be *pure* with respect to $\{a, b\}$. Hence, an empty set of voters is pure with respect to every possible couple. A couple $\{a, b\}$ *separates* two sets V_1 and V_2 of voters from each other if no voter in V_1 agrees with any voter in V_2 on the relative ordering of a and b; in other words, sets V_1 and V_2 must both be pure with respect to $\{a, b\}$, and if both are non-empty then their union $V_1 \cup V_2$ is mixed.

Definition 3.1 (Single-crossing profiles)**.** A linear order of the voters is *single-crossing with respect to couple* $\{a, b\}$ if it can be split into an *initial piece* and a *final piece* that are separated by $\{a, b\}$. An order of the voters is *single-crossing* if it is single-crossing with respect to every possible couple. Finally, a preference profile is *single-crossing* if there is a single-crossing order L of the voters. We call a profile *maximally single-crossing* if this profile is single-crossing, and adding another voter with any preference order not yet in the profile makes it non-single-crossing.

v_1	v_2	v_3	v_4	v_5	v_6	v_7
1	1	3	3	3	3	4
2	3	1	2	2	4	3
3	2	2	1	4	2	2
4	4	4	4	1	1	1

Figure 3.1.: A single-crossing profile with seven voters and four alternatives. The gray shadow lines illustrate that the order of the voters from left to right is single-crossing with respect to the pair $\{1,4\}$ of alternatives.

Example 3.1. Let us take a look at the 7×4 profile given in Figure 3.1. The preference orders are written from top to bottom. For instance, voter v_1's preference order is $1 \succ 2 \succ 3 \succ 4$. This profile is single-crossing, and a single-crossing order of the voters is $\langle v_1, v_2, \ldots, v_7 \rangle$ as already shown in the figure (from left to right). For instance, couple $\{1,4\}$ separates this order (see the shadow lines) into two parts: the voters up to and including v_4 prefer 1 to 4 while the voters from v_5 onwards prefer 4 to 1. In Section 3.4.1 we show that this single-crossing profile is maximal.

It is easy to see that single-crossingness is a monotone property of preference profiles:

Observation 3.1. *Let $\mathcal{P}$ be a preference profile, and let $\mathcal{P}'$ result from $\mathcal{P}$ by removing some alternatives and/or voters. If $\mathcal{P}$ is single-crossing, then $\mathcal{P}'$ is also single-crossing.*

In the remaining part of this section we present several instructive examples of preference profiles which are single-crossing (Section 3.4.1), and some which are not (Section 3.4.2).

3.4.1 Profiles from weak Bruhat orders

Let S_m denote the set of permutations of the integers $1, 2, \ldots, m$ (note that a permutation is also a linear order; we use the term permutation instead of linear order in order to avoid confusion with Bruhat orders). We specify permutations $\pi \in S_m$ by listing the entries as $\pi = \langle \pi(1), \pi(2), \ldots, \pi(m) \rangle$. The *identity* permutation $\langle 1, 2, \ldots, m \rangle$ arranges the integers in increasing order. The *order reversing* permutation $\langle m, m-1, \ldots, 2, 1 \rangle$ arranges them in decreasing order. An *inversion* in π is a pair $(\pi(i), \pi(j))$ of two entries with $i < j$ and $\pi(i) > \pi(j)$ and a *descent* in π is a pair $(\pi(i), \pi(i+1))$ of consecutive entries with $\pi(i) > \pi(i+1)$. By definition, a descent $(\pi(i), \pi(i+1))$ is also an inversion. For instance, the inversions in the permutation $\langle 2, 3, 1 \rangle$ are $(2,1)$ and $(3,1)$, but only $(3,1)$ is a descent. We write $\pi \lhd \rho$ if permutation π can be obtained from permutation ρ by a series of swaps, each swapping two elements of a descent.

The partially ordered set $(S_m, \lhd)$ is known as a *weak Bruhat order*; see for instance [Bon04]. The weak Bruhat order has the identity permutation as the minimum element and the order reversing permutation as the maximum element. Every maximal chain (that is, every maximal subset of pairwise comparable permutations) in the weak Bruhat order has length $\frac{1}{2}m \cdot (m-1) + 1$ and contains the identity permutation and the order reversing permutation.

Example 3.2 illustrates the connection between weak Bruhat orders and single-crossing preference profiles; we refer to the works of Abello [Abe91] and Galambos and Reiner [GR08a] for more information.

Example 3.2. Let $C = (\pi_1 \lhd \pi_2 \lhd \cdots \lhd \pi_n)$ be a maximal chain (that is, a maximal totally ordered subset) with $n = \frac{1}{2}m \cdot (m-1) + 1$ permutations in the weak Bruhat order $(S_m, \lhd)$. We construct a profile by using $1, 2, \ldots, m$ as alternatives, and by interpreting every permutation π as preference order $\pi(1) \succ \pi(2) \succ \ldots \succ \pi(n)$ over the alternatives. Voter v_i has preference order π_i. See Figure 3.1 for an illustration with $m = 4$ alternatives and $n = 7$ voters (in a vertical fashion).

The resulting profile is single-crossing: each two alternatives a and b start off in the right order in the identity permutation π_1, eventually are swapped into the wrong order, and then can never be swapped back again at later steps. Furthermore, the profile contains $n = \frac{1}{2}m \cdot (m-1) + 1$ voters with pairwise distinct preference orders.

If we start the construction in Example 3.2 from arbitrary (not necessarily maximal) chains in the weak Bruhat order, then we can generate every possible single-crossing preference profile (up to isomorphism). This is another well-known connection, which follows from the fact that $\pi \lhd \rho$ if and only if every inversion of permutation π also is an inversion of permutation ρ [GR08a].

3.4.2 Some profiles that are not single-crossing

We provide three examples of profiles that are not single-crossing. The first example is due to the work of Saporiti and Tohmé [ST06] and shows a profile that is single-peaked but not single-crossing. The other two examples introduce two principal concepts of this thesis.

Example 3.3. Consider four alternatives $1, 2, 3, 4$ and three voters v_1, v_2, v_3 with the following preference orders:

$$\begin{aligned}
&\text{voter } v_1: \ 2 \succ_1 3 \succ_1 4 \succ_1 1,\\
&\text{voter } v_2: \ 4 \succ_2 3 \succ_2 2 \succ_2 1,\\
&\text{voter } v_3: \ 3 \succ_3 2 \succ_3 1 \succ_3 4.
\end{aligned}$$

Below we will see that this profile is not single-crossing but single-peaked (with respect to the order $\langle 1, 2, 3, 4 \rangle$ of alternatives, for instance).

From now on, when we speak of *configurations*, we mean some forbidden profiles we will use for the characterization. An $n \times m$ configuration is a profile with n voters and m alternatives.

Example 3.4 (γ-Configuration)**.**
A profile with three voters v_1, v_2, v_3 and six (not necessarily distinct) alternatives a, b, c, d, e, f is a γ-configuration if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \ b \succ_1 a \ \text{ and } \ c \succ_1 d \ \text{ and } \ e \succ_1 f,\\
&\text{voter } v_2: \ a \succ_2 b \ \text{ and } \ d \succ_2 c \ \text{ and } \ e \succ_2 f,\\
&\text{voter } v_3: \ a \succ_3 b \ \text{ and } \ c \succ_3 d \ \text{ and } \ f \succ_3 e.
\end{aligned}$$

The γ-configuration represents a situation where each voter disagrees with the other two voters on the order of exactly two distinct alternatives. The profile is not single-crossing as none of the three voters can be put between the other two: couple $\{a, b\}$ prevents us from putting v_1 into the middle, couple $\{c, d\}$ prevents voter v_2 in the middle, and couple $\{e, f\}$ prevents v_3 in the middle.

Example 3.4 provides an easy proof that the profile in Example 3.3 is not single-crossing, as this profile contains a γ-configuration with $a = 3$, $b = c = 2$, $d = e = 4$, and $f = 1$.

Example 3.5 (δ-Configuration)**.**
A profile with four voters v_1, v_2, v_3, v_4 and four (not necessarily distinct) alternatives a, b, c, d is a δ-configuration if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \ a \succ_1 b \ \text{ and } \ c \succ_1 d,\\
&\text{voter } v_2: \ a \succ_2 b \ \text{ and } \ d \succ_2 c,\\
&\text{voter } v_3: \ b \succ_3 a \ \text{ and } \ c \succ_3 d,\\
&\text{voter } v_4: \ b \succ_4 a \ \text{ and } \ d \succ_4 c.
\end{aligned}$$

The δ-configuration shows a different kind of voter behavior: Two voters disagree with the other two voters on the order of two alternatives, but also disagree between each other on the order of two further alternatives. This profile is not single-crossing, as couple $\{a, b\}$ forces us to place v_1 and v_2 next to each other, and to put v_3 and v_4 next to each other; couple $\{c, d\}$ forces us to place v_1 and v_3 next to each other, and to put v_2 and v_4 next to each other. This means that no voter can be placed in the first position.

3.5 A characterization through forbidden configurations

Examples 3.4 and 3.5 demonstrate that preference profiles that contain a γ-configuration or a δ-configuration cannot be single-crossing. It turns out that these two configurations are the only obstructions for the single-crossing property.

Theorem 3.2. *A preference profile $\mathscr{P}$ is single-crossing if and only if $\mathscr{P}$ contains neither a γ-configuration nor a δ-configuration.*

Proof. The (only if) part immediately follows from the monotonicity of the single-crossing property (Observation 3.1) and from the observations stated in Examples 3.4 and 3.5.

For the (if) part, we first introduce some additional definitions and notations. An *ordered partition* $\langle X_1, \ldots, X_p \rangle$ of the voters $v_1, \ldots, v_n$ satisfies the following properties: every part X_i is a non-empty set, distinct parts are disjoint, the union of all parts is the set of all voters, and the arrangement of the parts X_i in the ordered partition is crucial. The *trivial* ordered partition has $p = 1$ and hence consists of a single part $\{v_1, \ldots, v_n\}$. We let $\{a_k, b_k\}$ with $1 \leq k \leq \frac{1}{2} m \cdot (m-1)$ be an enumeration of all possible couples, and we define $\mathscr{C}_k$ as the set containing the first k couples in this enumeration.

Now let us prove the (if) part of the theorem. We consider some arbitrary preference profile $\mathscr{P}$ that contains neither γ-configurations nor δ-configurations. Our argument is algorithmic in nature. We start from the trivial partition $\mathscr{X}^{(0)}$ of the voters, and then refine this partition step by step while working through $\frac{1}{2} m \cdot (m-1)$ phases. The k^{th} such phase generates an ordered partition $\mathscr{X}^{(k)} = \langle X_1^{(k)}, \ldots, X_p^{(k)} \rangle$ of the voters that satisfies the following two properties.

(i) For each index j with $1 \leq j \leq p-1$, the union of parts $X_1^{(k)}, \ldots, X_j^{(k)}$ is separated from the union of parts $X_{j+1}^{(k)}, \ldots, X_p^{(k)}$ by one of the couples in $\mathscr{C}_k$.

(ii) For each couple in $\mathscr{C}_k$, there is an index j with $1 \leq j \leq p-1$ such that the couple separates the union of $X_1^{(k)}, \ldots, X_j^{(k)}$ from the union of $X_{j+1}^{(k)}, \ldots, X_p^{(k)}$.

Note that property (ii) implies that every part $X_j^{(k)}$ is pure with respect to every couple in $\mathscr{C}_k$.

The following four statements summarize some useful combinatorial observations on the ordered partition $\mathscr{X}^{(k)}$ and how it relates to the couple $\{a_{k+1}, b_{k+1}\}$.

Claim 3.3. *At most one part in the ordered partition $\mathscr{X}^{(k)}$ is mixed with respect to couple $\{a_{k+1}, b_{k+1}\}$.*

Proof of Claim 3.3. Suppose for the sake of contradiction that the parts $X_s^{(k)}$ and $X_t^{(k)}$ with $1 \le s < t \le p$ both are mixed with respect to couple $\{a_{k+1}, b_{k+1}\}$. In other words, part $X_s^{(k)}$ contains a voter v_1' with $a_{k+1} \succ b_{k+1}$ and another voter v_2' with $b_{k+1} \succ a_{k+1}$, and part $X_t^{(k)}$ contains a voter v_3' with $a_{k+1} \succ b_{k+1}$ and another voter v_4' with $b_{k+1} \succ a_{k+1}$.

Property (i) yields the existence of a couple $\{x, y\} \in \mathscr{C}_k$ that separates the union of parts $X_1^{(k)}, \ldots, X_s^{(k)}$ from the union of the parts $X_{s+1}^{(k)}, \ldots, X_p^{(k)}$. In particular, this couple separates $X_s^{(k)}$ from $X_t^{(k)}$. This implies that voters v_1' and v_2' agree on the order of couple $\{x, y\}$ (say, with $x \succ y$), whereas voters v_3' and v_4' have the opposite ordering (say $y \succ x$). Then the four voters v_1', v_2', v_3', and v_4' together with the four alternatives a_{k+1}, b_{k+1}, x, and y form a δ-configuration; this yields the desired contradiction. (of Claim 3.3) ⋄

While the last claim is about the "mixed" property of one part, the next two claims are about the "mixed" property across three parts.

Claim 3.4. *Consider two indices s and t with $2 \le s < t \le p$. If some voter v_1' in part $X_1^{(k)}$ ranks $a_{k+1} \succ b_{k+1}$ and if some voter v_2' in part $X_s^{(k)}$ ranks $b_{k+1} \succ a_{k+1}$, then every voter v_3' in part $X_t^{(k)}$ ranks $b_{k+1} \succ a_{k+1}$.*

Proof of Claim 3.4. Suppose for the sake of contradiction that there is a voter v_3' from part $X_t^{(k)}$ that ranks $a_{k+1} \succ b_{k+1}$. Then the couple $\{a_{k+1}, b_{k+1}\}$ separates $\{v_2'\}$ from $\{v_1', v_3'\}$. Property (i) yields a couple $\{x, y\} \in \mathscr{C}_k$ that separates $X_1^{(k)}$ from the union set $X_s^{(k)} \cup X_t^{(k)}$; this couple separates $\{v_1'\}$ from $\{v_2', v_3'\}$. Property (i) also yields a couple $\{x', y'\} \in \mathscr{C}_k$ that separates $X_t^{(k)}$ from the union set $X_1^{(k)} \cup X_s^{(k)}$; this couple separates v_3' from $\{v_1', v_2'\}$.

Then the three voters v_1', v_2', and v_3' together with the six alternatives a_{k+1}, b_{k+1}, x, y, x', and y' form a γ-configuration; a contradiction. (of Claim 3.4) ⋄

The statement of the following claim is symmetric to the statement of Claim 3.4, and it can be proven by symmetric arguments.

Claim 3.5. *Consider two indices s and t with $1 \le s < t \le p-1$. If some voter v_2' in part $X_t^{(k)}$ ranks $a_{k+1} \succ b_{k+1}$ and some voter v_3' in part $X_p^{(k)}$ ranks $b_{k+1} \succ a_{k+1}$, then every voter v_1' in part $X_s^{(k)}$ ranks $a_{k+1} \succ b_{k+1}$.*

The next statement shows the existence of a "separation point" for the couple $\{a_{k+1}, b_{k+1}\}$.

Claim 3.6. *There exists an index ℓ with $1 \le \ell \le p$ such that the couple $\{a_{k+1}, b_{k+1}\}$ separates the union of parts $X_1^{(k)}, \dots, X_{\ell-1}^{(k)}$ from the union of parts $X_{\ell+1}^{(k)}, \dots, X_p^{(k)}$.*

Proof of Claim 3.6. If $p = 1$ or if all voters in the profile agree on the relative order of a_{k+1} and b_{k+1}, then $\ell = 1$. Hence we assume that $p \ge 2$ and that there are two voters who disagree on the relative order of a_{k+1} and b_{k+1}. By Claim 3.3 the parts $X_1^{(k)}$ and $X_p^{(k)}$ cannot both be mixed with respect to $\{a_{k+1}, b_{k+1}\}$.

If the first part $X_1^{(k)}$ is pure with respect to $\{a_{k+1}, b_{k+1}\}$, we pick an arbitrary voter v_1' from $X_1^{(k)}$. We choose ℓ as the smallest index for which $X_\ell^{(k)}$ contains some voter v_2' who ranks a_{k+1} versus b_{k+1} differently from voter v_1'. Then Claim 3.4 implies that every voter v_3' in the parts $X_{\ell+1}^{(k)}, \dots, X_p^{(k)}$ must rank a_{k+1} versus b_{k+1} differently from voter v_1'. Hence the chosen index ℓ has all desired properties, and this case is closed. In the remaining case the last part $X_p^{(k)}$ is pure with respect to $\{a_{k+1}, b_{k+1}\}$; this case can be settled in the same fashion using Claim 3.5. (of Claim 3.6) $\diamond$

Now let us finally describe how to construct the ordered partition $\mathscr{X}^{(k+1)}$ in the $(k+1)^{\text{st}}$ phase. Our starting point is the ordered partition $\mathscr{X}^{(k)}$, and we determine an index ℓ as defined in Claim 3.6. If part $X_\ell^{(k)}$ is pure with respect to $\{a_{k+1}, b_{k+1}\}$, then we make the new partition $\mathscr{X}^{(k+1)}$ coincide with the old partition $\mathscr{X}^{(k)}$; Properties (i) and (ii) are satisfied in $\mathscr{X}^{(k+1)}$. If part $X_\ell^{(k)}$ is mixed with respect to $\{a_{k+1}, b_{k+1}\}$, then we subdivide it into two parts Y and Z so that $\{a_{k+1}, b_{k+1}\}$ separates the union of parts $X_1^{(k)}, \dots, X_{\ell-1}^{(k)}, Y$ from the union of parts $Z, X_{\ell+1}^{(k)}, \dots, X_p^{(k)}$. Then the resulting partition

$$\mathscr{X}^{(k+1)} = \langle X_1^{(k)}, \dots, X_{\ell-1}^{(k)}, Y, Z, X_{\ell+1}^{(k)}, \dots, X_p^{(k)} \rangle$$

satisfies Properties (i) and (ii) by construction.

We keep working like this and complete phase after phase. After the very last phase $k = \frac{1}{2} m \cdot (m-1)$ we generate the final ordered partition $\mathscr{X}^* = \langle X_1^*, \dots, X_q^* \rangle$. We construct an order π^* of the voters that lists the voters in every part X_j^* before all the voters in part X_{j+1}^* $(1 \le j \le q-1)$. Property (ii) guarantees that every couple separates an initial piece of partition $\mathscr{X}^*$ from the complementary final piece, which implies that the order π^* for the voters in $\mathscr{P}$ is single-crossing. This completes the proof of Theorem 3.2. $\square$

We conclude this section with two comments on this proof; they deal with the uniqueness of single-crossing orders. If $\mathscr{P}$ is a single-crossing profile where all voters have distinct preference orders, then there are exactly two single-crossing orders of the voters and they are mirror images of each other. This follows directly from the last part of the proof of Theorem 3.2. Note that in contrast to the single-crossing property, a single-peaked profile may admit more than two single-peaked orders.

By Property (i), every two consecutive parts X_j^* and X_{j+1}^* must be separated by one of the couples. Since there are $\frac{1}{2}m \cdot (m-1)$ distinct couples, there are at most $\frac{1}{2}m \cdot (m-1) + 1$ parts in the final partition. This implies that in a single-crossing preference profile the number of distinct preference orders of the voters is at most $\frac{1}{2}m \cdot (m-1) + 1$. Of course, this bound is already known from the connection between single-crossing profiles and weak Bruhat orders as indicated in Section 3.4.1.

3.6 The size of forbidden configurations

Recall that we denote a profile with n voters and m alternatives as an $n \times m$ configuration. Theorem 3.2 characterizes single-crossing preference profiles through certain forbidden 3×6 and 4×4 configurations. Are there perhaps other characterizations that work with smaller forbidden configurations? The following lemma shows that this is not the case, and hence our characterization uses the smallest possible forbidden configurations.

Lemma 3.7. *Every characterization of single-crossing preference profiles through forbidden configurations must forbid*

(i) some $n \times m$ configuration with $n \geq 3$ and $m \geq 6$ and

(ii) some $n \times m$ configuration with $n \geq 4$ and $m \geq 4$.

Proof. Consider an arbitrary characterization of single-crossing profiles with forbidden configurations $F_1, \ldots, F_k$. Consider the following 3×6 configuration C:

$$\begin{aligned}
&\text{voter } v_1: \ b \succ_1 a \succ_1 c \succ_1 d \succ_1 e \succ_1 f,\\
&\text{voter } v_2: \ a \succ_2 b \succ_2 d \succ_2 c \succ_2 e \succ_2 f.\\
&\text{voter } v_3: \ a \succ_3 b \succ_3 c \succ_3 d \succ_3 f \succ_3 e.
\end{aligned}$$

This profile contains a γ-configuration and thus is not single-crossing. If we remove any alternative from C, then the resulting 3×5 configuration is single-crossing and cannot be forbidden. If we remove any voter from C, then the resulting 2×6 configuration is also single-crossing and cannot be forbidden. Hence, the only possibility for correctly recognizing C as not single-crossing is by either forbidding C itself or by forbidding appropriate larger configurations that contain C. This proves (i). The proof of (ii) is based on the following 4×4 configuration C' which contains a δ-configuration:

voter v_1: $a \succ_1 b \succ_1 c \succ_1 d$,
voter v_2: $a \succ_2 b \succ_2 d \succ_2 c$,
voter v_3: $b \succ_3 a \succ_3 c \succ_3 d$,
voter v_4: $b \succ_4 a \succ_4 d \succ_4 c$.

Since the argument closely follows the one in (i), we omit the details. □

3.7 Recognizing the single-crossing property

As we already mentioned, the arguments in Section 3.5 implicitly describe an $O(n \cdot m^2)$ algorithm which decides whether a given voter profile is single-crossing and, if so, computes a single-crossing order; recall that n denotes the number of voters and m denotes the number of alternatives. To show an extremely simple connection between single-crossing orderings and the so-called consecutive ones matrix property, we sketch an alternative way of recognizing single-crossing profiles by utilizing the PQ-tree algorithm of Booth and Lueker [BL76]. The PQ-tree algorithm was designed to recognize, for instance, *consecutive ones* 0-1*-matrices*. A 0-1-matrix has the *consecutive ones property* if its columns can be permuted so that the ones in each row are consecutive (and hence form an interval).

Given an arbitrary $n \times m$ preference profile $\mathcal{P}$ where no two voters have the same preference orders, and where n denotes the number of voters and m the number alternatives, we transform it into a 0-1-matrix $M(\mathcal{P})$ with $m \cdot (m-1)$ rows and n columns in the following way. For each voter, the matrix $M(\mathcal{P})$ contains a corresponding column. For each ordered pair $\langle a, b \rangle$ of alternatives, matrix $M(\mathcal{P})$ has a corresponding row with value 1 at column j if voter j ranks $a \succ b$, and value 0 otherwise.

Example 3.6. Consider a single-crossing profile with four voters and three alternatives as depicted in Figure 3.2a. We can construct a corresponding 0-1-matrix $M(\mathcal{P})$ for this profile which is depicted in Figure 3.2b. By applying the PQ-tree algorithm by Booth and Lueker [BL76], we can find all permutations of the columns with the consecutive ones property. One possible consecutive ones permutation of the columns is $\langle v_1, v_4, v_2, v_3 \rangle$. As we can easily verify, this is also a single-crossing order of the voters in the original profile.

We show the close relation between the single-crossing property and the consecutive ones property.

Lemma 3.8. *A preference profile $\mathcal{P}$ is single-crossing if and only if the corresponding* 0-1*-matrix $M(\mathcal{P})$ has the consecutive ones property.*

voter v_1: $3 \succ_1 1 \succ_1 2$,
voter v_2: $2 \succ_2 3 \succ_2 1$,
voter v_3: $2 \succ_3 1 \succ_3 3$,
voter v_4: $3 \succ_4 2 \succ_4 1$.

(a) A 4×3 preference profile

	v_1	v_2	v_3	v_4
$\langle 1,2\rangle$	1	0	0	0
$\langle 2,1\rangle$	0	1	1	1
$\langle 1,3\rangle$	0	0	1	0
$\langle 3,1\rangle$	1	1	0	1
$\langle 2,3\rangle$	0	1	1	0
$\langle 3,2\rangle$	1	0	0	1

(b) The corresponding 0-1 matrix

Figure 3.2.: A preference profile with four voters and three alternatives and its 0-1 matrix representation with six rows and four columns.

Proof. For the "only if" part, suppose that $\mathscr{P}$ is single-crossing and let L be a single-crossing order of the voters. We permute the columns of $M(\mathscr{P})$ according to the order L to obtain M' and we show that in each row of M' all ones are consecutive. Consider an arbitrary ordered pair $\langle a,b\rangle$ of alternatives. Then, by the definition of single-crossingness, it follows that $\langle a,b\rangle$ separates L into two pieces L_1 and L_2. This means that, by the definition of $M(\mathscr{P})$ and M', all ones in row $\langle a,b\rangle$ of M' are either in L_1 or in L_2.

For the "if" part, suppose that the columns in $M(\mathscr{P})$ can be permuted to obtain matrix M' so that the ones in each row of M' are consecutive. Let L be the order of the voters that corresponds to the arrangement of the columns (from left to right) in M'. We show that L is single-crossing. Suppose for the sake of contradiction that a pair $\{a,b\}$ of the alternatives is not single-crossing with respect to L. This means that there are three voters v_i, v_j, v_k with $v_i \succ_L v_j \succ_L v_k$ such that v_i and v_k rank $a \succ b$, but v_j ranks $b \succ a$. This implies that the ones in row $\langle a,b\rangle$ of matrix M' are not consecutive as the row vector restricted to the corresponding three columns is $\langle 1,0,1\rangle$; a contradiction. □

Since deciding whether a matrix has the consecutive ones property can be done in polynomial time [BL76], we can derive the following.

Theorem 3.9. *Finding a single-crossing for a profile with n voters and m alternatives can be done in $O(n \cdot m^2)$ time.*

Proof. The PQ-algorithm by [BL76] solves the consecutive ones matrix problem in $O(x+y+z)$ time, where x is the number of columns, y is the number of rows, and

z is the total number of 1s in the matrix. Hence, single-crossing profiles can be recognized in $O(m^2 + n + n \cdot m^2) = O(n \cdot m^2)$ time. □

We conclude this section by remarking that, first, the PQ-tree algorithm is not only useful for deciding the single-crossing property. For instance, Bartholdi III and Trick [BT86] used it to decide the single-peaked property. Second, Doignon and Falmagne [DF94] and Elkind, Faliszewski, and Slinko [EFS12] also presented polynomial-time algorithms to recognize the single-crossing property, avoiding the usage of PQ-tree algorithms. The running time of the algorithm by Doignon and Falmagne is $O(n^2 + n \cdot m \cdot \log(m))$. Elkind, Faliszewski, and Slinko do not provide an analysis of the running time of their algorithm, but a straightforward implementation of their algorithm runs in $O(n^3 \cdot m^2)$.

3.8 Concluding remarks

Following the research line of Ballester and Haeringer [BH11] who characterized single-peaked profiles by two finite forbidden substructures, we likewise characterized single-crossing profiles by two finite forbidden substructures. We also presented an efficient algorithm to recognize the single-crossing property.

For profiles which are not single-peaked or single-crossing, we may be interested in approximating these properties by deleting the fewest possible number of voters or alternatives. The resulting computational questions are the focus of Chapter 5. There, we model the concept of nearly nicely structured profiles and study the computational complexity of finding such profiles.

CHAPTER 4

One-Dimensional Euclidean Preferences

If a seller increases his price too far he will gradually lose business to his rivals, but he does not lose all his trade instantly when he raises his price only a trifle. Many customers will still prefer to trade with him because they live nearer to his store than to the others, ...

Harold Hotelling, 1929

We show that one-dimensional Euclidean preference profiles cannot be characterized in terms of finitely many forbidden substructures. This result is in strong contrast to the case of single-peaked and single-crossing preference profiles, for which such finite characterizations have been derived (see the work of Ballester and Haeringer [BH11] and Theorem 3.2 in Chapter 3).

4.1 Introduction

Single-peakedness, single-crossingness, and one-dimensional Euclideanness are popular domain restrictions that show up in a variety of models in the social sciences and in economics [Bla48, Hot29, Rob77]. In many situations, these domain restrictions guarantee the existence of a desirable entity that would not exist without the restriction, as for instance a strategy-proof voting rule or a weak Condorcet winner.

As already discussed in Section 3.1, *single-peakedness* describes a natural order of the alternatives, while *single-crossingness* describes a natural order of the voters. In this chapter, we study a third domain restriction, *one-dimensional Euclidean preferences*, that describes an embedding of voters and alternatives into the real numbers, such that every voter prefers alternatives that are embedded close to him to alternatives that are embedded farther away. For three alternatives, a single-crossing profile with the largest voter set where no two voters have the same preference order

This chapter is based on "The one-dimensional Euclidean domain: Finitely many obstructions are not enough" by J. Chen, K. Pruhs, and G. J. Woeginger, *Social Choice and Welfare* [CPW16] .

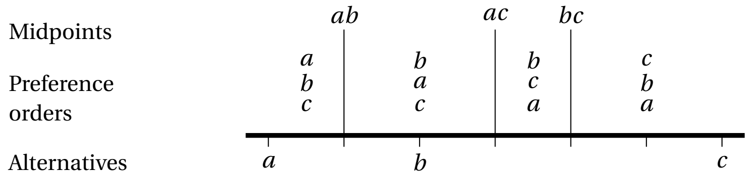


Figure 4.1.: A one-dimensional Euclidean representation of three alternatives: a, b, and c, and four voters with pairwise different preference orders: $a \succ b \succ c$, $b \succ a \succ c$, $b \succ c \succ a$, and $c \succ b \succ a$. The alternatives are embedded from left to right into the real number line so that the midpoints of each pair of the alternatives divide the real line into four intervals. Each of these four intervals can contain at most one of these four voters. For instance, the voter with preference order $b \succ a \succ c$ (read from top to bottom) prefers b to a, implying that he must be embedded at a point to the right of the midpoint of a and b. He also prefers a to c, implying that he must be embedded at a point to the left of the midpoint of a and c. Thus, he can be embedded at any point in the open interval between the midpoint of a and b, and the midpoint of a and c.

consists of four voters. Indeed, this coincides with the size of the maximum one-dimensional Euclidean profile, that is, a one-dimensional Euclidean profile with the largest voter set such that no two voters have the same preference order.

Figure 4.1 illustrates such a situation. Three alternatives, a, b, c, and four voters v_1, v_2, v_3, v_4 with preference orders

$$\begin{aligned}
\text{voter } v_1&: \ a \succ b \succ c,\\
\text{voter } v_2&: \ b \succ a \succ c,\\
\text{voter } v_3&: \ b \succ c \succ a,\\
\text{voter } v_4&: \ c \succ b \succ a,
\end{aligned}$$

are embedded into the real number line so that each voter prefers one alternative x to another y if his distance to x is shorter than to y.

One-dimensional Euclidean preferences go back to Hotelling [Hot29]. They have been discussed by Coombs [Coo64] under the name *unidimensional unfolding* representations, and they are restrictions of single-peaked and single-crossing preferences. Doignon and Falmagne [DF94] discussed Euclidean preferences in the context of behavioral sciences, and Brams, Jones, and Kilgour [BJK02] discussed them in the context of coalition formation.

Forbidden substructures. As already discussed in the previous chapter, mathematical structures allow characterizations through forbidden substructures. There, we have seen that single-crossing preferences can be characterized by finitely many forbidden substructures. Let us stress that every *monotone* property of profiles (that is, every property that is preserved under the removal of voters and/or alternatives) can be characterized by a set of forbidden substructures. For many monotone properties, however, the number of forbidden substructures may be infinite. The Condorcet principle, that is, the existence of a Condorcet winner, is a typical *non-monotone* example of a property in Social Choice that *cannot* be characterized at all through forbidden substructures because after the deletion of one or more voters, a Condorcet winner may no longer exist.

A characterization by finitely many forbidden substructures has many positive consequences. Whenever a family $\mathscr{F}$ of combinatorial objects allows such a finite characterization, this directly implies the existence of a polynomial time algorithm for recognizing the members of $\mathscr{F}$: one may simply work through the forbidden substructures one by one, and check whether the object in question contains the substructure. By looking deeper into the combinatorial structure of such families $\mathscr{F}$, one usually manages to find recognition algorithms that are much faster than this simple approach. As an example, there are algorithms for recognizing single-peaked preference profiles that are due to Bartholdi III and Trick [BT86], Doignon and Falmagne [DF94], and Escoffier, Lang, and Öztürk [ELÖ08]. Single-crossingness can also be recognized very efficiently; see Doignon and Falmagne [DF94], Elkind, Faliszewski, and Slinko [EFS12], and Bredereck, Chen, and Woeginger [BCW13a] and Chapter 3.

As another positive consequence, a characterization by finitely many forbidden substructures often helps in understanding the algorithmic and combinatorial behavior of family $\mathscr{F}$. For example, in the next chapter, we will investigate the problem of deciding whether a given preference profile is close to having a nice structure. The distance is measured by the number of voters or alternatives that have to be deleted from the given profile in order to reach a nicely structured profile. For the cases where 'nicely structured' means single-peaked or single-crossing, our proofs are heavily based on characterizations ([BH11] and Chapter 3) by finitely many forbidden substructures. We will see more results on this issue in Chapter 5. Elkind and Lackner [EL14] studied similar questions and derive approximation algorithms for the number of deleted voters or alternatives. All of their results are centered around preference profiles that can be characterized by a finite number of forbidden substructures, and some of the theorems are parameterized by the number of forbidden substructures.

4.2 Results

As one-dimensional Euclidean profiles form a special case of single-peaked and single-crossing profiles (see Section 4.4 for more information on this, and see Section 3.4 for more information on single-crossing preferences), every forbidden substructure to single-peakedness and every forbidden substructure to single-crossingness will automatically also form a forbidden substructure to one-dimensional Euclideanness. Now the question arises: *"Are there any further forbidden substructures to one-dimensional Euclideanness?"* Coombs [Coo64] answered this back in 1964: *"Yes, there are!"* (see Section 4.4.3 for more information). This immediately takes us to another question: *"Is there a characterization of one-dimensional Euclideanness in terms of finitely many forbidden substructures?"* The answer to this second question is negative, as we are going to show in this chapter.

To this end, we construct an infinite sequence of preference profiles that have two crucial properties. First, none of these profiles is one-dimensional Euclidean. Secondly, every such profile just *barely* violates one-dimensional Euclideanness, as the deletion of an arbitrary voter immediately makes the profile one-dimensional Euclidean. The second property implies that each profile in the sequence is on the edge of being Euclidean, and that the reason for its non-Euclideanness must lie in its overall structure. In other words, each of these infinitely many profiles yields a separate forbidden substructure for one-dimensional Euclideanness, and this is exactly what we want to establish.

The definition of the infinite profile sequence and the resulting analysis are quite involved. Ironically, the complicatedness of our proof is a consequence of the very statement we are going to prove. As part of our proof, we have to argue that the deletion of an arbitrary voter from an arbitrary profile in the sequence yields a one-dimensional Euclidean profile. If there was a characterization of one-dimensional Euclideanness by finitely many forbidden substructures, then the following argument would be relatively easy to get through: we could simply analyze the preference profile and show that the deletion of any voter removes all forbidden substructures. But unfortunately, such a characterization does not exist. The only viable (and fairly tedious) approach is to explicitly specify the corresponding Euclidean representations (one representation per deleted voter!) and to prove by case distinctions that each such representation correctly encodes the preferences of all remaining voters.

4.3 Chapter outline

In Section 4.4, we review single-peakedness and single-crossingness, and introduce one-dimensional Euclideanness. We also refer the reader to the previous chapter, where we characterize single-crossing preference profiles. In Section 4.5 we formulate our main results in Theorems 4.3 and 4.4, and we show how Theorem 4.4 follows from Theorem 4.3. Sections 4.6 through 4.10 present the long and technical proof of Theorem 4.3. More precisely, Section 4.6 defines an infinite sequence of profiles which are shown *not* to be one-dimensional Euclidean in Section 4.7. Section 4.8 gives, for each voter, an embedding for the profile resulting from the deletion of this voter. Section 4.9 contains some important but very technical lemmas which are used in Section 4.10 to show the correctness of these embeddings.

We remark that to follow the main idea of this chapter, it is sufficient to read through Sections 4.5 to 4.7.

Section 4.11 shows how to construct other infinite sequences of non-one-dimensional Euclidean profiles. Section 4.12 completes the chapter with some discussion on future research work.

4.4 Definitions, notations, and examples

In this section, we introduce the three major concepts used in this chapter: single-peaked profiles, single-crossing profiles, and one-dimensional Euclidean profiles.

4.4.1 Single-peaked profiles

A linear order of the alternatives is *single-peaked* with respect to a fixed voter v, if the preferences of v taken along this order have a single local maximum, that is, the preferences along this order never increase again once decreased. A preference profile is *single-peaked*, if it allows an order L of the alternatives that is single-peaked with respect to every voter v. Formally, L is single-peaked with respect to v if for each three distinct alternatives $a, b, c \in A$ with $a \succ_L b \succ_L c$ it holds that

$$a \succ_L b \succ_L c \text{ implies that if } a \succ_v b, \text{ then } b \succ_v c.$$

Note that for every single-peaked permutation $\langle \pi(1), \pi(2), \ldots, \pi(m) \rangle$ of the alternatives, the reverse permutation $\langle \pi(m), \ldots, \pi(2), \pi(1) \rangle$ is also single-peaked. We refer the reader to Section 5.5.2 for more information on single-peakedness and its characterization.

4.4.2 Single-crossing profiles

The single-crossing property describes a linear order of the voters that can be *separated* by every (unordered) pair of alternatives. For further information, we refer to Chapter 3, where we formally define the single-crossing property and characterize it through two finite forbidden substructures introduced in Examples 3.4 and 3.5.

4.4.3 One-dimensional Euclidean profiles

The one-dimensional Euclidean property describes the situation where voters rank alternatives according to their spatial perception.

Definition 4.1 (One-dimensional Euclidean profiles)**.** Consider a common embedding (E, F) of the voters and alternatives into the real number line, which assigns to every alternative j a real number $E[j]$, and to every voter v_i a real number $F[i]$. A preference profile is *one-dimensional Euclidean* or simply *1-D Euclidean* (as we only consider the one-dimensional case) if there is a common embedding of the voters and alternatives such that for every voter v_i and for every pair a and b of alternatives, $a \succ_i b$ holds if and only if the distance from $F[i]$ to $E[a]$ is strictly smaller than the distance from $F[i]$ to $E[b]$, that is,

$$|F[i] - E[a]| < |F[i] - E[b]|.$$

In other words, small spatial distances from the point $F[i]$ indicate strong preferences of voter v_i. We subsequently also call this 1-D Euclidean embedding (E, F) a *1-D Euclidean representation.*

It is well-known (and easy to see) that every one-dimensional Euclidean profile is single-peaked and single-crossing: the left-to-right order of the alternatives along the 1-D Euclidean representation is single-peaked, and the left-to-right order of the voters along the 1-D Euclidean representation is single-crossing. Thus, a 1-D Euclidean preference profile can have at most $m \cdot (m-1)/2 + 1$ voters with different preference orders as a single-crossing profile can have at most $m \cdot (m-1)/2 + 1$ voters with different preference orders, where m denotes the number of alternatives (see the discussion at the end of Section 3.5). Figure 4.1 shows a 1-D Euclidean representation for a 1-D Euclidean preference profile with $m = 3$ alternatives and $3 \cdot 2/2 + 1 = 4$ voters with different preference orders.

Coombs [Coo64, page 91] discussed a preference profile with 16 voters and 6 alternatives which is both single-peaked and single-crossing, but fails to be one-dimensional Euclidean. The following example contains the smallest profile known to us that has these intriguing properties.[1]

Example 4.1. Consider the following 3×6 profile $\mathcal{P}$:

voter v_1: $c \succ b \succ a \succ d \succ e \succ f$,
voter v_2: $c \succ d \succ b \succ e \succ f \succ a$,
voter v_3: $e \succ d \succ c \succ f \succ b \succ a$.

[1] Indeed, Bulteau and Chen [BC] recently showed that every single-peaked and single-crossing profile with up to two voters or with up to five alternatives is 1-D Euclidean.

This profile $\mathscr{P}$ is single-peaked with respect to the order $\langle a, b, c, d, e, f\rangle$ of alternatives, and it is single-crossing with respect to the ordering $\langle v_1, v_2, v_3\rangle$ of voters. Furthermore, we will see below (Example 4.2) that $\mathscr{P}$ is not 1-D Euclidean, but deleting any single voter makes the profile 1-D Euclidean.

As the profile in Example 4.1 is single-peaked and single-crossing, it does not contain any of the forbidden substructures listed in Definitions 5.3 and 5.5 and in Examples 3.4 and 3.5. Hence, there must be some other forbidden substructure contained in it, that is responsible for its non-1-D Euclideanness. Example 4.1 and the 16×6 profile by Coombs provide first indications that the forbidden substructures for one-dimensional Euclideanness might be complex and intricate to analyze.

The following two propositions state simple observations that will be used repeatedly in our arguments.

Proposition 4.1. *Let (E, F) be a 1-D Euclidean representation of some profile and let a and b be two alternatives in the profile with $E[a] < E[b]$. Then, voter v_i prefers a to b if and only if $F[i] < \frac{1}{2}(E[a]+E[b])$, and he prefers b to a if and only if $F[i] > \frac{1}{2}(E[a]+E[b])$.*

Proof. By the definition of 1-D Euclideanness, voter v_i prefers a to b if and only if

$$|F[i]-E[a]| < |F[i]-E[b]| \tag{$\star$}$$

holds. Since $E[a] < E[b]$, ($\star$) holds if and only if $F[i] < \frac{1}{2}(E[a]+E[b])$. The case of v_i preferring b to a can be shown using symmetric reasoning. □

Proposition 4.1 describes how to embed the voters if we know the relative order of the alternatives in the embedding. For instance, in Figure 4.1, since $E(a) < E(b)$, a voter preferring $a \succ b$ must be embedded left to the midpoint of $E(a)$ and $E(b)$, that is, $\frac{1}{2}(E[a]+E[b])$. The next result is derived from the single-peaked property that every 1-D Euclidean profile must possess.

Proposition 4.2. *Let (E, F) be a 1-D Euclidean representation of some preference profile and let a, b, c be three alternatives in the profile with $E[a] < E[b] < E[c]$.*

- *If voter v_i prefers $a \succ_i b$, then he also prefers $b \succ_i c$.*
- *If voter v_i prefers $c \succ_i b$, then he also prefers $b \succ_i a$.*

Proof. By the definition of 1-D Euclideanness, the left-to-right order of the alternatives obtained from the embedding E is a single-peaked order. Then, the two statements follow immediately from the definition of single-peakedness. □

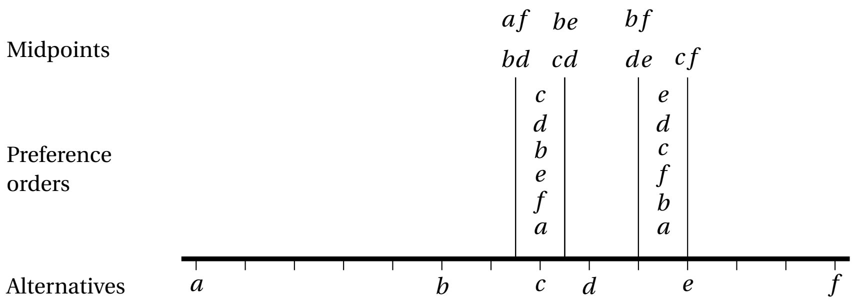


Figure 4.2.: An embedding showing that the profile from Example 4.1 with voters v_2 and v_3, with preference orders $c \succ d \succ b \succ e \succ f \succ a$ and $e \succ d \succ c \succ f \succ b \succ a$, is 1-D Euclidean. The embedding of alternatives $a, b, \ldots, f$ is depicted explicitly. Voter v_2 (resp. v_3) can be embedded at any point in the open interval between the midpoint of b and d, and the midpoint of c and d (resp. the midpoint of d and e, and the midpoint of c and f).

Example 4.2. We continue our discussion of Example 4.1. Suppose for the sake of contradiction that the profile in Example 4.1 is one-dimensional Euclidean. Let (E, F) be such a 1-D Euclidean representation. We can verify that the only single-peaked orders of the alternatives are $\langle a, b, c, d, e, f \rangle$ and its reverse. By Proposition 4.1 and by the fact that voter v_1 ranks $c \succ b$ and $a \succ d$, it follows that

$$E[b] + E[c] < 2F[1] < E[a] + E[d]. \tag{4.1}$$

Again, by Proposition 4.1 and by the fact that voter v_2 ranks $f \succ a$ and $b \succ e$, it follows that

$$E[a] + E[f] < 2F[2] < E[b] + E[e]. \tag{4.2}$$

Finally, by Proposition 4.1 and by the fact that voter v_3 ranks $e \succ d$ and $c \succ f$, it follows that

$$E[d] + E[e] < 2F[3] < E[c] + E[f]. \tag{4.3}$$

If we add up the inequalities (4.1)–(4.3), then we obtain the contradiction

$$\sum_{x=a}^{f} E[x] < \sum_{x=a}^{f} E[x].$$

As for the minimal non-1-D Euclideanness, Figure 4.2 shows a possible 1-D Euclidean representation of the profile without voter v_1. For the profile without voter v_2, we can verify that the following embedding is 1-D Euclidean:

$$E[a] = 0,\ E[b] = 1,\ E[c] = 3,\ E[d] = 6,\ E[e] = 7,\ E[f] = 12,$$
$$F[v_1] = 2.5,\ F[v_3] = 7.$$

For the profile without voter v_3, we can verify that the following embedding is 1-D Euclidean:

$$E[a] = 0,\ E[b] = 3,\ E[c] = 5,\ E[d] = 10,\ E[e] = 12,\ E[f] = 13,$$
$$F[v_1] = 4.5,\ F[v_3] = 7.$$

Finally, we mention that the (mathematical) literature on one-dimensional Euclidean preference profiles is scarce. Doignon and Falmagne [DF94], Knoblauch [Kno10], and Elkind and Faliszewski [EF14] designed polynomial-time algorithms for deciding whether a given preference profile has a one-dimensional Euclidean representation. Their approaches are not purely combinatorial, as they are partially based on linear programming formulations. In the first phase, while the approaches of Doignon and Falmagne [DF94] and of Elkind and Faliszewski [EF14] first construct a single-crossing order and then a single-peaked order, the approach of Knoblauch [Kno10] first constructs a single-peakd order and then a single-crossing order. In the second phase, they use linear programming to find an embedding satisfying the order found in phase one, and the 1-D Euclideanness.

4.5 Statement of the main results

In this section we formulate the two closely related main results of this chapter. The first result is technical and states the existence of infinitely many non-1-D Euclidean profiles that are minimal with respect to voter deletion.

Theorem 4.3. *For any integer $k \geq 2$, there exists a preference profile $\mathcal{P}_k^*$ with $n = 2k$ voters and $m = 4k$ alternatives, such that the following holds.*

(a) Profile $\mathcal{P}_k^*$ is not one-dimensional Euclidean.

(b) Profile $\mathcal{P}_k^*$ is minimal in the following sense: the deletion of an arbitrary voter from $\mathcal{P}_k^*$ yields a one-dimensional Euclidean profile.

The proof of Theorem 4.3 is long and will fill most of the rest of this chapter. Here is a quick overview of this proof: Section 4.6 describes the profiles $\mathscr{P}_k^*$. Section 4.7 shows that every profile $\mathscr{P}_k^*$ satisfies Property (a) in Theorem 4.3, while the three Sections 4.8 through 4.10 establish Property (b). Section 4.8 defines the underlying 1-D Euclidean embeddings, Section 4.9 lists a number of technical auxiliary statements, and Section 4.10 establishes the correctness of the 1-D Euclidean embeddings.

To grasp the general idea of why the embeddings defined in Section 4.8 are indeed 1-D Euclidean, the reader can safely skip sections 4.9 and 4.10 to the introduction of Section 4.10 and Section 4.10.1. The latter presents the first subproof of a total of four subproofs in Section 4.10.

As an immediate consequence of Theorem 4.3, we derive our second main result as stated in the following theorem.

Theorem 4.4. *One-dimensional Euclidean preference profiles cannot be characterized in terms of finitely many forbidden substructures.*

Proof. Suppose for the sake of contradiction that such a characterization with finitely many forbidden substructures exists. Let t denote the largest number of voters in any forbidden substructure, and consider a profile $\mathscr{P}_k^*$ from Theorem 4.3 with $k \geq t$. As $\mathscr{P}_k^*$ is not one-dimensional Euclidean, by Property (a), it must contain one of these finitely many forbidden substructures with at most t voters. As profile $\mathscr{P}_k^*$ contains $2k > t+1$ voters, one of its voters is not part of this forbidden substructure. If we delete this voter, the resulting profile will still contain the forbidden substructure; hence it is not one-dimensional Euclidean, which contradicts Property (b). □

4.6 Definition of the profiles

In this section, we begin the proof of Theorem 4.3 by defining the underlying profiles $\mathscr{P}_k^*$. Properties (a) and (b) stated in Theorem 4.3 will be established in the next sections.

We consider $n = 2k$ voters $v_1, v_2, \ldots, v_{2k}$ together with $m = 4k$ alternatives $1, 2, 3, \ldots, 4k$. The preference orders of the voters will be constructed from pieces X_i, Y_i, Z_i with $1 \leq i \leq k$:

$$\begin{aligned} X_i &:= 2k+2i-2 \succ 2k+2i-3 \succ 2k+2i-4 \succ \ldots \succ 2i+2, \\ Y_i &:= 2i-2 \succ 2i-3 \succ 2i-4 \succ \ldots \succ 1, \\ Z_i &:= 2k+2i+1 \succ 2k+2i+2 \succ 2k+2i+3 \succ \ldots \succ 4k. \end{aligned}$$

Note that for every $i = 1, \ldots, k$, the three pieces X_i, Y_i, Z_i cover contiguous intervals of $2k-3$, $2i-2$, and $2k-2i$ alternatives, respectively. Together, the three pieces cover $4k-5$ of the alternatives, and only the five alternatives in the set

$$U_i \;=\; \{2i-1,\; 2i,\; 2i+1\} \,\cup\, \{2k+2i-1,\; 2k+2i\}$$

remain uncovered. Also note that the pieces Y_1 and Z_k are empty. The preference orders of the voters are defined as follows. The two voters v_{2i-1} and v_{2i} always form a couple with fairly similar preferences. For $1 \le i \le k-1$, these voters v_{2i-1} and v_{2i} have the following preferences:

$$v_{2i-1}: \quad X_i \succ 2i+1 \succ 2k+2i-1 \succ 2i \succ 2i-1 \succ 2k+2i \succ Y_i \succ Z_i, \tag{4.4a}$$

$$v_{2i}: \quad X_i \succ 2k+2i-1 \succ 2k+2i \succ 2i+1 \succ 2i \succ 2i-1 \succ Y_i \succ Z_i. \tag{4.4b}$$

Note that the voters v_{2i-1} and v_{2i} both rank the three alternatives $2i+1, 2i, 2i-1$ in U_i in the same decreasing order, with the two other alternatives $2k+2i-1$ and $2k+2i$ shuffled into that order. The last two voters v_{2k-1} and v_{2k} are defined separately:

$$v_{2k-1}: \quad X_k \succ 2k+1 \succ 4k-1 \succ 2k \succ 2k-1 \succ 4k \succ Y_k, \tag{4.5a}$$

$$v_{2k}: \quad X_k \succ 2k+1 \succ 2k \succ \ldots \succ 3 \succ 2 \succ 4k-1 \succ 4k \succ 1. \tag{4.5b}$$

Since piece Z_k is empty, the preferences of voter v_{2k-1} in (4.5a) are actually very similar to the preferences of the other odd-numbered voters v_{2i-1} with $1 \le i \le k-1$ in (4.4a). The last voter v_{2k}, however, behaves quite differently from the other even-numbered voters: on top of his preference order, there are the alternatives in X_k, followed by an intermingling of the alternatives in Y_k and U_k (the alternatives $2k+1, \ldots, 2$ in decreasing order, and then the three alternatives $4k-1$, $4k$, and 1).

Example 4.3. For $k = 4$, the preference profile $\mathscr{P}_4^*$ has $n = 8$ voters and $m = 16$ alternatives and looks as follows (for the sake of readability, we omitted the preference symbol $\succ$ between the alternatives, and for each preference order, we list the alternatives from left to right starting with the most preferred alternative):

v_1:	8	7	6	5	4	3	9	2	1	10	11	12	13	14	15	16
v_2:	8	7	6	5	4	9	10	3	2	1	11	12	13	14	15	16
v_3:	10	9	8	7	6	5	11	4	3	12	2	1	13	14	15	16
v_4:	10	9	8	7	6	11	12	5	4	3	2	1	13	14	15	16
v_5:	12	11	10	9	8	7	13	6	5	14	4	3	2	1	15	16
v_6:	12	11	10	9	8	13	14	7	6	5	4	3	2	1	15	16
v_7:	14	13	12	11	10	9	15	8	7	16	6	5	4	3	2	1
v_8:	14	13	12	11	10	9	8	7	6	5	4	3	2	15	16	1

The alternatives in the five leftmost columns form the pieces X_i. In the first seven rows, the five columns in the center correspond to the sets U_i, while the remaining six columns make up the pieces the Y_i and Z_i. The last row illustrates the unique behavior of the last voter v_8.

4.7 The profiles are not 1-D Euclidean

In this section, we will discuss the single-crossing, single-peaked and one-dimensional Euclidean properties of the profiles $\mathscr{P}_k^*$. First, we can verify that every profile $\mathscr{P}_k^*$ with $k \geq 2$ is single-crossing with respect to the order

$$\langle v_1, v_2, \ldots, v_{2k-2}, v_{2k}, v_{2k-1} \rangle$$

of the voters (that is, the natural order of voters by increasing index, but with the last two voters v_{2k-1} and v_{2k} swapped). As this single-crossing property is of no relevance to our further considerations, the simple proof is omitted. Next, let us turn to single-peakedness, which is crucial for the construction of our embeddings in Section 4.8.

Lemma 4.5. *For $k \geq 2$, the profile $\mathscr{P}_k^*$ is single-peaked. Furthermore, the only two single-peaked orders of the alternatives are the increasing order $1, 2, 3, \ldots, 4k$ and the decreasing order $4k, \ldots, 3, 2, 1$.*

Proof. Every voter v_{2i-1} and v_{2i} with $1 \leq i \leq k$ ranks $2k + 2i - 2$ in the first position. Furthermore, he ranks the small alternatives $1, 2, \ldots, 2k + 2i - 2$ in decreasing order, and he ranks the large alternatives $2k + 2i - 2, 2k + 2i - 1, \ldots, 4k$ in increasing order. Hence, $\mathscr{P}_k^*$ indeed is single-peaked with respect to the permutations

$$\langle 1, 2, \ldots, 4k \rangle \text{ and } \langle 4k, \ldots, 2, 1 \rangle.$$

Next, consider an arbitrary single-peaked permutation $\langle \pi(1), \pi(2), \ldots, \pi(4k) \rangle$ of the alternatives. Since $4k$ and 1 are the least preferred alternatives of voters v_1 and v_{2k}, respectively, these two alternatives must be extremal in the single-peaked permutation; by symmetry we assume that $\pi(1) = 1$ and $\pi(4k) = 4k$.

- Voter v_1 ranks $1 \succ 2k+2 \succ 2k+3 \succ \ldots \succ 4k$, without any other alternatives ranked in-between. This implies $\pi(x) = x$ for $2k+2 \le x \le 4k$.
- Voter v_{2k-1} ranks $2k+2 \succ 2k+1 \succ 2k \succ \ldots \succ 3 \succ 2 \succ 1$. This implies $\pi(x) = x$ for the remaining alternatives x with $1 \le x \le 2k+1$.

Summarizing, we have $\pi(x) = x$ for all x, which completes the proof. $\square$

The following lemma shows that every profile $\mathscr{P}_k^*$ satisfies Property (a) of Theorem 4.3.

Lemma 4.6. *For $k \ge 2$, the profile $\mathscr{P}_k^*$ is not one-dimensional Euclidean.*

Proof. We suppose for the sake of contradiction that profile $\mathscr{P}_k^*$ is 1-D Euclidean. Let $F[j]$ for $j = 1, \ldots, 2k$ and $E[i]$ for $i = 1, \ldots, 4k$ denote a 1-D Euclidean representation of the voters and alternatives. As the 1-D Euclidean representation induces a single-peaked order of the alternatives, we will assume by Lemma 4.5 that the alternatives are embedded in increasing order with

$$E[1] \;<\; E[2] \;<\; E[3] \;<\; \ldots \;<\; E[4k-1] \;<\; E[4k]. \tag{4.6}$$

Next, we claim that in any 1-D Euclidean representation under (4.6), the embedded alternatives satisfy the following inequalities:

$$E[2k+2i-1] + E[2i] < E[2k+2i] + E[2i-1] \qquad \text{for } 1 \le i \le k \tag{4.7a}$$

$$E[2k+2i] + E[2i+1] < E[2k+2i-1] + E[2i+2] \qquad \text{for } 1 \le i \le k-1 \tag{4.7b}$$

$$E[4k] + E[1] < E[4k-1] + E[2] \tag{4.7c}$$

We show the correctness of this claim as follows. For each $i = 1, \ldots, k$, voter v_{2i-1} ranks $2k+2i-1 \succ 2i$ and $2i-1 \succ 2k+2i$, which by Proposition 4.1 yields

$$\frac{1}{2}\left(E[2k+2i-1] + E[2i]\right) \;<\; F[2i-1] \;<\; \frac{1}{2}\left(E[2k+2i] + E[2i-1]\right),$$

which in turn implies (4.7a). Similarly, for $i = 1, \ldots, k-1$, voter v_{2i} ranks $2i+2 \succ 2k+2i-1$ and $2k+2i \succ 2i+1$ which leads to (4.7b). Finally, voter v_{2k} ranks $2 \succ 4k-1$

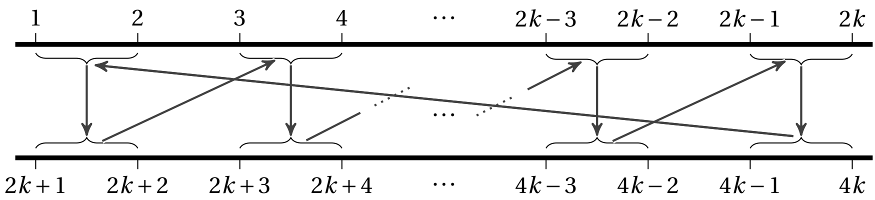


Figure 4.3.: The "smaller" relation of the distances between two neighboring alternatives $2i+1$ and $2i+2$. Arrow "→" indicates the "smaller than ($<$)" relation. For the sake of readability, the top line depicts the real line up to $2k$, and the bottom line depicts the real line starting from $2k+1$.

and $4k > 1$, which implies (4.7c). This shows the correctness of the inequalities (4.7a)–(4.7c). By adding up all the inequalities, we derive the contradiction $\sum_{x=1}^{4k} E[x] < \sum_{x=1}^{4k} E[x]$. □

Before we move on to the next section where we discuss the 1-D Euclidean embeddings, let us take a closer look at the cause of the non-1-D Euclideanness of our constructed profile. First, we rearrange the terms of the inequalities (4.7a)–(4.7c) to obtain the following.

$$E[2i] - E[2i-1] < E[2k+2i] - E[2k+2i-1] \text{ for } 1 \le i \le k \quad (4.8a)$$

$$E[2k+2i] - E[2k+2i-1] < E[2i+2] - E[2i+1] \quad \text{for } 1 \le i \le k-1 \quad (4.8b)$$

$$E[4k] - E[4k-1] < E[2] - E[1] \quad (4.8c)$$

By the chain of inequalities (4.6), the alternatives $1, 2, \ldots, 4k$ are embedded from left to right in increasing order. Thus, both the left-hand side and the right-hand side of the inequalities (4.8a)–(4.8c) refer to distances of two neighboring alternatives $2i-1$ and $2i$ in the embedding. Thus, if we consider the "smaller than ($<$)" relation of these distances, then we find that this relation is cyclic, which is impossible for any 1-D Euclidean representation. See Figure 4.3 for an illustration. In fact, any preference profile with a "cyclic" relation is not 1-D Euclidean. We will discuss this issue in Section 4.11.

4.8 Definition of the 1-D Euclidean embeddings

This section, as well as the following two sections, contains mostly technical results which, while important, are not necessary to read in order to follow the general idea of this chapter.

We fix an integer s with $1 \le s \le 2k$ and construct 1-D Euclidean embeddings F_s and E_s of the voters and alternatives in profile $\mathscr{P}_k^*$. We show that F_s (minus voter v_s's embedding) and E_s together form a 1-D Euclidean representation of the profile obtained from $\mathscr{P}_k^*$ by deleting v_s.

We start by defining the 1-D Euclidean embedding E_s of the alternatives. We anchor the embedding by placing the first alternative at the position

$$E_s[1] \;=\; 0. \tag{4.9}$$

The remaining values $E_s[2], \ldots, E_s[4k]$ are defined recursively by the equations (4.10)–(4.15) below. For $1 \le i \le k-1$, we set

$$E_s[2i+1] - E_s[2i] \;=\; 2 \tag{4.10}$$

and for $1 \le i \le k$ we set

$$E_s[2i] - E_s[2i-1] \;=\; (4i - 2s - 3 \bmod 4k). \tag{4.11}$$

Note that the relations (4.9)–(4.11) define $E_s[x]$ for all $x \le 2k$. For $1 \le i \le k-1$, we set

$$\begin{aligned} & E_s[2k+2i-1] - E_s[2k+2i-2] \\ =\; & \begin{cases} E_s[2k+2i-3] - E_s[2i+1] + 2 & \text{if } s \ne 2i-1 \\ E_s[2k+2i-3] - E_s[2i+2] + 2 & \text{if } s = 2i-1. \end{cases} \end{aligned} \tag{4.12}$$

For $1 \le i \le k-1$, we define

$$E_s[2k+2i] - E_s[2k+2i-1] \;=\; (4i - 2s - 1 \bmod 4k). \tag{4.13}$$

Note that the relations (4.12) and (4.13) define $E_s[x]$ for all x with $2k+1 \le x \le 4k-2$. Finally, we define the 1-D Euclidean embedding of the last two alternatives as

$$E_s[4k-1] - E_s[4k-2] \;=\; \begin{cases} E_s[4k-3] - E_s[2] + 2 & \text{if } s \ne 2k \\ E_s[4k-3] - E_s[2k+1] + 2 & \text{if } s = 2k \end{cases} \tag{4.14}$$

and

$$E_s[4k] - E_s[4k-1] \;=\; \begin{cases} E_s[2] - E_s[1] - 2 & \text{if } s \ne 2k \\ E_s[2k+1] - E_s[2k-1] & \text{if } s = 2k. \end{cases} \tag{4.15}$$

This completes the description of the 1-D Euclidean embedding E_s of the alternatives. Note that $E_s[x]$ is integer for all alternatives x.

	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}	d_{11}	d_{12}	d_{13}	d_{14}	d_{15}	d_{16}
E_1:	**15**	2	**3**	2	**7**	2	**11**	13	**1**	35	**5**	62	**9**	145	**13**
E_2:	**13**	2	**1**	2	**5**	2	**9**	12	**15**	30	**3**	68	**7**	151	**11**
E_3:	**11**	2	**15**	2	**3**	2	**7**	24	**13**	35	**1**	81	**5**	187	**9**
E_4:	**9**	2	**13**	2	**1**	2	**5**	20	**11**	30	**15**	68	**3**	171	**7**
E_5:	**7**	2	**11**	2	**15**	2	**3**	32	**9**	54	**13**	97	**1**	242	**5**
E_6:	**5**	2	**9**	2	**13**	2	**1**	28	**7**	46	**11**	84	**15**	207	**3**
E_7:	**3**	2	**7**	2	**11**	2	**15**	24	**5**	54	**9**	100	**13**	233	**1**
E_8:	**1**	2	**5**	2	**9**	2	**13**	20	**3**	46	**7**	84	**11**	142	**33**

Table 4.1.: This table is discussed in Example 4.4 and illustrates the 1-D Euclidean embedding of the alternatives in profile $\mathscr{P}_4^*$. Every row is labeled with the corresponding embedding E_s. If a column is labeled d_i, then its entries show the 1-D Euclidean distances $E_s[i] - E_s[i-1]$ between the two consecutively embedded alternatives $i-1$ and i.

Lemma 4.7. *The embedding E_s satisfies $E_s[x] < E_s[y]$ for all alternatives x and y with $1 \le x < y \le 4k$. In other words, E_s satisfies the inequalities in* (4.6).

Proof. The statement follows from (4.9)–(4.15) by an inductive argument. The right-hand sides in (4.10), (4.11), and (4.13) are all positive. The right-hand sides in (4.12) and (4.14) can be shown to be positive by induction. Finally for $i = 1$ and $s \neq 2k$, the right-hand side of (4.11) is a positive odd integer greater than 1; this yields $E_s[2] \ge 3$ which implies that the right-hand side $E_s[2] - E_s[1] - 2$ in (4.15) is positive, too. □

Example 4.4. We continue our discussion of profile $\mathscr{P}_4^*$ from Example 4.3. For every embedding E_s with $1 \le s \le 8$, the corresponding row in Table 4.1 lists the distances $d_i = E_s[i] - E_s[i-1]$ between pairs of consecutive alternatives according to formulas (4.10)–(4.15). For instance, the intersection of row E_5 and column d_4 contains an entry with value 11; this means that in the 1-D Euclidean representation E_5, the distance $E_5[4] - E_5[3]$ between the embedded alternatives 3 and 4 equals 11. As $E_s[1] = 0$, we see that for $2 \le i \le 4k$, $E_s[i]$ equals $d_2 + d_3 + \cdots + d_i$. For instance in E_5, alternative 4 will be embedded at $E_5[4] = 7 + 2 + 11 = 20$.

The reader will notice the periodic structure of part of the data in Table 4.1. For instance, every even-numbered column except the last one contains a circular shift of the eight numbers 15, 13, 11, 9, 7, 5, 3, 1 as shown in boldface, which results from equations (4.11) and (4.13). Furthermore, all entries in the three columns d_3, d_5, d_7 have the same value 2 because of (4.10). The values in other parts of the table look somewhat irregular and chaotic, which is caused by formula (4.12). For us, the most convenient way of analyzing this data is via the recursive definitions (4.9)–(4.15). □

Now let us turn to the 1-D Euclidean embedding of the voters. The Euclidean position $F_s[j]$ of every voter v_j is the average of exactly four embedded alternatives. For $1 \le i \le k-1$, we define

$$F_s[2i-1] \;=\; \frac{1}{4}\,(E_s[2i-1]+E_s[2i]+E_s[2k+2i-1]+E_s[2k+2i])\,. \tag{4.16}$$

Similarly, for $1 \le i \le k-1$, we define

$$F_s[2i] \;=\; \frac{1}{4}\,(E_s[2i+1]+E_s[2i+2]+E_s[2k+2i-1]+E_s[2k+2i])\,. \tag{4.17}$$

If $s \neq 2k$, then we embed voter v_{2k-1} according to (4.16), while for $s = 2k$, we embed it in a slightly different way. More precisely, we set

$$F_s[2k-1] \;=\; \begin{cases} \frac{1}{4}\,(E_s[2k-1]+E_s[2k]+E_s[4k-1]+E_s[4k]) & \text{if } s \neq 2k \\ \frac{1}{4}\,(E_s[2k-2]+E_s[2k+1]+E_s[4k-1]+E_s[4k]) & \text{if } s = 2k. \end{cases} \tag{4.18}$$

Finally, the very last voter v_{2k} is placed at

$$F_s[2k] \;=\; \frac{1}{4}\,(E_s[1]+E_s[2]+E_s[4k-1]+E_s[4k])\,. \tag{4.19}$$

Equations (4.16)–(4.19) define $F_s[j]$ for all voters v_j with $1 \le j \le 2k$. This completes the description of the 1-D Euclidean representation F_s of the voters.

We note that the location $F_s[s]$ of voter v_s has been specified, but will be irrelevant to our further arguments. We show that F_s and E_s together constitute a correct 1-D Euclidean representation of the $2k-1$ voters in $\{v_1,\ldots,v_{2k}\}\backslash\{v_s\}$ together with all $4k$ alternatives $1,2,\ldots,4k$. In other words, the deletion of voter v_s from profile $\mathscr{P}_k^*$ yields a one-dimensional Euclidean profile, which completes the proof of Property (b) in Theorem 4.3. To this end, we establish the following lemma which we prove in Section 4.10.

Lemma 4.8. *For all r and s with $1 \le r \neq s \le 2k$, the E_s and F_s together form a 1-D Euclidean representation of the preferences of voter v_r.*

The correctness of Lemma 4.8 for small profiles $\mathcal{P}_k^*$ with $k \in \{2,3,4\}$ can be easily verified by a computer program (or by a human going through a lot of tedious case distinctions). Hence, from now on we assume that

$$k \geq 5. \tag{4.20}$$

This assumption will considerably shorten and simplify our arguments. Furthermore, note that the proof of our main result in Theorem 4.4 is not affected by this assumption, as it builds on the profiles $\mathcal{P}_k^*$ for which k is large and tends towards infinity.

4.9 A collection of technical results

The embeddings we defined in the previous section are compact but hard to comprehend. In order to show that they are indeed one-dimensional Euclidean, it suffices to know the distance between two alternatives instead of their exact positions. Thus, in this section, we state five technical lemmas which basically restate or summarize the distances of two alternatives in the embeddings. We will see that they are extensively used in Section 4.10, where we show the correctness of the embeddings we defined.

We briefly explain the purpose of each of these five lemmas. Lemmas 4.9 and 4.10 summarize a number of useful identities which basically rewrite the distances between two consecutive alternatives in the embeddings. They serve as references in our later analysis. Lemmas 4.11 through 4.13 state important *inequalities* that will be central to our proofs of the 1-D Euclideanness of the embeddings defined in Section 4.8; the proofs are shown in Section 4.10. Readers can thus use the results of this section as a sort of black box and skip to the introduction of Section 4.10 in order to follow the narrative of this chapter.

Recall that throughout we assume that $k \geq 5$ (4.20) and that from Section 4.8, for each integer s, $1 \leq s \leq 2k$, we construct F_s and E_s for the profile obtained from $\mathcal{P}_k^*$ by deleting the voter v_s.

Lemma 4.9. *For $1 \leq s \leq 2k$, the 1-D Euclidean embedding E_s satisfies the following.*

$$E_s[2] = 4k - 2s + 1 \tag{4.21a}$$

$$E_s[3] = 4k - 2s + 3 \tag{4.21b}$$

$$E_s[4] = \begin{cases} 4k - 4s + 8 & \text{if } s \in \{1,2\} \\ 8k - 4s + 8 & \text{if } s \geq 3. \end{cases} \tag{4.21c}$$

Furthermore, for $s \in \{1,2\}$, the embedding E_s satisfies the following.

$$E_s[2k-2] - E_s[2k-3] \;= 4k-2s-7 \tag{4.22a}$$

$$E_s[2k-4] - E_s[2k-5] \;= 4k-2s-11 \tag{4.22b}$$

Proof. These statements follow by straightforward calculations from (4.9)–(4.13). □

Lemma 4.10. *If (a) $1 \le i \le k-1$ and $s \ne 2i-1$, or if (b) $i = k$ and $s \notin \{2k-1, 2k\}$, then the following holds:*

$$E_s[2k+2i] - E_s[2k+2i-1] \;=\; E_s[2i] - E_s[2i-1] + 2 \tag{4.23a}$$

If (c) $1 \le i \le k-1$ and $s \ne 2i$, the following holds:

$$E_s[2k+2i] - E_s[2k+2i-1] \;=\; E_s[2i+2] - E_s[2i+1] - 2 \tag{4.23b}$$

Proof. We distinguish five cases. The first case assumes $s = 2i-1$. In the setting of this lemma, this case can only occur under (c) with $1 \le i \le k-1$. Then (4.13) yields $E_s[2k+2i] - E_s[2k+2i-1] = 1$, while (4.11) yields $E_s[2i+2] - E_s[2i+1] = 3$. This implies the desired equality (4.23b) for this case.

The second case assumes $s = 2i$. In the setting of this lemma, this case can only occur under (a) with $1 \le i \le k-1$. Then (4.13) yields $E_s[2k+2i] - E_s[2k+2i-1] = 4k-1$, while (4.11) yields $E_s[2i] - E_s[2i-1] = 4k-3$. This implies the desired equality (4.23a).

The third case assumes $i = k$. In the setting of this lemma, this case can only occur under (b) with $1 \le s \le 2k-2$. Then (4.15) and (4.21a) yield $E_s[4k] - E_s[4k-1] = 4k-2s-1$, while (4.11) yields $E_s[2k] - E_s[2k-1] = 4k-2s-3$. This implies the desired equality (4.23a).

In the remaining cases we always have $s \notin \{2i-1, 2i\}$. The fourth case assumes that $1 \le i \le k-1$ and that $s = 2\ell - 1$ is odd, where $1 \le \ell \le k$ and $\ell \ne i$. In the setting of this lemma, this case can only occur under (a) and (b). Then (4.13) yields

$$E_s[2k+2i] - E_s[2k+2i-1] \quad = \quad 4(i-\ell)+1 \bmod 4k, \tag{4.24}$$

while (4.11) yields $E_s[2i] - E_s[2i-1] = 4(i-\ell) - 1 \bmod 4k$. Since $i - \ell \ne 0$, these two equations together yield (4.23a). Furthermore, (4.11) yields $E_s[2i+2] - E_s[2i+1] = 4(i-\ell)+3 \bmod 4k$, which together with (4.24) gives (4.23b).

The fifth case assumes that $1 \le i \le k-1$ and that $s = 2\ell$ is even, where $1 \le \ell \le k$ and $\ell \ne i$. In the setting of this lemma, this case can only occur under (a) and (c). Then (4.13) yields

$$E_s[2k+2i] - E_s[2k+2i-1] \quad = \quad 4(i-\ell)-1 \bmod 4k, \tag{4.25}$$

while (4.11) yields $E_s[2i] - E_s[2i-1] = 4(i-\ell) - 3 \bmod 4k$. Since $i - \ell \neq 0$, these two statements together imply (4.23a). Finally, (4.11) yields $E_s[2i+2] - E_s[2i+1] = 4(i-\ell) + 1 \bmod 4k$. As $i - \ell \neq 0$, this inequality together with (4.25) yields (4.23b). This completes the proof. □

Lemma 4.11. *For all alternatives x and y with $1 \leq y \leq x \leq 4k$, the embedding E_s satisfies the inequality $E_s[x] - E_s[y] \geq x - y$.*

Proof. This follows from Lemma 4.7 and the integrality of E_s. □

Lemma 4.12. *All i and s with $1 \leq i \leq 2k-1$ and $1 \leq s \leq 2k$ satisfy the following inequality.*

$$E_s[2i+1] - E_s[2i] \;\geq\; 2. \tag{4.26}$$

Proof. For $1 \leq i \leq k-1$, this follows directly from (4.10). For $k \leq i \leq 2k-1$, this follows from (4.12) and (4.14) in combination with Lemma 4.11. □

Lemma 4.13. *All i and s with $1 \leq i \leq k-1$ and $1 \leq s \leq 2k$ satisfy the following inequality.*

$$E_s[2k+2i-1] \;\geq\; E_s[2k] + E_s[2i] + 2. \tag{4.27}$$

Proof. We show the inequality by induction on $i = 1, \ldots, k-1$. For the inductive base case $i = 1$ we distinguish between two cases for the value of s. The first case assumes $s \in \{1, 2\}$. Then (4.12) and $k \geq 5$, together with (4.10), (4.22a), (4.22b), and (4.21a) yield

$$\begin{aligned}
E_s[2k+2i-1] - E_s[2k] \;&\geq\; E_s[2k-1] - E_s[4] + 2 \\
\geq\;& (E_s[2k-1] - E_s[2k-2]) + (E_s[2k-2] - E_s[2k-3]) \\
&\quad + (E_s[2k-3] - E_s[2k-4]) + (E_s[2k-4] - E_s[2k-5]) + 2 \\
=\;& 2 + (4k - 2s - 7) + 2 + (4k - 2s - 11) + 2 \\
=\;& 8k - 4s - 12 \;>\; (4k - 2s + 1) + 2 \;=\; E_s[2] + 2.
\end{aligned}$$

The second case assumes $s \geq 3$. Then, the first line of (4.12) together with $k \geq 5$, (4.21c), (4.21b) and (4.21a) yields

$$\begin{aligned}
E_s[2k+2i-1] - E_s[2k] \;&=\; E_s[2k-1] - E_s[3] + 2 \\
\geq\;& E_s[4] - E_s[3] + 2 \;=\; (8k - 4s + 8) - (4k - 2s + 3) + 2 \\
=\;& 4k - 2s + 7 \;>\; E_s[2] + 2.
\end{aligned}$$

Summarizing, in both cases we have established the desired inequality (4.27). This completes the analysis of the inductive base case $i = 1$. Next, let us state the inductive assumption as

$$E_s[2k+2i-3] \quad \geq \quad E_s[2k] + E_s[2i-2] + 2. \tag{4.28}$$

In the inductive step, we will use the following implication of (4.12):

$$E_s[2k+2i-1] - E_s[2k+2i-2] \quad \geq \quad E_s[2k+2i-3] - E_s[2i+2] + 2. \tag{4.29}$$

Furthermore, by (4.13), the left-hand side of the following inequality equals $(4i-2s-5 \bmod 4k)$, while by (4.11), its right-hand side equals $(4i-2s-3 \bmod 4k)-2$. This implies

$$E_s[2k+2i-2] - E_s[2k+2i-3] \quad \geq \quad E_s[2i] - E_s[2i-1] - 2. \tag{4.30}$$

Adding up (4.28), (4.29) and (4.30), and rearranging and simplifying the resulting inequality yields

$$\begin{aligned} & E_s[2k+2i-1] - E_s[2k] - E_s[2i] - 2 \\ & \quad \geq \quad E_s[2k+2i-3] - E_s[2i+2] + E_s[2i-2] - E_s[2i-1] \\ & \quad \geq \quad (2k+2i-3) - (2i+2) - 2 \ = \ 2k-7 \ > \ 0. \end{aligned}$$

Here we use Lemma 4.11 to bound $E_s[2k+2i-3]-E_s[2i+2]$, and we use equality (4.10) to get rid of $E_s[2i-2]-E_s[2i-1]$. As this implies equality (4.27), the inductive argument is complete. □

4.10 Correctness of the 1-D Euclidean embeddings

In this section, we prove Lemma 4.8, that is, we show that the embeddings defined in Section 4.8 are one-dimensional Euclidean. This directly implies Property (b) of Theorem 4.3, one of our main contribution.

Let v_r and v_s be two arbitrary voters with $r \neq s$. We recall that by Lemma 4.7 the 1-D Euclidean representation E_s embeds the alternatives $1, \ldots, 4k$ in increasing order from left to right. We show that any two alternatives x and y with $x \succ_r y$ which are consecutive in the preference order of voter v_r satisfy

$$2F_s[r] \ < E_s[x] + E_s[y] \qquad \text{whenever } x < y, \tag{4.31a}$$

$$2F_s[r] \ > E_s[x] + E_s[y] \qquad \text{whenever } x > y. \tag{4.31b}$$

By our construction, all preference orders in profile $\mathscr{P}_k^*$ contain long monotonously increasing or decreasing runs of alternatives. By Proposition 4.2, it is therefore sufficient to establish (4.31a) and (4.31b) at the few turning points where the preference order of voter v_r changes its monotonicity behavior. We emphasize that the first pair of alternatives in every preference order forms a turning point by default.

Our proof distinguishes between four cases which are handled separately in the following four subsections. Sections 4.10.1 and 4.10.2 deal with the cases with odd r, while Sections 4.10.3 and 4.10.4 deal with the cases where r is even.

Again, we remark that this section consists of four parts of long technical proofs. To better follow the main idea of this chapter, it may be worthwhile to skip to the next section where we show an interesting technique of how to obtain non-1-D Euclidean profiles for the so-called "cyclic relation" mentioned at the end of Section 4.7.

4.10.1 The cases with odd r (with a single exception)

In this section we assume that $r = 2i-1$ for $1 \le i \le k$, implying that $s \neq 2i-1$ because v_s is the deleted voter. If $i = k$ (and hence $r = 2k-1$), then we additionally assume $s \neq 2k$; the remaining case with $i = k$ and $s = 2k$ will be dealt with in the next subsection. Note that under these assumptions, the value $F_s[2i-1]$ is given by (4.16). Furthermore (4.23a) in Lemma 4.10 yields

$$E_s[2k+2i] + E_s[2i-1] \; = \; E_s[2i] + E_s[2k+2i-1] + 2. \tag{4.32}$$

In order to prove (4.31a) and (4.31b) for the preference orders in (4.4a) and (4.5a), it is sufficient to establish the following six inequalities for the turning points.

$$2F_s[2i-1] \; > E_s[2k+2i-2] + E_s[2k+2i-3] \tag{4.33a}$$

$$2F_s[2i-1] \; < E_s[2i+1] + E_s[2k+2i-1] \tag{4.33b}$$

$$2F_s[2i-1] \; > E_s[2k+2i-1] + E_s[2i] \tag{4.33c}$$

$$2F_s[2i-1] \; < E_s[2i-1] + E_s[2k+2i] \tag{4.33d}$$

$$2F_s[2i-1] \; > E_s[2k+2i] + E_s[2i-2] \tag{4.33e}$$

$$2F_s[2i-1] \; < E_s[1] + E_s[2k+2i+1] \tag{4.33f}$$

Note that for $i = 1$, inequality (4.33e) vanishes as Y_1 is empty, and that for $i = k$ inequality (4.33f) vanishes as Z_k is empty. We use equation (4.16) or the first line of equation (4.18) together with equation (4.32), and rewrite the left-hand side of all inequalities (4.33a)–(4.33f) as

$$
\begin{aligned}
2F_s[2i-1] &= \frac{1}{2}\left(E_s[2i-1]+E_s[2i]+E_s[2k+2i-1]+E_s[2k+2i]\right) \\
&= E_s[2i]+E_s[2k+2i-1]+1 \;=\; E_s[2i-1]+E_s[2k+2i]-1.
\end{aligned}
\tag{4.34}
$$

For (4.33a), we distinguish between two subcases. The first subcase assumes $i \le k-1$. We use (4.34), (4.12) with $s \neq 2i-1$, and (4.10) to obtain

$$
\begin{aligned}
&2F_s[2i-1]-E_s[2k+2i-2]-E_s[2k+2i-3] \\
&\quad= \quad (E_s[2i]+E_s[2k+2i-1]+1)-E_s[2k+2i-2]-E_s[2k+2i-3] \\
&\quad= \quad E_s[2i]+1-E_s[2i+1]+2 \;=\; 1 \;>\; 0.
\end{aligned}
$$

The second subcase deals with the remaining case $i = k$. We use (4.34), the first line in (4.14), and Lemma 4.7 to obtain

$$
\begin{aligned}
&2F_s[2i-1]-E_s[2k+2i-2]-E_s[2k+2i-3] \\
&\quad= \quad (E_s[2k]+E_s[4k-1]+1)-E_s[4k-2]-E_s[4k-3] \\
&\quad= \quad (E_s[2k]+1)+(-E_s[2]+2) \;=\; (E_s[2k]-E_s[2])+3 \;>\; 0.
\end{aligned}
$$

For (4.33b), using (4.34) and (4.26) yields

$$
\begin{aligned}
&2F_s[2i-1]-E_s[2i+1]-E_s[2k+2i-1] \\
&\quad= \quad (E_s[2i]+E_s[2k+2i-1]+1)-E_s[2i+1]-E_s[2k+2i-1] \;<\; 0.
\end{aligned}
$$

For (4.33c), from (4.34) we obtain

$$
\begin{aligned}
&2F_s[2i-1]-E_s[2k+2i-1]-E_s[2i] \\
&\quad= \quad (E_s[2i]+E_s[2k+2i-1]+1)-E_s[2k+2i-1]-E_s[2i] \;=\; 1 \;>\; 0.
\end{aligned}
$$

For (4.33d), we use (4.34) and get

$$
\begin{aligned}
&2F_s[2i-1]-E_s[2i-1]-E_s[2k+2i] \\
&\quad= \quad (E_s[2i-1]+E_s[2k+2i]-1)-E_s[2i-1]-E_s[2k+2i] \;=\; -1 \;<\; 0.
\end{aligned}
$$

For (4.33e) with $i \ge 2$, using (4.34) and (4.10) we obtain

$$
\begin{aligned}
&2F_s[2i-1]-E_s[2k+2i]-E_s[2i-2] \\
&\quad= \quad (E_s[2i-1]+E_s[2k+2i]-1)-E_s[2k+2i]-E_s[2i-2] \;=\; 1 \;>\; 0.
\end{aligned}
$$

It remains to prove inequality (4.33f) which takes more effort. Since (4.33f) vanishes for $i = k$, we assume $i \le k-1$. We first use (4.34) and (4.9) to derive

$$\begin{aligned} & 2F_s[2i-1] - E_s[1] - E_s[2k+2i+1] \\ = \quad & E_s[2i-1] - 1 - (E_s[2k+2i+1] - E_s[2k+2i]). \end{aligned} \tag{4.35}$$

Our goal is to show that the value in (4.35) is strictly negative, and for this we distinguish between three subcases. The first subcase assumes $i \le k-2$. We use (4.12), (4.27), and Lemma 4.7 to obtain

$$\begin{aligned} & E_s[2i-1] - 1 - (E_s[2k+2i+1] - E_s[2k+2i]) \\ \le \quad & E_s[2i-1] - 1 - (E_s[2k+2i-1] - E_s[2i+4] + 2) \\ \le \quad & E_s[2i-1] - 3 + E_s[2i+4] - (E_s[2k] + E_s[2i] + 2) \\ = \quad & (E_s[2i+4] - E_s[2k]) + (E_s[2i-1] - E_s[2i]) - 5 \ < \ 0. \end{aligned}$$

The second subcase assumes $i = k-1$ and $s \ne 2k$. We use (4.14), (4.27), and Lemma 4.7 to obtain

$$\begin{aligned} & E_s[2i-1] - 1 - (E_s[2k+2i+1] - E_s[2k+2i]) \\ = \quad & E_s[2k-3] - 1 - (E_s[4k-1] - E_s[4k-2]) \\ = \quad & E_s[2k-3] - 1 - (E_s[4k-3] - E_s[2] + 2) \\ \le \quad & E_s[2k-3] - 3 + E_s[2] - (E_s[2k] + E_s[2k-2] + 2) \\ = \quad & (E_s[2k-3] - E_s[2k]) + (E_s[2] - E_s[2k-2]) - 5 \ < \ 0. \end{aligned}$$

The third and last subcase assumes $i = k-1$ and $s = 2k$. We use the second line of (4.14), the first line of (4.12), inequality (4.27), equality (4.10), and Lemma 4.7 to obtain

$$\begin{aligned} & E_s[2i-1] - 1 - (E_s[2k+2i+1] - E_s[2k+2i]) \\ = \quad & E_s[2k-3] - 1 - (E_s[4k-1] - E_s[4k-2]) \\ = \quad & E_s[2k-3] - 1 - (E_s[4k-3] - E_s[2k+1] + 2) \\ = \quad & E_s[2k-3] - 3 + E_s[2k+1] - (E_s[4k-4] + E_s[4k-5] - E_s[2k-1] + 2) \\ \le \quad & E_s[2k-3] - 5 + E_s[2k+1] - E_s[4k-4] \\ & \quad + E_s[2k-1] - (E_s[2k] + E_s[2k-4] + 2) \\ = \quad & (E_s[2k+1] - E_s[4k-4]) + (E_s[2k-1] - E_s[2k]) - 5 \ < \ 0. \end{aligned}$$

As (4.35) is strictly negative in each of the three subcases, the proof of (4.33f) is complete. Together, E_s and F_s form a 1-D Euclidean representation of the preferences of voter v_r.

4.10.2 The exceptional case with odd r

In this section we consider the special case when $i = k$ (and hence $r = 2k-1$) and $s = 2k$, which has been left open in the previous subsection. In this case, the embedding $F_s[2k-1]$ is given by the second option in formula (4.18). Furthermore, (4.15) and (4.10) yield

$$E_s[4k] - E_s[4k-1] \quad = \quad E_s[2k+1] - E_s[2k-2] - 2.$$

Altogether this leads to

$$\begin{aligned} 2F_s[2k-1] &= \frac{1}{2}\left(E_s[2k-2] + E_s[2k+1] + E_s[4k-1] + E_s[4k]\right) \\ &= E_s[2k-2] + E_s[4k] + 1 \;=\; E_s[2k+1] + E_s[4k-1] - 1. \end{aligned} \tag{4.36}$$

As inequality (4.33f) vanishes for $i = k$, our goal in this section is to establish the five inequalities (4.33a)–(4.33e) for $i = k$ and $s = 2k$. For (4.33a), we use (4.36) and (4.14) and get

$$\begin{aligned} & 2F_s[2k-1] - E_s[4k-2] - E_s[4k-3] \\ &= \quad (E_s[2k+1] + E_s[4k-1] - 1) - E_s[4k-2] - E_s[4k-3] \\ &= \quad E_s[2k+1] - E_s[4k-3] - 1 + (E_s[4k-3] - E_s[2k+1] + 2) \;=\; 1 \;>\; 0. \end{aligned}$$

For (4.33b), we use (4.36) and obtain

$$\begin{aligned} & 2F_s[2k-1] - E_s[2k+1] - E_s[4k-1] \\ &= \quad (E_s[2k+1] + E_s[4k-1] - 1) - E_s[2k+1] - E_s[4k-1] \;=\; -1 \;<\; 0. \end{aligned}$$

For (4.33c), using (4.36) and (4.26) yields

$$\begin{aligned} & 2F_s[2k-1] - E_s[4k-1] - E_s[2k] \\ &= \quad (E_s[2k+1] + E_s[4k-1] - 1) - E_s[4k-1] - E_s[2k] \;>\; 0. \end{aligned}$$

For (4.33d), we use (4.36) and (4.10) and get

$$\begin{aligned} & 2F_s[2k-1] - E_s[2k-1] - E_s[4k] \\ &= \quad (E_s[2k-2] + E_s[4k] + 1) - E_s[2k-1] - E_s[4k] \;=\; -1 \;<\; 0. \end{aligned}$$

For (4.33e), from (4.36) we obtain

$$\begin{aligned} & 2F_s[2k-1] - E_s[4k] - E_s[2k-2] \\ = \quad & (E_s[2k-2] + E_s[4k] + 1) - E_s[4k] - E_s[2k-2] \;=\; 1 \;>\; 0. \end{aligned}$$

This completes the analysis of the special case with odd r. In this case as well, E_s and F_s together form a 1-D Euclidean representation of preferences of voter v_r.

4.10.3 The cases with even r (with a single exception)

In this section, we consider the cases of even $r = 2i$ for $1 \le i \le k-1$, and of $s \neq 2i$. We deal with the remaining case where $r = 2k$ in the next subsection. Note that in this case, the value $F_s[2i]$ is given by (4.17). Furthermore (4.23b) in Lemma 4.10 yields

$$E_s[2i+1] + E_s[2k+2i] \;=\; E_s[2i+2] + E_s[2k+2i-1] - 2. \tag{4.37}$$

In order to prove (4.31a) and (4.31b) for the preference orders in (4.4b), it is sufficient to show that the following four inequalities for the turning points hold.

$$2F_s[2i] \;> E_s[2k+2i-2] + E_s[2k+2i-3], \tag{4.38a}$$

$$2F_s[2i] \;< E_s[2i+2] + E_s[2k+2i-1], \tag{4.38b}$$

$$2F_s[2i] \;> E_s[2k+2i] + E_s[2i+1], \tag{4.38c}$$

$$2F_s[2i] \;< E_s[1] + E_s[2k+2i+1]. \tag{4.38d}$$

We use the definition of $F_s[2i]$ in (4.17) together with (4.37) to rewrite the left-hand side of all inequalities (4.38a)–(4.38d) as

$$\begin{aligned} 2F_s[2i] &= \frac{1}{2}\,(E_s[2i+1] + E_s[2i+2] + E_s[2k+2i-1] + E_s[2k+2i]) \\ &= E_s[2i+1] + E_s[2k+2i] + 1 \;=\; E_s[2i+2] + E_s[2k+2i-1] - 1. \end{aligned} \tag{4.39}$$

For (4.38a), we use (4.39) and (4.12) to obtain

$$\begin{aligned} & 2F_s[2i] - E_s[2k+2i-2] - E_s[2k+2i-3] \\ = \quad & (E_s[2i+2] + E_s[2k+2i-1] - 1) - E_s[2k+2i-2] - E_s[2k+2i-3] \\ \ge \quad & E_s[2i+2] - E_s[2k+2i-3] - 1 + (E_s[2k+2i-3] - E_s[2i+2] + 2) = 1 > 0. \end{aligned}$$

For (4.38b), from (4.39) we get

$$\begin{aligned}
&2F_s[2i] - E_s[2i+2] - E_s[2k+2i-1] \\
&\quad = \quad (E_s[2i+2] + E_s[2k+2i-1] - 1) - E_s[2i+2] - E_s[2k+2i-1] = -1 < 0.
\end{aligned}$$

For (4.38c), using (4.39) yields

$$\begin{aligned}
&2F_s[2i] - E_s[2k+2i] - E_s[2i+1] \\
&\quad = \quad (E_s[2i+1] + E_s[2k+2i] + 1) - E_s[2k+2i] - E_s[2i+1] = 1 > 0.
\end{aligned}$$

It remains to prove inequality (4.38d) which takes a considerable amount of work. We distinguish between three subcases. The first subcase assumes $1 \le i \le k-2$. Then Lemma 4.7 implies $E_s[2i+4] \le E_s[2k]$. We use (4.39), (4.9), (4.12), (4.27) and (4.10) to derive

$$\begin{aligned}
&2F_s[2i] - E_s[1] - E_s[2k+2i+1] \\
&\quad = \quad (E_s[2i+1] + E_s[2k+2i] + 1) - E_s[2k+2i+1] \\
&\quad \le \quad E_s[2i+1] + 1 - (E_s[2k+2i-1] - E_s[2i+4] + 2) \\
&\quad \le \quad E_s[2i+1] + E_s[2i+4] - 1 - (E_s[2k] + E_s[2i] + 2) \\
&\quad = \quad E_s[2i+4] - E_s[2k] - 1 \; < \; 0.
\end{aligned}$$

The second subcase assumes $i = k-1$ and $s \neq 2k$. To prove (4.38d), we use (4.39), (4.9), (4.14), (4.27) and (4.10) and obtain

$$\begin{aligned}
&2F_s[2k-2] - E_s[1] - E_s[4k-1] \\
&\quad = \quad (E_s[2k-1] + E_s[4k-2] + 1) - E_s[4k-1] \\
&\quad = \quad E_s[2k-1] + 1 - (E_s[4k-3] - E_s[2] + 2) \\
&\quad \le \quad E_s[2k-1] + E_s[2] - 1 - (E_s[2k] + E_s[2k-2] + 2) \\
&\quad = \quad E_s[2] - E_s[2k] - 1 \; < \; 0.
\end{aligned}$$

The third and last subcase assumes $i = k-1$ and $s = 2k$. We begin by deriving a number of auxiliary equations and inequalities. First, we obtain $E_s[3] = 3$ from (4.21b), and use (4.12) and (4.10) to get

$$\begin{aligned}
&E_s[2k+1] - E_s[2k-2] \\
&\quad = \quad (E_s[2k+1] - E_s[2k]) + (E_s[2k] - E_s[2k-1]) + (E_s[2k-1] - E_s[2k-2]) \\
&\quad = \quad (E_s[2k-1] - E_s[3] + 2) + (E_s[2k] - E_s[2k-1]) + 2 \; = \; E_s[2k] + 1. \qquad (4.40)
\end{aligned}$$

Next, (4.23b) gives us

$$E_s[2k-2]-E_s[2k-3]-2 \quad = \quad E_s[4k-4]-E_s[4k-5]. \tag{4.41}$$

We express $E_s[4k-3]$ once by the first line of (4.12) and once by the second line of (4.14), which by equating yields

$$\begin{aligned} & E_s[4k-4]+E_s[4k-5]-E_s[2k-1]+2 \\ = \quad & E_s[2k+1]-2+E_s[4k-1]-E_s[4k-2]. \end{aligned} \tag{4.42}$$

Next, we add (4.40), (4.41), (4.42) and rearrange the result to obtain

$$\begin{aligned} & E_s[2k-1]+E_s[4k-2]-E_s[4k-1]+1 \\ = \quad & 2E_s[2k-1]+E_s[2k]+E_s[2k-3]-2E_s[4k-5]. \end{aligned} \tag{4.43}$$

We obtain $E_s[2k]-E_s[2k-1]=4k-3$ and $E_s[2k-2]-E_s[2k-3]=4k-7$ from (4.11), and use these together with (4.10) to get

$$\begin{aligned} & E_s[2k-1]-E_s[2k-4]+E_s[2k-1]-E_s[2k] \\ = \quad & (E_s[2k-2]+2)-(E_s[2k-3]-2)-(E_s[2k]-E_s[2k-1]) \\ = \quad & 2+(4k-7)+2-(4k-3) \ = \ 0. \end{aligned} \tag{4.44}$$

In the third and last subcase, to finally prove (4.38d), we use (4.39), (4.9), (4.43), (4.27), (4.44) and (4.10) to obtain

$$\begin{aligned} & 2F_s[2k-2]-E_s[1]-E_s[4k-1] \\ = \quad & (E_s[2k-1]+E_s[4k-2]+1)-E_s[4k-1] \\ = \quad & 2E_s[2k-1]+E_s[2k]+E_s[2k-3]-2E_s[4k-5] \\ \leq \quad & 2E_s[2k-1]+E_s[2k]+E_s[2k-3]-2(E_s[2k]+E_s[2k-4]+2) \\ = \quad & E_s[2k-3]-E_s[2k-4]-4 \ = \ -2 \ < \ 0. \end{aligned}$$

This completes the proof of inequality (4.38d). Summarizing, E_s and F_s together form a 1-D Euclidean representation of the preferences of v_r.

4.10.4 The exceptional case with even r

In this section, we consider the last remaining case of even r, where $r=2k$ and $s \neq 2k$. In order to prove (4.31a) and (4.31b) for the preference orders in (4.5b), it is sufficient to establish the following three inequalities for the turning points.

$$2F_s[2k] > E_s[4k-2]+E_s[4k-3], \tag{4.45a}$$

$$2F_s[2k] < E_s[2]+E_s[4k-1], \tag{4.45b}$$

$$2F_s[2k] > E_s[4k]+E_s[1]. \tag{4.45c}$$

For the common left-hand side of (4.45a)–(4.45c), the definition of $F_s[2k]$ in (4.19), and (4.15) with $s \neq 2k$ yield that

$$\begin{aligned} 2F_s[2k] &= \frac{1}{2}\left(E_s[1]+E_s[2]+E_s[4k-1]+E_s[4k]\right) \\ &= E_s[4k]+E_s[1]+1 \;=\; E_s[4k-1]+E_s[2]-1. \end{aligned} \tag{4.46}$$

For (4.45a), we use (4.46) and (4.14) with $s \neq 2k$ to obtain

$$\begin{aligned} &2F_s[2k]-E_s[4k-2]-E_s[4k-3] \\ &= \quad (E_s[4k-1]+E_s[2]-1)-E_s[4k-2]-E_s[4k-3] \\ &= \quad E_s[2]-E_s[4k-3]-1+(E_s[4k-3]-E_s[2]+2) \;=\; 1 \;>\; 0. \end{aligned}$$

For (4.45b), using (4.46) we get

$$\begin{aligned} &2F_s[2k]-E_s[2]-E_s[4k-1] \\ &= \quad (E_s[4k-1]+E_s[2]-1)-E_s[2]-E_s[4k-1] \;=\; -1 \;<\; 0. \end{aligned}$$

For (4.45c), using (4.46) we obtain

$$\begin{aligned} &2F_s[2k]-E_s[4k]+E_s[1] \\ &= \quad (E_s[4k]+E_s[1]+1)-E_s[4k]-E_s[1] \;=\; 1 \;>\; 0. \end{aligned}$$

This settles the last case. The proof of Lemma 4.8 and with it the proof of Theorem 4.3 are finally completed.

4.11 Other profiles that are not 1-D Euclidean

We have just shown that one-dimensional Euclidean preferences cannot be characterized by finitely many forbidden substructures. We achieved this by providing an infinite sequence of $2k \times 4k$ profiles which fail to be 1-D Euclidean, but deleting any arbitrary voter makes them 1-D Euclidean. This naturally leads to the following question: *"Is this infinite sequence of profiles, together with the four forbidden* *substructures for the single-peakedness and single-crossingness, sufficient to characterize one-dimensional Euclideanness?"* The answer to this question is "no". For instance, the 3×6 profile shown in Example 4.1 is a forbidden substructure for 1-D Euclideanness, but it is not part of the sequence of profiles we constructed. In fact, any forbidden substructure with an odd number of voters cannot be constructed in that way. But how did we find the 3×6 profile? To answer this, let us take a closer look at the profiles constructed in Section 4.6. We observe that in any 1-D Euclidean embedding, the distances of two consecutive alternatives of such profiles display a so-called cyclic relation: (4.8a)–(4.8c), whose existence precludes 1-D Euclideanness. Indeed, we can construct an infinite sequence of non-1-D Euclidean $n \times 2n$ profiles using the cyclic relations discussed at the end of Section 4.7. We consider n voters $v_1, v_2, \ldots, v_n$ and $2n$ alternatives $1, 2, \ldots, 2n$. We assume that the single-peaked orders of the alternatives are $\langle 1, 2, \ldots, 2n \rangle$ and its reverse. First, we observe that any cyclic relation of the distances between neighboring alternatives $2i-1$ and $2i$, $1 \le i \le n$, in the single-peaked order can be considered as a permutation over the numbers $1, 2, \ldots, n$. For instance, the cyclic relation of the 8×16 profile $\mathcal{P}$ in Example 4.3 is

$$
\begin{aligned}
E[2]-E[1] < E[10]-E[9] < E[4]-E[3] < E[12]-E[11] < E[6]-E[5] \\
< E[14]-E[13] < E[8]-E[7] < E[16]-E[15] < E[2]-E[1].
\end{aligned}
$$

It corresponds to the permutation $\langle 1,5,2,6,3,7,4,8 \rangle$. We have shown that this profile $\mathcal{P}$ is not 1-D Euclidean (Lemma 4.6). In fact, we can show that every profile with a unique single-peaked order and with a cyclic relation with respect to this order is non-1-D Euclidean. But how do we obtain this profile given a permutation that corresponds to a cyclic relation?

In the remainder of this section, we show how to construct an infinite sequence of non-1-D Euclidean profiles for a given permutation $\langle \sigma(1), \sigma(2), \ldots, \sigma(n) \rangle$ of the numbers $1, 2, \ldots, n$. For each index i with $1 \le i \le n$, let $\min(i)$ and $\max(i)$ denote the minimum resp. maximum of the two numbers $\sigma(i)$ and $\sigma(i+1 \bmod n)$. Just as in Section 4.6, we define three preference pieces R_i, S_i, T_i.

$$
\begin{aligned}
R_i &:= 2\max(i)-2 \succ 2\max(i)-3 \succ \ldots \succ 2\min(i)+1 \\
S_i &:= 2\min(i)-2 \succ 2\min(i)-3 \succ \ldots \succ 1 \\
T_i &:= 2\max(i)+1 \succ 2\max(i)+2 \succ \ldots \succ 2n
\end{aligned}
$$

Note that for every $i = 1, \ldots, n$, the pieces X_i, Y_i, Z_i cover contiguous intervals of respectively $2\max(i) - 2\min(i) - 2$, $2\min(i) - 2$ and $2n - 2\max(i)$ alternatives. Hence, they jointly cover $2n-4$ of the alternatives, and only the four alternatives $2\min(i)-1, 2\min(i), 2\max(i)-1, 2\max(i)$ remain uncovered. We set

$$U_i := \begin{cases} 2\min(i) \succ 2\max(i)-1 \succ 2\max(i) \succ 2\min(i)-1 & \text{if } \min(i) \neq \sigma(i) \\ 2\max(i)-1 \succ 2\min(i) \succ 2\min(i)-1 \succ 2\max(i) & \text{if } \min(i) = \sigma(i). \end{cases} \quad (4.47)$$

As to the preference orders of the voters, for $1 \le i \le n$, voter v_i has the following preferences:

$$v_i: \quad R_i \succ U_i \succ S_i \succ T_i. \quad (4.48)$$

Using a similar reasoning as in the proof of Lemma 4.6, we can show that the profile with voters $v_1, v_2, \ldots, v_{2n}$ whose preference orders are given by (4.48) is single-peaked and single-crossing, but not 1-D Euclidean. The 3×6 profile in Example 4.1 was constructed using this approach for alternatives $a = 1, b = 2, c = 3, d = 4, e = 5, f = 6$ and for the permutation $\langle 1,2,3 \rangle$. Furthermore, the profiles in Section 4.6 were constructed using the permutation $\langle 1, k+1, 2, k+2, \ldots, k, 2k \rangle$.

We conclude this section by conjecturing that the profile we constructed is minimal non-1-D Euclidean with respect to voter deletion (see Property (b) of Theorem 4.3). One possible approach to proving this conjecture would require, similarly to the one for the profiles constructed in Section 4.6, for each voter v_i, a 1-D Euclidean representation of the profile without this voter v_i. Such representations are, however, not easy to obtain.

4.12 Concluding remarks

We have shown that one-dimensional Euclidean preference profiles cannot be characterized in terms of finitely many forbidden substructures. This is similar to interval graphs, which also cannot be characterized by finitely many forbidden substructures. For interval graphs, however, we have a full understanding of all the forbidden substructures that are *minimal* with respect to vertex deletion; see Lekkerkerker and Boland [LB62]. In a similar vein, it would be interesting to determine all (infinitely many) forbidden substructures for one-dimensional Euclidean preferences that are minimal with respect to deletion of voters or alternatives. At the moment, we only know how to construct non-1-D Euclidean, but single-peaked and single-crossing profiles using our cyclic construction (Section 4.11). We can neither show that such profiles are minimal with respect to voter deletion, nor can we show that they are sufficient to characterize the one-dimensional Euclidean property.

As for general d-dimensional Euclidean preference profiles, we feel that the situation should be similar to the one-dimensional case: we conjecture that for each fixed value of $d \geq 2$, there will be no characterization of d-dimensional Euclidean profiles through finitely many forbidden substructures. However, we see no realistic way of generalizing our current approach to the higher-dimensional situations. Recently, Peters [Pet16] confirms our conjecture for each fixed $d \geq 2$. We remind the reader that in a d-dimensional Euclidean preference profile, the voters and alternatives are embedded in d-dimensional Euclidean space, such that small distance corresponds to strong preference; see for instance the work of Bogomolnaia and Laslier [BL07].

Despite the uncertainty of whether higher-dimensional Euclidean properties can be characterized by finitely many forbidden substructures, Bulteau and Chen [BC] study small non-Euclidean profiles for the two-dimensional case. In particular, they show (via Computer program) that any 3×7 profile, that is, a profile with three voters and up to seven alternatives is 2-dimensional Euclidean. The corresponding proof is based on brute-force search trough all possible profiles via computer programs. They also provide a non-2-dimensional Euclidean profile with three voters and 28 alternatives. It would be interesting to know the exact boundary between Euclideanness and non-Euclideanness.

CHAPTER 5

Nearly Structured Preferences

萝卜白菜，各有所爱。
Every man has his hobbyhorse.
Chinese proverb

In the previous two chapters, we dealt with restricted domains of nicely structured profiles, and we showed that single-crossing profiles can be characterized by finitely many forbidden substructures while one-dimensional Euclidean profiles can not. In this chapter, we focus on a slightly different setting where the profiles themselves are not yet nicely structured. We investigate the computational problem of deciding whether they are close to ones that have a nice structure, as for instance single-peaked, single-crossing, value-restricted, or group-separable profiles. We measure this distance by the number of voters or alternatives that have to be deleted to obtain a nicely structured profile. Our results classify the problem variants with respect to their computational complexity, and draw a clear line between computationally tractable (polynomial-time solvable) and computationally intractable (NP-hard) questions.

5.1 Introduction

The area of Social Choice (and in particular the subarea of *Computational* Social Choice) is full of so-called *negative* results. On the one hand, there are many axiomatic impossibility results, and, on the other hand, there are just as many computational intractability results. For instance, the famous impossibility result of Arrow [Arr50] states that any "fair" social welfare function, which satisfies certain desirable properties, is dictatorial; see the end of Section 2.4 for some discussion on this. As another example, Bartholdi III, Tovey, and Trick [BTT89] established that it is computationally intractable (NP-hard) to determine whether some particular

This chapter is based on "Are There Any Nicely Structured Preference Profiles Nearby?" by R. Bredereck, J. Chen, and G. J. Woeginger, *Mathematical Social Sciences* [BCW16] .

alternative is a winner of a profile under a voting rule designed by Lewis Carroll. Most of these negative results hold for general preference profiles where *any* combination of preference orders may occur.

One branch of Social Choice studies *restricted domains* [ASS02, Chapter 3] of preference profiles, where only certain nicely structured combinations of preference orders are admissible. The standard example for this approach are *single-peaked* preference profiles as introduced by Black [Bla48] (also see Section 4.4.1 for more discussion on the single-peaked property).

Single-peakedness implies a number of nice properties, as for instance strategy-proofness of a family of voting rules [Mou80] and the absence of majority cycles [Ina69] (also known as the Condorcet principle; see the discussion on our working example of Chapter 1). Furthermore, Arrow's impossibility result collapses for single-peaked profiles. In a similar spirit (but in the algorithmic branch), Walsh [Wal07], Brandt et al. [Bra+15], and Faliszewski et al. [Fal+11] showed that many electoral bribery, control, and manipulation problems that are NP-hard in the general case become tractable under single-peaked profiles. Besides the single-peaked domain, the literature contains many other *restricted domains* of nicely structured preference profiles (see Section 5.5 for precise mathematical definitions).

- Sen [Sen66] and Sen and Pattanaik [SP69] introduced the domain of *value-restricted* preference profiles which satisfy the following: for every triple of alternatives, one alternative is not preferred most by any individual (*best-restricted* profile), or one is not preferred least by any individual (*worst-restricted* profile), or one is not ranked as the intermediate alternative by any individual (*medium-restricted* profile).

- Inada [Ina64, Ina69] described the domain of *group-separable* preference profiles where the alternatives can be split into two groups such that every voter prefers every alternative in the first group to those in the second group, or prefers every alternative in the second group to those in the first group. Every group-separable profile is also medium-restricted.

- *Single-caved* preference profiles are derived from single-peaked profiles by reversing the preferences of every voter. Sometimes single-caved profiles are also called *single-dipped* [KPS97].

- *Single-crossing* preference profiles go back to the work of Karlin [Kar68] in applied mathematics and the papers of Mirrlees [Mir71] and Roberts [Rob77] on income taxation. We refer the reader to Chapter 3 for more discussion on this profile. There, we also show that recognizing the single-crossing property is polynomial-time solvable. Doignon and Falmagne [DF94] and Elkind, Faliszewski, and Slinko [EFS12] obtained similar polynomial-time solvability results.

Unfortunately, real-world preference profiles are almost never single-peaked, value-restricted, group-separable, single-caved, or single-crossing. Usually there are maverick voters whose preferences are determined for instance by culture, religion, or gender, destroying all nice structures in the preference profile. In a very recent line of research, Faliszewski, Hemaspaandra, and Hemaspaandra [FHH14] searched for a cure against such mavericks, and arrived at *nearly* single-peaked preference profiles: a profile is nearly single-peaked if it is very close to a single-peaked profile. Of course there are many mathematical ways of measuring the closeness of profiles. Natural ways to make a given profile single-peaked are (i) by deletion of voters and (ii) by deletion of alternatives. This leads to the two central problems of our work for a specific property Π and a given number k:

(i) The Π MAVERICK DELETION problem asks whether it is possible to delete at most k *voters* to make a given profile satisfy the Π-property.

(ii) The Π ALTERNATIVE DELETION problem asks whether it is possible to delete at most k *alternatives* to make a given profile satisfy the Π-property.

In this chapter, we consider classical properties Π that can be characterized by finitely many forbidden substructures; the one-dimensional Euclidean representation from Chapter 4 thus does not fall into this category. We have $\Pi \in$ {worst-restricted, medium-restricted, best-restricted, value-restricted, single-peaked, single-caved, single-crossing, group-separable, β-restricted}. We provide the formal definitions of these properties in Section 5.5.

5.2 Results

We investigate the problem of deciding how close (by deletion of maverick voters or by deletion of alternatives) a given preference profile is to having a nice structure (such as being single-peaked, or single-crossing, or group-separable). We focus on the most fundamental definitions of closeness and on the most popular restricted domains. Our results draw a clear line between polynomial-time solvable and NP-hard problems as they classify all considered problem variants with respect to their computational complexity. In most cases, both of our central problems are shown to be NP-complete (with the exceptions of maverick deletion when the specific property Π is single-crossing, and of alternative deletion when Π is either single-peaked or single-caved).[1] We also find that except for the case with the group-separability, a more restricted preference property seems to have a stronger effect on lowering the complexity of at least one of the computational questions. For instance, while the single-crossing property is more restricted than the γ-restriction (every single-crossing profile is γ-restricted, but not all γ-restricted profiles are single-crossing), maverick deletion is polynomial-time solvable for the single-crossing property, but it is NP-hard for the γ-restriction.

All properties studied in this chapter can be characterized by a fixed number of small forbidden substructures. Thus, by branching over all possible voters (resp. alternatives) of each forbidden substructure in the profile, we can solve all our problems in $c^k \cdot |I|^{O(1)}$ time, where c is a constant value, $|I|$ denotes the input size, and k denotes the respective distance measure parameter (that is, either the number of maverick voters or the number of alternatives to delete). Moreover, there are trivial brute-forcing algorithms that try all possible subsets of the voters or the alternatives and run in $2^{O(n+m)}$ time, where n and m denote the number of alternatives and voters. However, subexponential running time algorithms seem unlikely: We will see from the proofs for the NP-hardness results that the reduction from VERTEX COVER is actually a linear-time reduction. By the subexponential time hypothesis and by the sparsification lemma [IPZ01]–which implies that VERTEX COVER cannot be solved within $2^{o(|U|+|E|)}$ time ($|U|$ and $|E|$ are the number of vertices and edges, respectively)–it follows that except SINGLE-CROSSING ALTERNATIVE DELETION, all our NP-complete problems cannot be solved in $2^{o(n+m)} \cdot |I|^{O(1)}$ time. Our results are summarized in Table 5.1.

5.3 Related work

As for different notions of closeness to restricted domains, Erdélyi, Lackner, and Pfandler [ELP13] study various concepts of nearly single-peakedness. Besides deletion of voters and deletion of alternatives, they also study closeness measures that are based on swapping alternatives in the preference orders of some voters, or on introducing additional political axes. Yang and Guo [YG14] study k-*peaked* domains, where every preference order can have at most k peaks (that is, at most k rising streaks that alternate with falling streaks). Cornaz, Galand, and Spanjaard [CGS12, CGS13] introduce a closeness measure, the *width*, for single-peaked, single-crossing, and group-separable profiles which is based on the notion of a clone set [Tid87]. For

[1] The NP-completeness result for SINGLE-PEAKED MAVERICK DELETION was jointly established with Nadja Betzler.

Restriction	Maverick deletion		Alternative deletion	
Single-peaked	NP-complete	(*, Cor 5.5)	$O(n^5 \cdot m)$	(*, ◊)
Single-caved	NP-complete	(*, Cor 5.5)	$O(n^5 \cdot m)$	(*)
Group-separable	NP-complete	(Cor 5.5)	NP-complete	(Cor 5.6)
Single-crossing	$O(n^3 \cdot m^2)$	(Thm 5.12)	NP-complete	(Thm 5.9)
Value-restricted	NP-complete	(Thm 5.1)	NP-complete	(Thm 5.3)
Best-restricted	NP-complete	(Pro 5.2)	NP-complete	(Pro 5.4)
Worst-restricted	NP-complete	(Pro 5.2)	NP-complete	(Pro 5.4)
Medium-restricted	NP-complete	(Pro 5.2)	NP-complete	(Pro 5.4)
β-restricted	NP-complete	(Thm 5.7)	NP-complete	(Thm 5.8)

Table 5.1.: Summary of the complexity results. The variables n and m denote the number of voters and the number of alternatives, respectively. Entries marked by "*" and by "◊" are due to Erdélyi, Lackner, and Pfandler [ELP13] and Przedmojski [Prz16], respectively. Przedmojski [Prz16] improves on Erdélyi, Lackner, and Pfandler's $O(n^5 \cdot m)$ time algorithm for SINGLE-PEAKED ALTERNATIVE DELETION to an $O(n^3 \cdot m)$ time algorithm. The definitions of the respective domain restrictions can be found in Section 5.5.

instance, the *single-peaked width* of a preference profile is the smallest number k such that partitioning all alternatives into disjoint *intervals*, each with size at most $k+1$, and replacing each of these intervals with a single alternative, results in a single-peaked profile. An interval of alternatives is a set of alternatives that appear consecutively (in any order) in the preference orders of all voters.

There are several generalizations of the single-peaked property. For instance, Barberà, Gul, and Ennio [BGE93] introduced the concept of multi-dimensional single-peaked domains. The 1-dimensional special case is equivalent to our single-peaked property. Sui, Francois-Nienaber, and Boutilier [SFB13] studied this concept empirically. They present polynomial-time approximation algorithms (for several optimization goals) of finding multi-dimensional single-peaked profiles and show that their two real-world data sets are far from being single-peaked but are nearly 2-dimensional single-peaked. While Erdélyi, Lackner, and Pfandler [ELP13] and this chapter show that deciding the distance to restricted domains is NP-complete in most cases (with disjoint results), Elkind and Lackner [EL14] presented efficient approximation and fixed-parameter algorithms for deciding the distance to restricted domains studied in this chapter with the exception of the β-restricted property.

Finally, we remark that the closeness concept can also be used to characterize voting rules [Bai87, EFS12, MN08]. The basic idea is to first fix a specific property, for instance, the Condorcet principle, and then to define a closeness measure from a given profile to the "nearest" profile with this specific property such that the relevant voting rule always selects the same winner in both profiles. For instance, the Young rule [You77] takes the subprofile that is closest to being Condorcet consistent by deleting the fewest number of voters and selects the corresponding Condorcet winner as a winner; see Elkind, Faliszewski, and Slinko [EFS12] for more information on this.

For restricted and nearly restricted domains, there are various studies on single-winner determination [Bra+15], on multi-winner determination [BSU13, Sko+15], on control, manipulation, and bribery [Bra+15, ELP15, Fal+11, FHH14] (also see Section 6.7 where we study control problems for single-peaked profiles and single-crossing profiles), and on possible/necessary winner problems [Wal07]. Usually, the expectation is that domain restrictions help in lowering the computational complexity of many voting problems. Many publications, however, report that this is not always the case. For instance, Faliszewski, Hemaspaandra, and Hemaspaandra [FHH14] showed that the computational complexity of "controlling approval-based rules" for nearly single-peaked profiles is polynomial-time solvable if the distance to single-peaked is a constant, and thus, coincides with the one for single-peaked profiles, whereas the computational complexity of "manipulating the veto rule" for nearly single-peaked profiles is still NP-complete and thus, coincides with the one for unrestricted profiles. In Section 6.7, we show that for some special case, the combinatorial control problem remains intractable for single-peaked profiles while it becomes polynomial-time solvable for single-crossing profiles.

5.4 Chapter outline

Section 5.5 summarizes all additional basic definitions and notations. Our results are presented in Sections 5.6 to 5.8:

Section 5.6 presents results for the value-restricted, best-restricted, worst-restricted, and medium-restricted properties. All results are NP-completeness results and are obtained through reduction from the NP-complete VERTEX COVER problem (see the beginning of Section 5.6 for the definition).

Section 5.7 shows results for single-peakedness, single-cavedness, and group-separability. In addition, this section shows results for the β-restricted property, a necessary condition for group-separability. Again, all results are NP-completeness results and are obtained through reduction from VERTEX COVER.

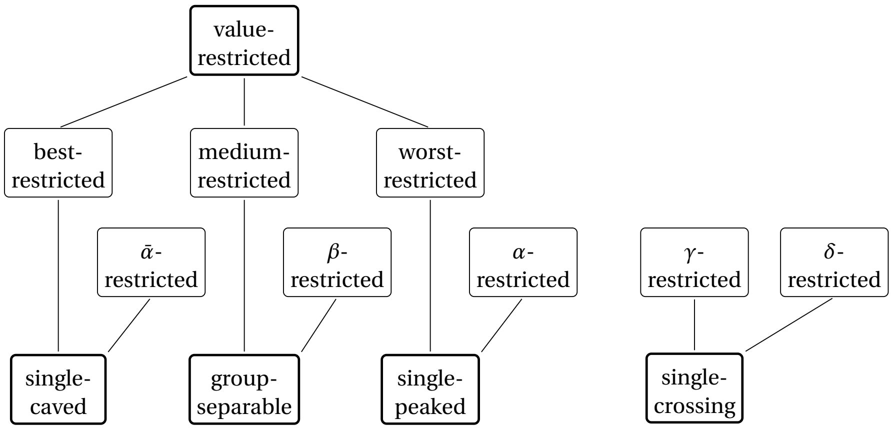


Figure 5.1.: Hasse diagram of the relations between the different properties. An edge between two properties means that a profile with the property in the lower tier implies the property in the upper tier. For instance, there is an edge between "value-restricted" and "best-restricted", because a best-restricted profile is also value-restricted. If there is no edge between two properties, then they are not comparable to each other.

Section 5.8 shows that achieving the single-crossing property by deleting as few alternatives as possible is NP-hard; the reduction is from the NP-complete MAXIMUM 2-SATISFIABILITY problem (see the beginning of Section 5.8 for the definition), and shows that finding a single-crossing profile with the largest voter set is polynomial-time solvable; this is done by reducing the problem to finding a longest path in a directed acyclic graph.

We conclude with some future research directions in Section 5.9.

5.5 Preliminaries and basic notations

In this section we review some preference structures with special properties studied in the literature [BH11, Ina64, Ina69, Sen66, SP69] and in Chapter 3. The main use of these structures is to characterize nicely structured profiles such as single-peaked or single-crossing preferences. Recall that we use *configurations* to denote such preference structures. We use them to characterize some properties of the preference profiles and we illustrate the relation between the respective properties in Figure 5.1.

5.5.1 Value-restriction

The first three configurations [BH11] describe profiles with three alternatives where each alternative is in the best, medium, or worst position in some voter's preference order.

Definition 5.1 (Best-diverse configuration)**.** A profile with three voters v_1, v_2, v_3, and three distinct alternatives a, b, c is a *best-diverse configuration* if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \ a \succ \{b, c\},\\
&\text{voter } v_2: \ b \succ \{a, c\},\\
&\text{voter } v_3: \ c \succ \{a, b\}.
\end{aligned}$$

Definition 5.2 (Medium-diverse configuration)**.** A profile with three voters v_1, v_2, v_3, and three distinct alternatives a, b, c is a *medium-diverse configuration* if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \ b \succ a \succ c \ \text{ or } \ c \succ a \succ b,\\
&\text{voter } v_2: \ a \succ b \succ c \ \text{ or } \ c \succ b \succ a,\\
&\text{voter } v_3: \ a \succ c \succ b \ \text{ or } \ b \succ c \succ a.
\end{aligned}$$

Definition 5.3 (Worst-diverse configuration)**.** A profile with three voters v_1, v_2, v_3 and three distinct alternatives a,b,c is a *worst-diverse configuration* if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \ \{b, c\} \succ a,\\
&\text{voter } v_2: \ \{a, c\} \succ b,\\
&\text{voter } v_3: \ \{a, b\} \succ c.
\end{aligned}$$

We use these three configurations to characterize several restricted domains: A profile is *best-restricted* (resp. *medium-restricted*, *worst-restricted*) *with respect to a triple of alternatives* if it contains no three voters that form a best-diverse configuration (resp. a medium-diverse configuration, a worst-diverse configuration) with respect to this triple. A profile is *best-restricted* (resp. *medium-restricted*, *worst-restricted*) if it is best-restricted (resp. medium-restricted, worst-restricted) with respect to every possible triple of alternatives.

A profile is *value-restricted* [Sen66] if for every triple T of alternatives, it is best-restricted, medium-restricted, or worst-restricted with respect to T. In other words, a profile is *not* value-restricted if and only if it contains a triple of alternatives and three voters v_1, v_2, v_3 that form a best-diverse configuration, a medium-diverse configuration, and a worst-diverse configuration with respect to this triple.

Definition 5.4 (Cyclic configuration)**.** A profile with three voters v_1, v_2, v_3 and three distinct alternatives a,b,c is a *cyclic configuration* if it satisfies the following:

voter v_1: $a \succ b \succ c$,
voter v_2: $b \succ c \succ a$,
voter v_3: $c \succ a \succ b$.

Value-restricted profiles with pairwise different preference orders are also known as *acyclic domain of linear orders*. Many research groups [AJ84, GR08b, Mon09, PS15] investigated maximal acyclic domains for a given number m of alternatives, where an acyclic domain is *maximal* if adding any new linear order destroys the value-restricted property.

5.5.2 Single-peakedness and single-cavedness

Given a set A of alternatives and a linear order L over A, we say that a voter v is *single-peaked* with respect to L if his preference along L is always strictly increasing, always strictly decreasing, or first strictly increasing and then strictly decreasing. Formally, a voter v is single-peaked with respect to L if for each three distinct alternatives a, b, $c \in A$, it holds that

$$(a \succ_L b \succ_L c \text{ or } c \succ_L b \succ_L a) \text{ implies that if } a \succ_v b, \text{ then } b \succ_v c.$$

A profile with the alternative set A is *single-peaked* if there is a linear order over A such that every voter is single-peaked with respect to this order. Single-peaked profiles are necessarily worst-restricted. To see this, we observe that in a profile with at least three alternatives, the alternative that is ranked last by at least one voter must not be placed between the other two along any single-peaked order. But then, none of the alternatives a, b, and c from a worst-diverse configuration can be placed between the other two in any single-peaked order. Thus, a profile with worst-diverse configurations cannot be single-peaked.

To fully characterize the single-peaked domain, we need the following configuration.

Definition 5.5 (α-configuration)**.** A profile with two voters v_1 and v_2, and four distinct alternatives a, b, c, d is an α-configuration if it satisfies the following:

voter v_1: $\{a,d\} \succ b \succ c$,
voter v_2: $\{c,d\} \succ b \succ a$.

The α-configuration describes a situation where two voters have opposite opinions on the order of three alternatives a, b and c but agree that a fourth alternative d is "better" than the one ranked in the middle. A profile with this configuration is not single-peaked as we must put alternatives b and d between alternatives a and c, but then voter v_1 prevents us from putting b next to a and voter v_2 prevents us from putting b next to c.

A profile is *single-peaked* if and only if it contains neither worst-diverse configurations nor α-configurations [BH11]. Since reversing the preference orders of a single-peaked profile results in a single-caved one, an analogous characterization of single-caved profiles follows. A profile is *single-caved* if and only if it contains neither best-diverse configurations nor $\bar{\alpha}$-configurations where an $\bar{\alpha}$-configuration is an α-configuration with both partial preference orders being inverted:

Definition 5.6 ($\bar{\alpha}$-configuration)**.** A profile with two voters v_1 and v_2, and four distinct alternatives a, b, c, d is an $\bar{\alpha}$-configuration if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \; a \succ b \succ \{c,d\},\\
&\text{voter } v_2: \; c \succ b \succ \{a,d\}.
\end{aligned}$$

5.5.3 Group-separability

Given a profile with A being the set of alternatives, the *group-separable* property requires that every size-at-least-three subset $A' \subseteq A$ can be partitioned into two disjoint non-empty subsets A'_1 and A'_2 such that for each voter v_i, either $A'_1 \succ_i A'_2$ or $A'_2 \succ_i A'_1$ holds. One can verify that group-separable profiles are necessarily medium-restricted. Ballester and Haeringer [BH11] characterized the group-separable property using the following configuration.

Definition 5.7 (β-configuration)**.**
A profile with two voters v_1 and v_2 and four distinct alternatives a, b, c, d is a β-configuration if it satisfies the following:

$$\begin{aligned}
&\text{voter } v_1: \; a \succ b \succ c \succ d,\\
&\text{voter } v_2: \; b \succ d \succ a \succ c.
\end{aligned}$$

The β-configuration describes a situation where the most preferred alternative and the least preferred alternative of voter v_1 are a and d which are different from the ones of voter v_2: b and c. Both voters agree that b is better than c, but disagree on whether d is better than a. This profile is not group-separable: We can not partition $\{a,b,c,d\}$ into one singleton and one set of three alternatives as each alternative is ranked in the middle once, but neither can we partition them into two sets each of size two since voter v_1 prevents us from putting alternatives a and c or alternatives a

and d together and voter v_2 prevents us from putting alternatives a and b together. Profiles without β-configurations are called *β-restricted* [BH11].

A profile is *group-separable* if and only if it contains neither medium-diverse configurations nor β-configurations [BH11].

5.5.4 Single-crossingness

The single-crossing property describes the existence of a "natural" linear order of the voters. A preference profile is *single-crossing* if there exists a *single-crossing order* of the voters, that is, a linear order L of the voters, such that each pair of alternatives separates L into two sub-orders where in each sub-order, all voters agree on the relative order of this pair. Formally, this means that for each pair of alternatives a and b such that the first voter along the order L prefers a to b and for each two voters v, v' with $v \succ_L v'$,

$$b \succ_v a \text{ implies } b \succ_{v'} a.$$

In Chapter 3, we define the single-crossing property (Definition 3.1) and introduce two configurations, γ-configurations and δ-configurations (Examples 3.4 and 3.5) and show that a profile is *single-crossing* if and only if it contains neither γ-configurations nor δ-configurations (Theorem 3.2).

5.5.5 Two central problems

As already discussed before, two natural ways of measuring the closeness of profiles to some restricted domains are to count the number of voters resp. alternatives which have to be deleted to make a profile single-crossing. Hence, for $\Pi \in$ {worst-restricted, medium-restricted, best-restricted, value-restricted, single-peaked, single-caved, single-crossing, group-separable, β-restricted}, we study the following two decision problems: Π MAVERICK DELETION and Π ALTERNATIVE DELETION.

Π MAVERICK DELETION
Input: A profile with n voters and a non-negative integer $k \leq n$.
Question: Can we delete at most k voters so that the resulting profile satisfies the Π-property?

Π ALTERNATIVE DELETION
Input: A profile with m alternatives and a non-negative integer $k \leq m$.
Question: Can we delete at most k alternatives so that the resulting profile satisfies the Π-property?

An upper bound on the computational complexity of Π Maverick Deletion and Π Alternative Deletion is easy to see. Both problems are contained in NP for each property Π we study: given a preference profile one can check in polynomial time whether it has property Π since Π is characterized by a finite set of forbidden finite substructures. Thus, in order to show NP-completeness of Π Maverick Deletion and Π Alternative Deletion, we only have to show their NP-hardness.

5.6 Value-restricted properties

In this section, we show NP-hardness for the value-restricted, best-restricted, worst-restricted, and medium-restricted domains, respectively. Notably, we show all these results by reducing from the NP-complete Vertex Cover problem [GJ79].

> Vertex Cover
> **Input:** An undirected graph $G = (U, E)$ and an integer $k \leq |U|$.
> **Question:** Is there a *vertex cover* $U' \subseteq U$ of at most k vertices, that is, $|U'| \leq k$ and $\forall e \in E\colon\ e \cap U' \neq \emptyset$?

In every reduction from Vertex Cover we describe, the vertex cover size k coincides with the maximum number k of voters (resp. alternatives) to delete. Hence, we use the same variable name.

We first deal with the case of maverick voter deletion (Section 5.6.1) and then, with the case of deleting alternatives (Section 5.6.2). In both cases, the general idea is to transform every edge of a given graph into an appropriate forbidden configuration.

5.6.1 Maverick Voter Deletion

Theorem 5.1. Value-Restricted Maverick Deletion *is* NP*-complete.*

Proof. We provide a polynomial-time reduction from Vertex Cover to show NP-hardness. We present an example (see Figure 5.2) for the reduction right after this proof.

Let $(G = (U, E), k)$ denote a Vertex Cover instance with vertex set $U = \{u_1, \ldots, u_r\}$ and edge set $E = \{e_1, \ldots, e_s\}$; without loss of generality we assume that the input graph G is connected and that graph G has at least four vertices, that is, $r \geq 4$, and that $k \leq r - 3$.

The set of alternatives consists of three *edge alternative*s a_j, b_j, and c_j for each edge $e_j \in E$. For each vertex in U, we construct one voter. We define $A := \{a_j, b_j, c_j \mid e_j \in E\}$ and $V := \{v_i \mid u_i \in U\}$. In total, the number m of alternatives is $3s$ and the number n of voters is r.

Every voter prefers $\{a_j, b_j, c_j\}$ to $\{a_{j'}, b_{j'}, c_{j'}\}$ whenever $j < j'$. For each edge e_j with two incident vertices u_i and $u_{i'}$, $i < i'$, and for each non-incident vertex $u_{i''} \notin e_j$, the following holds:

$$\begin{aligned} \text{voter } v_i &: \ c_j \succ a_j \succ b_j, \\ \text{voter } v_{i'} &: \ b_j \succ c_j \succ a_j, \\ \text{voter } v_{i''} &: \ a_j \succ b_j \succ c_j. \end{aligned}$$

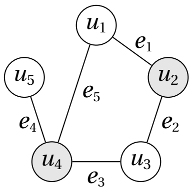
In this way, the two vertex voters that correspond to the vertices in e_j and any voter v_z not in e_j form a cyclic configuration with regard to the three edge alternatives a_j, b_j, and c_j. By the definition of cyclic configurations, this configuration is also best-diverse, medium-diverse, and worst-diverse.

Let the maximum number of voters to delete equal the maximum vertex cover size k. This completes the construction which can be done in linear time.

It remains to show the correctness. In particular, we show that $(G = (U, E), k)$ has a vertex cover of size at most k if and only if the constructed profile can be made value-restricted by deleting at most k voters.

For the "only if" part, suppose that $U' \subseteq U$ with $|U'| \leq k$ is a vertex cover. We show that, after deleting the voters corresponding to the vertices in U' the resulting profile is value-restricted. Suppose for the sake of contradiction that the resulting profile is *not* value-restricted. That is, it still contains a cyclic configuration σ. By the definition of cyclic configurations, for each pair of alternatives x and y in σ, there are two voters, one preferring x to y and the other preferring y to x. Together with the fact that all voters agree on the relative order of two edge alternatives that correspond to different edges, this implies that the three alternatives a_j, b_j, and c_j in σ correspond to the same edge e_j. Furthermore, σ involves two voters corresponding to the incident vertices of e_j, and one other voter, because all voters corresponding to vertices not in e_j have the same ranking $a_j \succ b_j \succ c_j$. Then, edge e_j is not covered by any vertex in U'—a contradiction.

For the "if" part, suppose that the profile becomes value-restricted after the removal of a subset $V' \subseteq V$ of voters with $|V'| \leq k$. That is, no three remaining voters form a cyclic configuration. We show by contradiction that V' corresponds to a vertex cover of graph G. Assume towards a contradiction that an edge e_j is not covered by the vertices corresponding to the voters in V'. Then, the two voters corresponding to the vertices that are incident with edge e_j together with a third voter form a cyclic configuration with regard to the three alternatives a_j, b_j, and c_j—a contradiction. Thus, V' corresponds to a vertex cover of graph G and its size is at most k. □



(a)

voter v_1: $c_1 \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ c_2 \succ a_3 \succ b_3 \succ c_3 \succ a_4 \succ b_4 \succ c_4 \succ c_5 \succ a_5 \succ b_5$
voter v_2: $b_1 \succ c_1 \succ a_1 \succ c_2 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ c_3 \succ a_4 \succ b_4 \succ c_4 \succ a_5 \succ b_5 \succ c_5$
voter v_3: $a_1 \succ b_1 \succ c_1 \succ b_2 \succ c_2 \succ a_2 \succ c_3 \succ a_3 \succ b_3 \succ a_4 \succ b_4 \succ c_4 \succ a_5 \succ b_5 \succ c_5$
voter v_4: $a_1 \succ b_1 \succ c_1 \succ a_2 \succ b_2 \succ c_2 \succ b_3 \succ c_3 \succ a_3 \succ c_4 \succ a_4 \succ b_4 \succ b_5 \succ c_5 \succ a_5$
voter v_5: $a_1 \succ b_1 \succ c_1 \succ a_2 \succ b_2 \succ c_2 \succ a_3 \succ b_3 \succ c_3 \succ b_4 \succ c_4 \succ a_4 \succ a_5 \succ b_5 \succ c_5$

(b)

Figure 5.2.: An illustration of the reduction from VERTEX COVER to VALUE-RESTRICTED MAVERICK DELETION (Theorem 5.1). (a) An undirected graph with five vertices and five edges. The graph has a vertex cover of size 2 (filled in gray). (b) A reduced instance $((A,V), k=2)$ of VALUE-RESTRICTED MAVERICK DELETION, where $A = \{a_i, b_i, c_i \mid 1 \le i \le 5\}$ and $V = \{v_1, v_2, v_3, v_4, v_5\}$. Deleting v_2 and v_4 results in a value-restricted profile. In fact, the resulting profile is also best-restricted, single-peaked (and hence worst-restricted), and group-separable (and hence medium-restricted).

We illustrate our reduction through an example. Figure 5.2(a) depicts an undirected graph with 5 vertices and 4 edges. Vertices u_2 and u_4 form a vertex cover of size two. Figure 5.2(b) shows the reduced instance with 5 voters and $3 \cdot 4 = 12$ alternatives. Deleting voters v_2 and v_4 results in a value-restricted profile which is also best-restricted, single-peaked for instance with respect to the order

$$(b_5, a_5, a_4, b_3, a_3, c_2, b_2, b_1, a_1, c_1, a_2, c_3, b_4, c_4, c_5)$$

(and hence worst-restricted), and group-separable (and hence medium-restricted).

Taking a closer look at the reduction shown in the proof of Theorem 5.1, the constructed profile contains cyclic configurations which are simultaneously best-diverse, worst-diverse, and medium-diverse. It turns out that we can use the same construction to show the following three NP-hardness results with regard to the best-restricted, worst-restricted, and medium-restricted properties.

Proposition 5.2. Π MAVERICK DELETION *is* NP*-complete for every property* $\Pi \in$ $\{$*best-restricted, worst-restricted, medium-restricted*$\}$.

Proof. Let $(G = (U, E), k)$ be a VERTEX COVER instance and let A and V be the set of alternatives and the set of voters that are constructed in the same way as in the proof of Theorem 5.1. Let k be the number of voters to be deleted. As we already observed in that proof, for each edge $e_j \in E$, the two vertex voters that correspond to the vertices in e_j and any other voter v_z form a cyclic configuration, that is, a best-diverse, worst-diverse, and medium-diverse configuration. It remains to show that $(G = (U, E), k)$ has a vertex cover of size at most k if and only if the constructed profile can be made best-restricted (or worst-restricted or medium-restricted) by deleting at most k voters.

For the "if" part, suppose that the profile becomes best-restricted (or worst-restricted or medium-restricted) by deleting a subset $V' \subseteq V$ of voters with $|V'| \le k$. Then, the resulting profile is also value-restricted. Thus, we can use the "if" part in the proof of Theorem 5.1 and obtain that the vertices corresponding to V' form a vertex cover of size at most k.

For the "only if" part, suppose that $U' \subseteq U$ with $|U'| \le k$ is a vertex cover. As in the "only if" part proof of Theorem 5.1, we can show that deleting the voters corresponding to U' results in a best-restricted and a worst-restricted profile.

As for the medium-restricted property, suppose towards a contradiction that after deleting the voters corresponding to the vertices in U' there is still a medium-diverse configuration σ. By the definition of medium-diverse configurations, we know that in σ, each alternative is ranked between the other two by one voter. This implies that σ involves three alternatives that correspond to the same edge e_j and involves two voters that correspond to e_j's incident vertices. Thus, e_j is an uncovered edge—a contradiction. □

5.6.2 Alternative Deletion

Next, we consider the case of deleting alternatives. Just as for the voter deletion case, we first show NP-hardness of deciding the distance to value-restricted profiles. Then, we show how to adapt the reduction to also work for deciding the distance to best-restricted, worst-restricted, and medium-restricted profiles, respectively.

Theorem 5.3. VALUE-RESTRICTED ALTERNATIVE DELETION *is* NP*-complete.*

Proof. We reduce from VERTEX COVER to VALUE-RESTRICTED ALTERNATIVE DELETION. Let $(G = (U, E), k)$ be a VERTEX COVER instance with vertex set $U = \{u_1, \ldots, u_r\}$ and edge set $E = \{e_1, \ldots, e_s\}$. The set of alternatives consists of one *vertex alternative* a_j for each vertex u_j in U and of $k+1$ additional *dummy alternatives*. Let A denote the set of all vertex alternatives and let D denote the set of all dummy alternatives. We arbitrarily fix a canonical order $\langle D \rangle$ of D and we set $\langle A \rangle := a_1 \succ a_2 \succ \ldots \succ a_r$. The number m of constructed alternatives is $r + k + 1$.

We introduce a voter v_0 with the canonical preference order $\langle D \rangle \succ \langle A \rangle$. For each edge $e_i = \{u_j, u_{j'}\}$ with $j < j'$, we introduce two *edge voters* v_{2i-1} and v_{2i} with preference orders

$$\begin{aligned}&\text{voter } v_{2i-1}: && a_j \succ a_{j'} \succ \langle D\rangle \succ \langle A \setminus \{a_j, a_{j'}\}\rangle,\\ &\text{voter } v_{2i} \quad : && a_{j'} \succ \langle D\rangle \succ \langle A \setminus \{a_{j'}\}\rangle.\end{aligned}$$

Together with voter v_0, the two voters v_{2i-1} and v_{2i} form a cyclic configuration with respect to the two vertex alternatives a_j, $a_{j'}$, and an arbitrary dummy alternative from D. Let V denote the set of all voters. In total, the number n of constructed voters is $2s+1$. To finalize the construction, let the maximum number of alternatives to delete equal the maximum vertex cover size k.

Our reduction runs in linear time. It remains to show that graph G has a vertex cover of size at most k if and only if the constructed profile can be made value-restricted by deleting at most k alternatives.

For the "only if" part, suppose that $U' \subseteq U$ with $|U'| \le k$ is a vertex cover. We show that after deleting the vertex alternatives corresponding to U', the resulting profile is value-restricted. Suppose for the sake of contradiction that the resulting profile is not value-restricted, that is, it contains a cyclic configuration σ. By definition, it must hold that for each pair of alternatives x and y in σ, there are two voters, one preferring x to y and the other preferring y to x. Together with the fact that each voter agrees on the relative order of two distinct dummy alternatives, this implies that σ contains at most one dummy alternative. But if σ contains one dummy alternative $d \in D$, then there is a voter with preferences $a_j \succ a_{j'} \succ d$ where a_j, $a_{j'} \in A$, which means that edge $\{u_j, u_{j'}\}$ is not covered by U'. Hence, σ contains no dummy alternative. This means that σ contains three vertex alternatives a_j, $a_{j'}$, and $a_{j''}$ with $j < j' < j''$ and, by the definition of cyclic configurations, σ involves three voters with preferences $\{a_j, a_{j'}\} \succ a_{j''}$, $\{a_j, a_{j''}\} \succ a_{j'}$, and $\{a_{j'}, a_{j''}\} \succ a_j$, respectively. However, the last preference implies that $\{u_{j'}, u_{j''}\}$ is an edge which is not covered by U'—a contradiction.

For the "if" part, suppose that the constructed profile is a yes-instance of VALUE-RESTRICTED ALTERNATIVE DELETION. Let $A' \subseteq A \cup D$ be the set of deleted vertex alternatives with $|A'| \le k$. We show that the vertex set U' corresponding to A' forms a vertex cover of graph G and has size at most k. Clearly, $|U'| \le k$. Assume towards a contradiction that $e_i = \{u_j, u_{j'}\}$, $j < j'$, is not covered by U'. Since $|D| > k$, at least one dummy alternative d is not deleted. Then, v_0 and the two edge voters v_{2i-1} and v_{2i} form a cyclic configuration with regard to a_j, $a_{j'}$, d—a contradiction. □

Using the same construction as in the last proof, we can show that achieving best-restriction, worst-restriction, or medium-restriction via deleting the fewest number of alternatives is intractable.

Proposition 5.4. Π ALTERNATIVE DELETION *is* NP*-complete for every property* $\Pi \in$ {*best-restricted, worst-restricted, medium-restricted*}.

Proof. Let $((U,E),k)$ denote a VERTEX COVER instance with vertex set $U = \{u_1,\ldots,u_r\}$ and edge set $E = \{e_1,\ldots,e_s\}$. Let A, D, and V be the sets constructed in the same way as in the proof of Theorem 5.3.

It remains to show that $((U,E),k)$ has a vertex cover of size at most k if and only if the constructed profile can be made best-restricted (or worst-restricted or medium-restricted) by deleting at most k alternatives.

For the "if" part, suppose that the profile becomes best-restricted (or worst-restricted or medium-restricted) after deleting a set $A' \subseteq A \cup D$ of at most k alternatives. Then, the resulting profile is also value-restricted. Thus, we can use the "if" part in the proof of Theorem 5.3 and obtain that the vertex set corresponding to A' forms a vertex set of size at most k.

For the "only if" part, suppose that $U' \subseteq U$ with $|U'| \le k$ is a vertex cover. Just as in the "only if" part proof of Theorem 5.3, we can show that deleting the alternatives corresponding to U' results in a best-restricted and a worst-restricted profile.

As for the medium-restricted property, suppose for the sake of contradiction that the resulting profile is not medium-restricted, that is, it contains a medium-diverse configuration σ. Since all voters rank $\langle D\rangle$ and since no voter ranks $d \succ a_j \succ d'$ with d, $d' \in D$ and $a_j \in A$, configuration σ contains at most one dummy alternative. Now, if σ involves one dummy alternative $d \in D$ and two vertex alternatives a_j, $a_{j'} \in A$ with $j < j'$, then the voter ranking $a_{j'}$ between a_j and d must rank $a_j \succ a_{j'} \succ d$. But this means that edge $\{u_j, u_{j'}\}$ is uncovered—a contradiction. Hence, σ contains no dummy alternative. This means that σ involves three vertex alternatives a_j, $a_{j'}$, $a_{j''}$ with $j < j' < j''$. By the definition of medium-diverse configurations, σ must contain a voter that ranks $a_{j''}$ between a_j and $a_{j'}$, that is, a voter ranks either $a_j \succ a_{j''} \succ a_{j'}$ or $a_{j'} \succ a_{j''} \succ a_j$. This, however, implies that either edge $\{u_j, u_{j''}\}$ or edge $\{u_{j'}, u_{j''}\}$ is uncovered—a contradiction. $\square$

5.7 Single-peaked, single-caved, and group-separable properties

Since single-peaked, group-separable, and single-caved profiles are necessarily worst-restricted, medium-restricted, and best-restricted, respectively, it seems reasonable to expect that the intractability result (Proposition 5.2) transfers. Indeed, we can show that this immediately follows from the proofs of Proposition 5.2 (and hence of Theorem 5.1) because the profile constructed in the NP-hardness reduction contains neither α-configurations nor β-configurations nor $\bar{\alpha}$-configurations.

Corollary 5.5. Π MAVERICK DELETION *is* NP*-complete for every property* $\Pi \in \{$*single-peaked, single-caved, group-separable*$\}$.

Proof. First, the profile constructed in the proof of Proposition 5.2 does not contain any three alternatives x, y, and z such that there is one voter with $x \succ y \succ z$ and one voter with $z \succ y \succ x$. Thus, the profile contains neither α-configurations nor $\bar{\alpha}$-configurations.

Second, one can partition every set T of four alternatives into two non-empty subsets T_1 and T_2 such that $T_1 \succ T_2$ holds for each voter because at most three alternatives can correspond to the same edge and all voters have the same ranking over the alternatives that correspond to different edges. However, this is not possible in a β-configuration. Thus, the profile does not contain any α-configuration, $\bar{\alpha}$-configuration, or β-configuration.

As a consequence, the reduction in the proof of Proposition 5.2 also works for SINGLE-PEAKED MAVERICK DELETION, SINGLE-CROSSING MAVERICK DELETION, and GROUP-SEPARABLE MAVERICK DELETION. □

Note that NP-hardness of SINGLE-PEAKED MAVERICK DELETION is already known by a different proof of Erdélyi, Lackner, and Pfandler [ELP13]. However, their proof does not work for Π MAVERICK DELETION with $\Pi \in \{$best-restricted, medium-restricted, worst-restricted, group-separable$\}$.

Just as the result for the maverick deletion, the NP-hardness result of MEDIUM-RESTRICTED ALTERNATIVE DELETION also transfers to the group-separable case. After deleting the alternatives corresponding to a vertex cover, the resulting profile from the proof of Proposition 5.4 does not contain any β-configurations. Thus, the following holds.

Corollary 5.6. GROUP-SEPARABLE ALTERNATIVE DELETION *is* NP*-complete.*

Proof. The profile constructed in the proof of Proposition 5.4 may contain β-configurations, but we show that destroying all medium-diverse configurations by deleting at most k alternatives also destroys all β-configurations. Consider the profile $\mathscr{P}$ after the deletion of the alternatives. Assume towards a contradiction that profile $\mathscr{P}$ contains a β-configuration which involves four alternatives w, x, y, z and two voters v, v' with preferences

$$\text{voter } v\colon w \succ x \succ y \succ z \text{ and voter } v'\colon x \succ z \succ w \succ y.$$

Observe that a β-configuration may contain at most one dummy alternative because no two alternatives appear consecutively in both preference orders of a β-configuration, but all dummy alternatives appear consecutively in all preference orders of the profile $\mathscr{P}$. Furthermore, w, y, and z are vertex alternatives since in $\mathscr{P}$, no voter prefers more than one vertex alternative to a dummy alternative. By the definition of β-configurations, voter v ranks $a_j \succ x \succ a_{j'} \succ a_{j''}$ and voter v' ranks $x \succ a_{j''} \succ a_j \succ a_{j'}$ with a_j, $a_{j'}$, $a_{j''} \in A$. However, the preference order of voter v implies that $j' < j''$ and the preference order of voter v' implies that $j'' < j'$—a contradiction. As consequence, the reduction in the proof of Proposition 5.4 also works for GROUP-SEPARABLE MAVERICK DELETION. □

In order to be group-separable, a preference profile must be medium-restricted and β-restricted. As already shown in Corollary 5.5 and in Corollary 5.6, deleting as few maverick voters (or alternatives) as possible to obtain the group-separable property is NP-hard. Alternatively, we can also derive this intractability result from the following two theorems.

Theorem 5.7. *β-RESTRICTED MAVERICK DELETION is NP-complete.*

Proof. We reduce from VERTEX COVER to show NP-hardness. Let $(G = (U,E),k)$ denote a VERTEX COVER instance with vertex set $U = \{u_1,\ldots,u_r\}$ and edge set $E = \{e_1,\ldots,e_s\}$; without loss of generality, we assume that the input graph G is connected and that it has at least four vertices, that is, $r \geq 4$. The set of alternatives consists of four *edge alternatives* a_j, b_j, c_j, d_j for each edge $e_j \in E$. For each vertex in U, we construct one voter. That is, we define $A := \{a_j, b_j, c_j, d_j \mid e_j \in E\}$ and $V := \{v_i \mid u_i \in U\}$. In total, the number m of alternatives is $4s$ and the number n of voters is r.

Now we specify the preference order of each voter. Every voter prefers $\{a_j, b_j, c_j, d_j\}$ to $\{a_{j'}, b_{j'}, c_{j'}, d_{j'}\}$ whenever $j < j'$. For each edge e_j with two incident vertices u_i and $u_{i'}$, $i < i'$, and for each non-incident vertex $u_{i''} \notin e_j$, the following holds:

$$\begin{aligned}
\text{voter } v_i &: \; a_j \succ b_j \succ c_j \succ d_j,\\
\text{voter } v_{i'} &: \; b_j \succ d_j \succ a_j \succ c_j,\\
\text{voter } v_{i''} &: \; d_j \succ a_j \succ b_j \succ c_j.
\end{aligned}$$

In this way, any β-configuration regarding alternatives a_j, b_j, c_j, d_j must involve voters v_i and $v_{i'}$. To finalize the construction, let the maximum number of voters to delete equal the maximum vertex cover size k.

Clearly, the whole construction runs in linear time. It remains to show that $(G = (U,E),k)$ has a vertex cover of size at most k if and only if the constructed profile can be made β-restricted by deleting at most k voters.

For the "only if" part, suppose that $U' \subseteq U$ with $|U'| \leq k$ is a vertex cover. We show that after deleting the voters corresponding to the vertices in U' the resulting profile is β-restricted. Suppose for the sake of contradiction that the resulting profile is *not* β-restricted. That is, it still contains a β-configuration σ. Since all voters prefer $\{a_j, b_j, c_j, d_j\}$ to $\{a_{j'}, b_{j'}, c_{j''}, d_{j'''}\}$ whenever $j < j'$, the profile restricted to every four alternatives that correspond to at least two edges is group-separable. But since σ is not group-separable, σ involves four alternatives a_j, b_j, c_j, and d_j that correspond to a single edge e_j. As already observed, any β-configuration regarding the four alternatives a_j, b_j, c_j, d_j must involve voters v_i and $v_{i'}$ that correspond to both endpoints of e_j. Then, edge e_j is not covered by any vertex in U'—a contradiction.

For the "if" part, suppose that the profile becomes β-restricted by deleting a subset $V' \subseteq V$ of voters with $|V'| \le k$. That is, no two voters form a β-configuration. We show by contradiction that V' corresponds to a vertex cover of graph G and has size at most k. Clearly, $|V'| \le k$. Assume towards a contradiction that an edge e_j is not covered by the vertices corresponding to the voters in V'. Then, the two voters corresponding to the vertices that are incident with edge e_j form a β-configuration with regard to a_j, b_j, c_j, and d_j—a contradiction. Thus, voter set V' corresponds to a vertex cover of graph G and its size is at most k. □

Theorem 5.8. β-RESTRICTED ALTERNATIVE DELETION *is* NP*-complete.*

Proof. We reduce from VERTEX COVER to β-RESTRICTED ALTERNATIVE DELETION. Let $(G = (U, E), k)$ denote a VERTEX COVER instance with vertex set $U = \{u_1, \dots, u_r\}$ and edge set $E = \{e_1, \dots, e_s\}$; without loss of generality we assume that the input graph G is connected and that $r \ge k + 2$. The set of alternatives consists of one *vertex alternative* a_j and one *dummy alternative* d_j for each vertex u_j in U. Let A denote the set of all vertex alternatives and let D denote the set of all dummy alternatives. The number m of constructed alternatives is $2r$.

We fix the canonical order of $A \cup D$ to be

$$\langle A \cup D \rangle := d_1 \succ a_1 \succ d_2 \succ a_2 \succ \ldots \succ d_r \succ a_r.$$

We introduce a voter v_0 with the preference order $\langle A \cup D \rangle$. For each edge $e_i = \{u_j, u_{j'}\}$ with $j < j'$, we introduce one *edge voter* v_i with preference order

$$a_j \succ a_{j'} \succ \langle A \cup D \setminus \{a_j, a_{j'}\} \rangle.$$

Observe that voter v_0 and v_i form a β-configuration with respect to the four alternatives corresponding to the vertices in edge $\{u_j, u_{j'}\}$ with $j < j'$, that is, with respect to a_j, $a_{j'}$, d_j, and $d_{j'}$:

$$\text{voter } v_0: \ \ d_j \succ a_j \succ d_{j'} \succ a_{j'},$$
$$\text{voter } v_i: \ \ a_j \succ a_{j'} \succ d_j \succ d_{j'}.$$

Furthermore, for each pair of alternatives a and b, if there is a voter v preferring a to b, then the following holds:

(i) If neither a nor b is in the first two positions of voter v's preference order, then a and b correspond to two (not necessarily distinct) vertices u_j and $u_{j'}$ with $j \le j'$.

(ii) If a and b correspond to two vertices u_j and $u_{j'}$ with $j > j'$ and if there is a third alternative c such that v prefers c to a, then c and a correspond to two adjacent vertices.

We utilize these two facts several times to show some contradictions.

Let V denote the set of all voters. In total, the number n of constructed voters is $s+1$. To finalize the construction, let the maximum number of alternatives to delete equal the maximum vertex cover size k.

Our construction runs in linear time. It remains to show that $(G=(U,E),k)$ has a vertex cover of size at most k if and only if the constructed profile can be made β-restricted by deleting at most k alternatives.

For the "only if" part, suppose that $U' \subseteq U$ with $|U'| \le k$ is a vertex cover of graph G. We show that after deleting the vertex alternatives corresponding to U', denoted by A', the resulting profile is β-restricted. Suppose for the sake of contradiction that the resulting profile still contains a β-configuration σ with regard to four alternatives w, x, y, z and two voters v, v' with preferences

$$\text{voter } v\colon w \succ x \succ y \succ z \;\text{ and }\; \text{voter } v'\colon x \succ z \succ w \succ y.$$

Let w, x, y, and z correspond to four non-deleted vertices u_j, $u_{j'}$, $u_{j''}$, and $u_{j'''}$, respectively. By Property (i), the preference order of voter v implies $j'' \le j'''$ (note that voter v ranks neither y nor z in the first two positions) and the preference order of voter v' implies that $j \le j''$ (note that voter v' ranks neither w nor y in the first two positions). Since x, y, and z correspond to at least two distinct vertices, it follows that $j < j'''$. Since voter v' ranks $x \succ z \succ w \succ y$, by Property (ii) the inequality $j < j'''$ implies that $u_{j'}$ and $u_{j'''}$ are adjacent—a contradiction to U' being a vertex cover. Indeed, the resulting profile is group-separable. To see this, note that any size-at-least-three subset $T \subseteq (A \cup D) \setminus A'$ of alternatives can be partitioned into two non-empty subsets $\{a\}$ and $T \setminus \{a\}$ with a being the last alternative in the canonical order restricted to the alternatives in set T.

For the "if" part, suppose that the constructed profile is a yes-instance of β-RESTRICTED ALTERNATIVE DELETION. Let $A' \subseteq A \cup D$ be the set of deleted alternatives with $|A'| \leq k$. Consider the vertex set U' containing all vertices corresponding to a vertex alternative or to a dummy alternative in A', that is, $U' := \{u_j \mid a_j \in A' \vee d_j \in A'\}$. Obviously, $|U'| \leq k$. We show that set U' is a vertex cover. Suppose towards a contradiction that there is an uncovered edge $e_i = \{u_j, u_{j'}\}$ with $j < j'$. By the definition of U', we have that $A' \cap \{d_j, a_j, d_{j'}, a_{j'}\} = \emptyset$. Then, voters v_0 and v_i form a β-configuration with respect to the four alternatives d_j, a_j, $d_{j'}$, and $a_{j'}$—a contradiction. □

5.8 Single-crossing properties

In this section, we show that for the single-crossing property, the alternative deletion problem is NP-hard while the maverick deletion problem is polynomial-time solvable. The NP-hardness proof is based on the NP-complete MAXIMUM 2-SATISFIABILITY (MAX2SAT) problem [GJ79].

> MAXIMUM 2-SATISFIABILITY (MAX2SAT)
> **Input:** A set U of Boolean variables, a collection C of size-two clauses over U and a positive integer h.
> **Question:** Is there a truth assignment for U which satisfy at least h clauses in C?

Theorem 5.9. SINGLE-CROSSING ALTERNATIVE DELETION *is* NP*-complete.*

Proof. For the NP-hardness result we reduce from MAX2SAT [GJ79]. We provide an example for the reduction (Table 5.2) right after this proof.

Let (U, C, h) be a MAX2SAT instance with variable set $U = \{x_1, \ldots, x_r\}$ and clause set $C = \{c_1, \ldots, c_s\}$. We construct two sets O and $\overline{O}$ of *dummy alternatives* with $|O| = |\overline{O}| = 2(r \cdot s + r + s) + 1$. For each variable $x_i \in U$, we construct two sets X_i and $\overline{X_i}$ of *variable alternative*s with $|X_i| = |\overline{X_i}| = s + 1$. We say that X_i corresponds to x_i and that $\overline{X_i}$ corresponds to $\overline{x_i}$. The canonical orders $\langle O \rangle$, $\langle \overline{O} \rangle$, $\langle X_i \rangle$, and $\langle \overline{X_i} \rangle$, $i \in \{1, \ldots, r\}$, are arbitrary but fixed. Let X be the union $\bigcup_{i=1}^{r} X_i \cup \overline{X_i}$ of all variable alternatives. The canonical order $\langle X \rangle$ is defined as

$$\langle X \rangle := \langle X_1 \rangle \succ \langle \overline{X_1} \rangle \succ \langle X_2 \rangle \succ \langle \overline{X_2} \rangle \succ \ldots \succ \langle X_r \rangle \succ \langle \overline{X_r} \rangle.$$

For each clause $c_j \in C$, we construct two *clause alternative*s a_j and b_j. Let A denote the set of all clause alternatives. The canonical order $\langle A \rangle$ is defined as

$$\langle A \rangle := a_1 \succ b_1 \succ a_2 \succ b_2 \succ \ldots \succ a_s \succ b_s.$$

The total number m of alternatives is $6(r \cdot s + r + s) + 2$.

We introduce voters and their preference orders such that

(1) deleting all alternatives in X_i corresponds to setting variable x_i to true,

(2) deleting all alternatives in $\overline{X_i}$ corresponds to setting variable x_i to false, and

(3) deleting b_j or a_j corresponds to clause c_j not being satisfied.

We construct two sets V and W of voters with $|V| = 2r$ and $|W| = 4s$. Voter set V consists of two voters v_{2i-1} and v_{2i} for each variable x_i, $1 \le i \le r$. Their preference orders are

$$\begin{aligned}
&\langle O\rangle \succ \langle \overline{O}\rangle \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \ldots \succ \langle X_{i-1}\rangle \succ \langle \overline{X_{i-1}}\rangle \succ \\
&\qquad \langle \overline{X_i}\rangle \succ \langle X_i\rangle \succ \langle X_{i+1}\rangle \succ \langle \overline{X_{i+1}}\rangle \succ \ldots \succ \langle X_r\rangle \succ \langle \overline{X_r}\rangle \succ \langle A\rangle, \\
&\langle \overline{O}\rangle \succ \langle O\rangle \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \ldots \succ \langle X_{i-1}\rangle \succ \langle \overline{X_{i-1}}\rangle \succ \\
&\qquad \langle \overline{X_i}\rangle \succ \langle X_i\rangle \succ \langle X_{i+1}\rangle \succ \langle \overline{X_{i+1}}\rangle \succ \ldots \succ \langle X_r\rangle \succ \langle \overline{X_r}\rangle \succ \langle A\rangle,
\end{aligned}$$

respectively. These two voters together with any other two voters v_ℓ and $v_{\ell'} \in V \setminus \{v_{2i-1}, v_{2i}\}$ with odd number ℓ and even number ℓ' form a δ-configuration with regard to each four alternatives o, $\overline{o}$, x, $\overline{x}$ where $o \in O$, $\overline{o} \in \overline{O}$, $x \in X_i$, $\overline{x} \in \overline{X_i}$:

$$\begin{aligned}
&\text{voter } v_{2i-1}: && o \succ \overline{o} \text{ and } \overline{x} \succ x, \\
&\text{voter } v_{2i} \quad: && \overline{o} \succ o \text{ and } \overline{x} \succ x, \\
&\text{voter } v_{\ell} \quad\;\,: && o \succ \overline{o} \text{ and } x \succ \overline{x}, \\
&\text{voter } v_{\ell'} \quad\;: && \overline{o} \succ o \text{ and } x \succ \overline{x}.
\end{aligned}$$

Voter set W consists of four voters w_{4j-3}, w_{4j-2}, w_{4j-1}, w_{4j} for each clause c_j, $1 \le j \le s$. These four voters have the same preference order

$$\langle \overline{O}\rangle \succ \langle O\rangle \succ \langle A_1\rangle \succ \langle X\rangle \succ \langle A_2\rangle$$

over the set $O \cup \overline{O} \cup A_1 \cup A_2 \cup X$, where $A_1 = \{a_{j'}, b_{j'} \mid j' < j\}$ and $A_2 = \{a_{j'}, b_{j'} \mid j' > j\}$. Note that $A_1 \cup A_2 = A \setminus \{a_j, b_j\}$. Thus, it remains to specify the exact positions of a_j and b_j in the four voters' preference orders: Let $\widehat{X}_1^j$ denote the set of variable alternatives corresponding to the literal in c_j with the lower index and $\widehat{X}_2^j$ denote the set of variable alternatives corresponding to the literal in c_j with the higher index. For instance, if $c_j = \overline{x_2} \vee x_4$, then $\widehat{X}_1^j$ equals $\overline{X_2}$ and $\widehat{X}_2^j$ equals $\overline{X_4}$.

Voters w_{4j-3} and w_{4j-2} rank the clause alternative a_j right below the last alternative in $\langle \widehat{X}_1^j\rangle$ while voters w_{4j-1} and w_{4j} rank it right above the first alternative in $\langle \widehat{X}_1^j\rangle$. As for alternative b_j, voters w_{4j-3} and w_{4j-1} rank b_j right above the first variable alternative in $\langle \widehat{X}_2^j \rangle$ while voters w_{4j-2} and w_{4j} rank it right below the last variable alternative in $\langle \widehat{X}_2^j \rangle$. Thus, these four voters form a δ-configuration with regard to a_j, b_j, x, and y, where $x \in \widehat{X}_1^j$ and $y \in \widehat{X}_2^j$:

$$\begin{aligned}
&\text{voter } w_{4j-3}: && x \succ a_j \text{ and } b_j \succ y,\\
&\text{voter } w_{4j-2}: && x \succ a_j \text{ and } y \succ b_j],\\
&\text{voter } w_{4j-1}: && a_j \succ x \text{ and } b_j \succ y,\\
&\text{voter } w_{4j} \;: && a_j \succ x \text{ and } y \succ b_j.
\end{aligned}$$

We complete the construction by setting the number k of alternatives that may be deleted to $k := r(s+1)+(s-h)$.

The construction clearly runs in polynomial time. It remains to show that (U, C, h) is a yes-instance of MAX2SAT if and only if the constructed profile together with k is a yes-instance of SINGLE-CROSSING ALTERNATIVE DELETION.

For the "only if" part, suppose that there is a truth assignment $U \to \{\text{true}, \text{false}\}^r$ of the variables such that at least h clauses are satisfied. We delete all variable alternatives in X_i if x_i is assigned to true. Otherwise, we delete all variable alternatives in $\overline{X_i}$. Furthermore, we delete the clause alternative b_j if c_j is not satisfied by the assignment. Let X_{rem} be the set of remaining variable alternatives, and let A_{rem} be the set of remaining clause alternatives. Then, $|X_{\text{rem}}| = r(s+1)$ and $|A'| \geq s+h$, implying that the number of deleted alternatives is $|X|+|A|-(|X_{\text{rem}}|+|A_{\text{rem}}|) \leq r(s+1)+(s-h) = k$.

For each $j \in \{1, \ldots, s\}$, we define $\langle z_j \rangle = w_{4j-2} \succ w_{4j} \succ w_{4j-3} \succ w_{4j-1}$ if the literal in clause c_j with the lower index is satisfied; otherwise, $\langle z_j \rangle = w_{4j-3} \succ w_{4j-2} \succ w_{4j-1} \succ w_{4j}$. The resulting profile is single-crossing with respect to the voter order L:

$$v_1 \succ v_3 \succ \ldots \succ v_{2r-1} \succ v_2 \succ v_4 \succ \ldots \succ v_{2r} \succ \langle z_1 \rangle \succ \langle z_2 \rangle \succ \ldots \succ \langle z_s \rangle.$$

Suppose for the sake of contradiction that L is not a single-crossing order, which means that there is a pair $\{a, a'\} \subset O \cup \overline{O} \cup X_{\text{rem}} \cup A_{\text{rem}}$ of alternatives and three voters u, v, w with $u \succ_L v \succ_L w$ such that voter v disagrees with voters u and w on the relative order of a and a'.

Note that all voters along L up to and including voter v_{2r-1} rank $\langle O \rangle \succ \langle \overline{O} \rangle \succ \langle X \rangle$ while all voters from v_2 onwards rank $\langle \overline{O} \rangle \succ \langle O \rangle \succ \langle X \rangle$. Hence, a and a' can neither both be in $O \cup \overline{O}$ nor both be in X_{rem}. Furthermore, a and a' cannot both be in A_{rem} as all voters rank $\langle A \rangle$. Moreover, since all voters rank $(O \cup \overline{O}) \succ (X \cup A)$, neither a nor a' belongs to $O \cup \overline{O}$. This means, without loss of generality, that $a \in X_{\text{rem}}$ and $a' \in A_{\text{rem}}$.

Assume that a' corresponds to clause c_j for some j, that is, $a' \in \{a_j, b_j\}$. Then, for each alternative $a'' \in X_{\text{rem}} \setminus (\widehat{X}_1^j \cup \widehat{X}_2^j)$ that does not correspond to a literal in c_j (recall that $\widehat{X}_1^j$ and $\widehat{X}_2^j$ denote the two sets of variable alternatives corresponding to the literal in c_j with the lower index and the literal in c_j with the lower index, respectively), the following holds. If the first voter in $\langle z_j \rangle$ prefers a'' to a', which means either that all voters rank a'' in front of $\widehat{X}_1^j \cup \widehat{X}_2^j$ or that all voters rank $a' = a_j$ and a'' in front of $\widehat{X}_2^j$, then all voters along the order L up to and including the last voter in $\langle z_j \rangle$ prefer a'' to a' while all remaining voters prefer a' to a''; otherwise all voters along the order L up to and including the last voter in $\langle z_{j-1} \rangle$ prefer a'' to a' while all remaining voters prefer a' to a''. Thus, alternative a' cannot be in $X_{\text{rem}} \setminus (\widehat{X}_1^j \cup \widehat{X}_2^j)$. That is, we have $a \in \widehat{X}_1^j \cup \widehat{X}_2^j$. We distinguish four cases regarding a and a'.

(i) If $a \in \widehat{X}_1^j$ and if $a' = a_j$, then the literal corresponding to $\widehat{X}_1^j$ is not satisfied because $\widehat{X}_1^j$ is not deleted. Thus, $\langle z_j \rangle$ is defined as $w_{4j-3} \succ w_{4j-2} \succ w_{4j-1} \succ w_{4j}$. All voters along L up to and including w_{4j-2} prefer a to a', and all remaining voters prefer a' to a.

(ii) If $a \in \widehat{X}_1^j$ and if $a' = b_j$, then all voters along L up to and including the last voter in $\langle z_j \rangle$ prefer a to a', and all remaining voters prefer a' to a.

(iii) If $a \in \widehat{X}_2^j$ and if $a' = a_j$, then all voters along L up to and including the last voter in $\langle z_{j-1} \rangle$ prefer a to a', and all remaining voters a' to a.

(iv) If $a \in \widehat{X}_2^j$ and if $a' = b_j$, then clause c_j is satisfied because b_j is not deleted. Furthermore, since $\widehat{X}_2^j$ is not deleted, $\widehat{X}_1^j$ must be deleted because clause c_j is satisfied. This implies that the literal in clause c_j with the lower index is satisfied. Thus, $\langle z_j \rangle$ is defined as $w_{4j-2} \succ w_{4j} \succ w_{4j-3} \succ w_{4j-1}$. All voters along L up to and including w_{4j} prefer a to a', and all remaining voters prefer a' to a.

In summary, there is single a voter v along the order L such that all voters up to and including v have the same preference over $\{a, a'\}$ and all remaining voters have the same preference over $\{a, a'\}$—a contradiction to the assumption that L is not a single-crossing order.

For the "if" part, suppose that deleting a set K of at most k alternatives makes the remaining profile single-crossing. Since $|O| = |\overline{O}| \geq k$, at least one pair $\{o, \overline{o}\}$ of dummy alternatives is not deleted, where $o \in O$ and $\overline{o} \in \overline{O}$. Let X_{del} denote the set of all deleted variable alternatives, and A_{del} denote the set of all deleted clause alternatives. Clearly, $|X_{\text{del}}| + |A_{\text{del}}| \leq |K|$. For each $x_i \in U$, at least one set of X_i and $\overline{X_i}$ must be deleted to destroy all δ-configurations involving alternatives in $\{o, \overline{o}\} \cup X_i \cup \overline{X_i}$. This means that $|X_{\text{del}}| \geq r(s+1)$. Thus, $|A_{\text{del}}| \leq |K| - |X_{\text{del}}| \leq k - r(s+1) \leq s - h$. Let C_{both} denote the set of clauses such that neither a_j nor b_j is deleted, $1 \leq j \leq s$, that is,

$C_{\text{both}} := \{c_j \mid \{a_j, b_j\} \cap A_{\text{del}} = \emptyset\}$. Set C_{both} has cardinality at least h because $|A_{\text{del}}| \le s - h$. We show that by setting variable $x_i \in U$ to true if $X_i \subseteq X_{\text{del}}$, and false otherwise, all clauses c_j from C_{both} are satisfied. Suppose for the sake of contradiction that clause $c_j \in C_{\text{both}}$ is not satisfied. This means that $\{a_j, b_j\} \cap A_{\text{del}} = \emptyset$, and that both $\widehat{X}_1^j$ and $\widehat{X}_2^j$ are not completely contained in X_{del}. But then, voters w_{4i-3}, w_{4i-2}, w_{4i-1}, and w_{4i} form a δ-configuration with regard to a_j, b_j, x, x', where $x \in \widehat{X}_1^j \setminus X_{\text{del}}$ and $x' \in \widehat{X}_2^j \setminus X_{\text{del}}$—a contradiction. □

We illustrate our NP-hardness reduction through an example. Consider an instance of MAX2SATwith two variables x_1 and x_2 and four clauses

$$c_1 = x_1 \vee x_2, \quad c_2 = x_1 \vee \overline{x_2}, \quad c_3 = \overline{x_1} \vee x_2, \quad c_4 = \overline{x_1} \vee \overline{x_2}.$$

We set the maximum number h of clauses that can be satisfied by a truth assignment to be three. For instance, the truth assignment $x_1 \mapsto$ true and $x_2 \mapsto$ false satisfies clauses c_1, c_2, c_4. Table 5.2 depicts the reduced instance of SINGLE-CROSSING MAVERICK DELETION with alternative set $O \cup \overline{O} \cup X_1 \cup \overline{X_1} \cup X_2 \cup \overline{X_2} \cup \{a_i, b_i \mid 1 \le i \le 4\}$ and voter set $\{v_i, w_{4i-3}, w_{4i-2}, w_{4i-1}, w_{4i} \mid 1 \le i \le 4\}$. The number k of alternatives that may be deleted is set to $2 \cdot (4+1) + (4-3) = 11$. We can verify that deleting the alternatives from $X_1 \cup \overline{X_2} \cup \{b_3\}$ results in a single-crossing profile where a single-crossing order is $v_1 \succ v_3 \succ v_2 \succ v_4 \succ w_2 \succ w_4 \succ w_1 \succ w_3 \succ w_6 \succ w_8 \succ w_5 \succ w_7 \succ w_9 \succ w_{10} \succ \ldots \succ w_{16}$.

In contrast to the other NP-hard Π MAVERICK DELETION problems, SINGLE-CROSSING MAVERICK DELETION is polynomial-time solvable. The algorithm, which is similar to the single-crossing detection algorithm by Elkind, Faliszewski, and Slinko [EFS12], not only solves the decision problem, but also the optimization problem asking for the maximum-size subset of voters such that the profile restricted to this subset is single-crossing.

Before we proceed to describe the algorithm, we define some notions and make some observations about single-crossing profiles. We call a set S of preference orders *single-crossing* if there is a single-crossing order of the elements in S. We introduce the notion $\Delta(\succ, \succ')$ of the set of *conflict pairs* for two given preference orders $\succ$ and $\succ' \in \mathcal{S}$. By $\Delta(\succ, \succ')$, we denote the set of pairs $\{a, b\}$ of alternatives whose relative order differs between preference orders $\succ$ and $\succ'$. Formally,

$$\Delta(\succ, \succ') := \{\{a, b\} \mid a \succ b \text{ and } b \succ' a\}.$$

For instance, given three alternatives a, b, c, if the preference orders $\succ, \succ'$ are the same, then $\Delta(v, v') = \emptyset$; if the two preference orders are specified as : $b \succ a \succ c$ and $c \succ' b \succ' a$, then $\Delta(\succ, \succ') := \{\{a, c\}, \{b, c\}\}$.

voter v_1:	$\langle O\rangle \succ \langle \overline{O}\rangle \succ \langle \overline{X_1}\rangle \succ \langle X_1\rangle \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter v_2:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ \langle \overline{X_1}\rangle \succ \langle X_1\rangle \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter v_3:	$\langle O\rangle \succ \langle \overline{O}\rangle \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \langle \overline{X_2}\rangle \succ \langle X_2\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter v_4:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \langle \overline{X_2}\rangle \succ \langle X_2\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_1:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ \langle X_1\rangle \succ a_1 \succ \langle \overline{X_1}\rangle \succ b_1 \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_2:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ \langle X_1\rangle \succ a_1 \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ b_1 \succ \langle \overline{X_2}\rangle \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_3:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ b_1 \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_4:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ b_1 \succ \langle \overline{X_2}\rangle \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_5:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ \langle X_1\rangle \succ a_2 \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ b_2 \succ \langle \overline{X_2}\rangle \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_6:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ \langle X_1\rangle \succ a_2 \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_7:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ b_2 \succ \langle \overline{X_2}\rangle \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_8:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ b_2 \succ a_3 \succ b_3 \succ a_4 \succ b_4$
voter w_9:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ a_3 \succ b_3 \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ a_4 \succ b_4$
voter w_{10}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ a_3 \succ \langle X_2\rangle \succ b_3 \succ \langle \overline{X_2}\rangle \succ a_4 \succ b_4$
voter w_{11}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ \langle X_1\rangle \succ a_3 \succ \langle \overline{X_1}\rangle \succ b_3 \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ a_4 \succ b_4$
voter w_{12}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ \langle X_1\rangle \succ a_3 \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ b_3 \succ \langle \overline{X_2}\rangle \succ a_4 \succ b_4$
voter w_{13}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ a_4 \succ \langle X_2\rangle \succ b_4 \succ \langle \overline{X_2}\rangle$
voter w_{14}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ \langle X_1\rangle \succ \langle \overline{X_1}\rangle \succ a_4 \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ b_4$
voter w_{15}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ \langle X_1\rangle \succ a_4 \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ b_4 \succ \langle \overline{X_2}\rangle$
voter w_{16}:	$\langle \overline{O}\rangle \succ \langle O\rangle \succ a_1 \succ b_1 \succ a_2 \succ b_2 \succ a_3 \succ b_3 \succ \langle X_1\rangle \succ a_4 \succ \langle \overline{X_1}\rangle \succ \langle X_2\rangle \succ \langle \overline{X_2}\rangle \succ b_4$

Table 5.2.: An instance $((A, V), k = 11)$ with alternative set $O \cup \overline{O} \cup X_1 \cup \overline{X_1} \cup X_2 \cup \overline{X_2} \cup \{a_i, b_i \mid 1 \le i \le 4\}$ and voter set $\{v_i, w_{4i-3}, w_{4i-2}, w_{4i-1}, w_{4i} \mid 1 \le i \le 4\}$ reduced from the MAX2SAT instance with two variables x_1 and x_2, and with four clauses $c_1 = x_1 \vee x_2$, $c_2 = x_1 \vee \overline{x_2}$, $c_3 = \overline{x_1} \vee x_2$, and $c_4 = \overline{x_1} \vee \overline{x_2}$. We set the maximum number h of clauses that can be satisfied by a truth assignment to be three.

Based on this notion, we can redefine the single-crossing property of a set of preference orders using set inclusions. For the sake of readability, we use the vector notation $(\cdot, \cdots, \cdot)$ to denote a linear order over a set of preference orders.

Lemma 5.10. *A linear order $(\succ_1^*, \succ_2^*, \ldots, \succ_n^*)$ over a set of n preference orders is single-crossing if and only if for each two preference orders $\succ_i^*$ and $\succ_j^*$ with $1 \le i \le j \le n$ it holds that $\Delta(\succ_1^*, \succ_i^*) \subseteq \Delta(\succ_1^*, \succ_j^*)$.*

Proof. The "only if" part follows directly from the definition of the single-crossing property and the set of conflict pairs. For the "if" part, suppose towards a contradiction that the order $(\succ_1^*, \succ_2^*, \ldots, \succ_n^*)$ is not single-crossing. This means that there are two alternatives a, b, and there are two preference orders $\succ_i^*, \succ_j^*$ with $1 < i < j$ such that $a \succ_1^* b$ and $a \succ_j^* b$, but $b \succ_i^* a$. Then it follows that $\{a, b\} \in \Delta(\succ_1^*, \succ_i^*)$ but $\{a, b\} \notin \Delta(\succ_1^*, \succ_j^*)$—a contradiction. □

The following observation states that the single-crossing property only depends on the preference orders, not on the voters.

Observation 5.11. *Let V be a set of voters and let $w \notin V$ be an additional voter such that there is a voter in V who has the same preference order as voter w. Then, the profile with voter set V is single-crossing if and only if the profile with voter set $V \cup \{w\}$ is single-crossing.*

Proof. By the definition of single-crossing orders, a profile is single-crossing if and only if the set of the preference orders of all voters in this profile is single-crossing. Since adding voter w to voter set V does not change the set of the preference orders of all voters in V, the statement follows. □

Based on the notions of conflicting pairs and single-crossing sets of preference orders, and Lemma 5.10 and Observation 5.11, we can solve the maximization variant of the SINGLE-CROSSING MAVERICK DELETION problem by reducing it to finding a longest path in an appropriately constructed directed acyclic graph. This implies the following theorem.

Theorem 5.12. SINGLE-CROSSING MAVERICK DELETION *is solvable in $O(n^3 \cdot m^2)$ time, where n denotes the number of voters and m denotes the number of alternatives.*

Proof. Suppose that we are given a profile with A being the set of m alternatives and V being the set of n voters, each voter having a preference order over A. Our goal is to find a maximum-size subset of voters such that the profile restricted to this subset is single-crossing. To this end, we use two further notions: Let $\mathcal{S}(V) := \{\succ_v \mid v \in V\}$ be the set of the preference orders of all voters from V; without loss of generality, let $\mathcal{S}(V) := \{\succ_1, \succ_2, \ldots, \succ_{n'}\}$. For each preference order $\succ \in \mathcal{S}(V)$, let $\#(\succ, V)$ denote the number of voters in V with the same preference order $\succ$. By Observation 5.11, it follows that finding the maximum-size single-crossing voter subset is equivalent to finding a *single-crossing* subset $\mathcal{S}' \subseteq \mathcal{S}(V)$ of preference orders that maximizes the sum $\sum_{\succ \in \mathcal{S}'} \#(\succ, V)$.

We observe that if $\succ$ is the first preference order along the single-crossing order over set $\mathcal{S}'$, then for each two further preference orders $\succ', \succ'' \in \mathcal{S}'$ with $\succ'$ being the predecessor of $\succ''$ along the single-crossing order, by Lemma 5.10, it holds that $\Delta(\succ, \succ') \subseteq \Delta(\succ, \succ'')$. This inspires us to build a directed graph based on the set inclusion relation and, then, to find a maximum-weight path. Thus, the idea of our algorithm is to first construct a directed graph with weighted arcs and, then, to find a maximum-weight path on this graph. We provide an example to illustrate this idea right after this proof.

The construction of the desired directed graph works as follows: For each two numbers z, $i \in \{1,2,\dots,n'\}$, we construct one vertex u_i^z; this vertex will represent the preference order $\succ_i$ in a linear order starting with preference order $\succ_z$. Then, for each further number $i' \in \{1,2,\dots,n'\}$ with $i \neq i'$, we add an arc with weight $\#(\succ_{i'}, V)$ from vertex v_i^z to vertex $v_{i'}^z$ if $\Delta(\succ_z, \succ_i) \subseteq \Delta(\succ_z, \succ_{i'})$. Finally, we construct a root vertex u_r, and for each number $z \in \{1,2,\dots,n'\}$, we add an arc with weight $\#(\succ_z, V)$ from root u_r to u_z^z. This completes the construction. Observe that the constructed directed graph is acyclic:

1. For each three numbers z, z', $i \in \{1,2,\dots,n'\}$ with $z \neq z'$, there are no arcs between vertices u_i^z and $u_i^{z'}$.

2. For each three numbers z, i, $i' \in \{1,2,\dots,n'\}$ with $i \neq i'$, a path from u_i^z to $u_{i'}^z$ implies that $\Delta(\succ_z, \succ_i) \subseteq \Delta(\succ_z, \succ_{i'})$, while a path from $u_{i'}^z$ to u_i^z implies that $\Delta(\succ_z, \succ_{i'}) \subseteq \Delta(\succ_z, \succ_i)$. Thus, both paths cannot exist simultaneously because $\succ_i \neq \succ_{i'}$.

Now, an order of the vertices along a maximum-weight directed path corresponds to a subset $\mathscr{S}' \subseteq \mathscr{S}(V)$ of preference orders such that $\mathscr{S}'$ is *single-crossing*, and the sum $\sum_{\succ \in \mathscr{S}'} \#(\succ, V)$ is maximum: The second vertex on the maximum-weight path fixes the first preference order of the single-crossing order. Each successive vertex u_i^z on the path represents the successive preference order $\succ_i$ in the single-crossing order (this is true by Lemma 5.10 and by the way we define an arc). The arc weights ensure that the sum of the weights on the path equals the total number of represented voters.

As to the running time analysis, we need $O(n \cdot m)$ time to compute the set $\mathscr{S}(V)$. Then, for each two (not necessarily distinct) preference orders $\succ$, $\succ' \in \mathscr{S}(V)$, we compute $\Delta(\succ, \succ')$. This can be done by checking the relative order of each pair of alternatives in $O(n^2 \cdot m^2)$ time. Further, we construct the directed graph in $O(n^3 \cdot m^2)$ time. Finally, we compute the maximum-weight path in a directed acyclic graph with n^2 vertices and n^3 arcs in $O(n^3)$ time. To achieve this, we first replace all positive weights w with $-w$, and then use the algorithm in the textbook of Cormen et al. [Cor+09, Sec 24.2] to find a minimum-weight path. In total, the running time is $O(n^3 \cdot m^2)$. $\square$

We have just seen that SINGLE-CROSSING MAVERICK DELETION is solvable in polynomial time by reducing our problem to finding a maximum-weight path in an appropriately constructed directed acyclic graph. Now, we illustrate this approach with a concrete example.

Example 5.1. Consider a profile $\mathcal{P}$ with three alternatives a, b, c, and five voters v_1, v_2, v_3, v_4, v_5, whose preference orders are depicted in Figure 5.3(a). This profile is not single-crossing as it contains a γ-configuration with regard to the three pairs of alternatives $\{a,c\}$, $\{a, b\}$, $\{a,b\}$ and voters v_1, v_3, v_4. The set of the preference orders of all voters is $\{a \succ b \succ c,\ b \succ c \succ a,\ c \succ a \succ b,\ c \succ b \succ a\}$. According to our algorithm of finding a single-crossing profile with maximum number of voters, we first construct a weighted directed graph as depicted in Figure 5.3(b). Then, we will find a maximum-weight path in the graph. We can verify that there are four maximum-weight paths, including the path $u_r \to (a \succ b \succ c) \to (b \succ c \succ a) \to (b \succ c \succ a)$ with weight four. Thus, a single-crossing profile with the largest voter subset contains four voters. For instance, the profile with voters v_1, v_2, v_3, v_5.

5.9 Concluding remarks

We have shown that SINGLE-CROSSING MAVERICK DELETION can be solved in polynomial time (Theorem 5.12). The algorithm proposed there, first constructs an arc-weighted acyclic directed graph for a given profile, and then finds a maximum-weight path on this graph. This rather straight-forward approach yields a running time of $O(n^3 \cdot m^2)$ where n denotes the number of voters and m denotes the number of alternatives. However, Doignon and Falmagne [DF94] used a special data structure and proposed an $O(n^2 + n \cdot m \cdot \log(m))$ algorithm for detecting single-crossingness, which can be interpreted as making a profile single-crossing without deleting any voters. It would be interesting to know whether their idea can be adapted to our problem to obtain an algorithm with lower running time.

In contrast to the polynomial-time solvability for SINGLE-CROSSING MAVERICK DELETION, deciding whether it is possible to make a profile nicely structured by deleting at most k voters or at most k alternatives is NP-complete for all other considered properties (SINGLE-PEAKED ALTERNATIVE DELETION has been proven NP-complete by Erdélyi, Lackner, and Pfandler [ELP13]). However, we note that all these problems become tractable when k is a small constant: All considered properties are characterized by a fixed number of small forbidden substructures. Thus, by branching over all possible voters (resp. alternatives) of each forbidden substructure in the profile, we obtain a fixed-parameter algorithm that is efficient for small distances

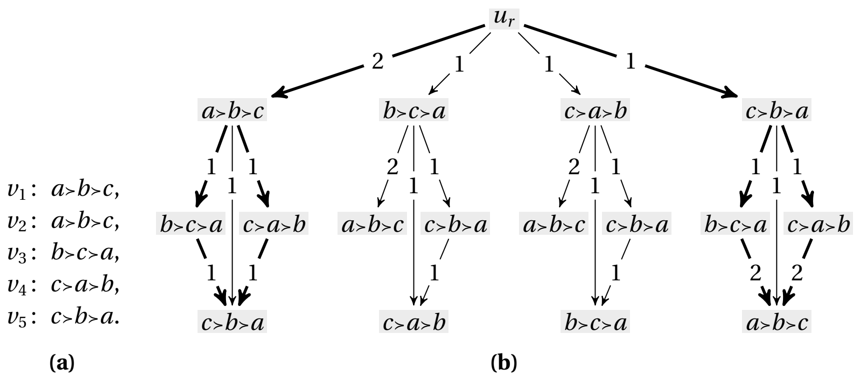


Figure 5.3.: An example illustrating how to construct a weighted directed graph for a given profile in order to solve SINGLE-CROSSING MAVERICK DELETION. (a) A profile with four voters and three alternatives. Note that the first two voters have the same preference order, and this profile is not single-crossing. (b) A weighted directed graph for the profile in (a). Note that we label each vertex with its corresponding preference order. The weight of an arc denotes the number of voters in the profile that have the preference order denoted by the label of the arc's target vertex. For instance, there is an arc from the root u_r to its left most "child" $a{\succ}b{\succ}c$ with weight 2. This means that the left profile has two voters with preference order $a{\succ}b{\succ}c$. Any path that starts with the root u_r and consists of thick arrows is a maximum-weight path.

(see Section 2.6 for more information on parameterized complexity notions). To complement our theoretical results, it would be interesting to perform an empirical study on real-world data with regard to two questions. The first question is how close real-world preference profiles are to having a specific structure. Przedmojski [Prz16] has recently conducted experiments to measure the minimum number of voters (resp. alternatives) to delete to make a profile single-peaked (resp. single-crossing). He found out that almost all of the real-world profiles available from PrefLib [MW13] are far from being single-peaked or single-crossing. However, since the investigated profiles are from some special types of non-political elections and since only very few profiles are investigated (about 300 with different numbers of alternatives and voters), the finding of little evidence for being close to single-peaked or single-crossing is not surprising. The second questions is about the exact computational complexity of finding the distance to the "nearest" structured profile. Of course, empirical study needs neatly designed algorithms. Thus, it is of practical interest to investigate more sophisticated and more efficient (fixed-parameter) algorithms to compute the distance of a profile to a nicely structured one; work on this has been recently started by Elkind and Lackner [EL14].

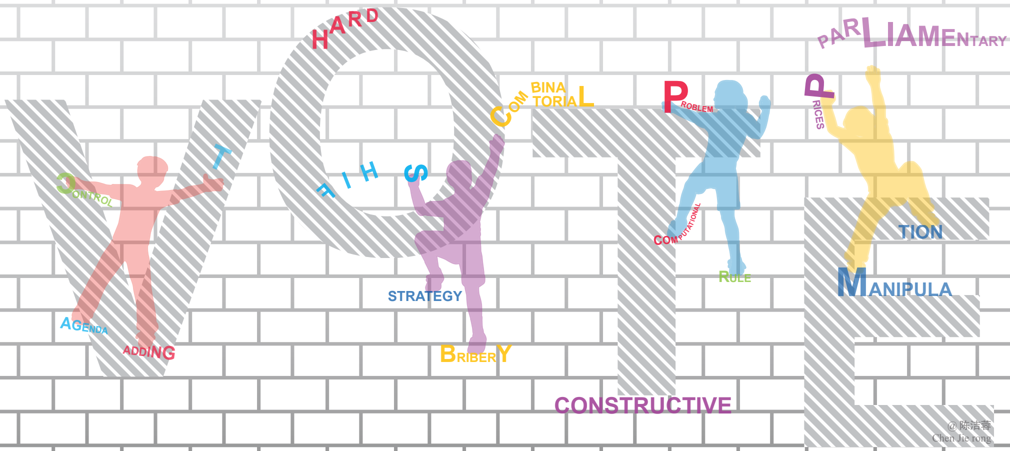
Besides the domain restrictions studied in this chapter, we have looked at the one-dimensional (1-D) Euclidean property in Chapter 4. One-dimensional Euclidean profiles are necessarily single-peaked and single-crossing. Like the single-peaked and single-crossing properties, 1-D Euclidean profiles can be recognized in polynomial time [DF94, EF14, Kno10]. The computational complexity of making a profile 1-D Euclidean using a minimum number of modifications remains unexplored. We remark that our NP-hardness reduction for the single-peaked property and the maverick deletion case does not work for this property as we can easily verify that the resulting single-peaked instance is not single-crossing which is a necessary condition for being 1-D Euclidean.

As already mentioned in the related work section, many voting problems that are computationally hard for general preference profiles become tractable for nicely structured preference profiles. It would be interesting to know whether and in which way such tractability results transfer to profiles that are only close to being nicely structured (these are the key questions 6 and 7 in [Bre+14a]); see for instance, Faliszewski, Hemaspaandra, and Hemaspaandra [FHH14] and Menon and Larson [ML16] for some more discussion of the nearly single-peaked domain.

We close this chapter by remarking that our domain restrictions have also been studied for dichotomous preference profiles, where each voter says either yes or no to each of the alternatives [EL15], and for profiles with incomplete preference orders [Elk+15, Lac14]. It would be interesting to investigate nearly restricted domains for both cases.

Part II

Computationally Hard Voting Problems



CHAPTER 6

Combinatorial Voter Control

Information, knowledge, is power. If you can control information, you can control people.

Tom Clancy, 1995

Voter control problems model situations in which an external agent tries to affect a voting result by adding voters, for example, by convincing some voters to vote who would otherwise not participate. In the standard model, voters are added one at a time, with the goal of making a distinguished alternative win by adding a minimum number of voters. In this chapter, we initiate the study of combinatorial variants of control by adding voters: In our setting, when we choose to add a voter, we also have to add a whole bundle of voters associated with him. We study the computational complexity of this problem for two of the most basic voting rules, namely, the plurality rule and the Condorcet rule.

6.1 Introduction

We study the computational complexity of the control by adding voters problem [BTT92, HHR07], investigating the case where the sets of voters that we can add have some combinatorial structure. The problem of control by adding voters models situations where some agent (for example, a campaign manager of one of the alternatives) tries to ensure a given alternative's victory by convincing some undecided voters to vote. Traditionally, in this problem we are given a description of a preference profile (that is, a set A of alternatives and a set V of voters who already decided to vote), and also a set W of undecided voters. For each voter in $V \cup W$, we assume that we know how this voter intends to vote, which is expressed as a linear order over the set A; while this assumption seems somewhat unrealistic, it is a standard assumption within computational social choice, and we might have a good

This chapter is based on "Combinatorial Voter Control in Elections" by L. Bulteau, J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon, *Theoretical Computer Science* [Bul+15] .

approximation of this knowledge from pre-election polls. Our goal is to ensure that our preferred alternative p becomes a winner by convincing as few voters from W as possible to vote—provided that it is at all possible to ensure p's victory in this way.

Control by adding voters corresponds, for example, to situations where supporters of a given alternative make direct appeals to other supporters of the alternative to vote. For example, they may stress the importance of voting or help with the voting process by offering rides to the voting locations. Unfortunately, in its traditional phrasing, control by adding voters does not model larger-scale attempts at convincing people to vote. For instance, a campaign manager might be interested in airing a TV advertisement that would motivate supporters of a given alternative to vote (though, of course, it might also motivate some of this alternative's enemies), or maybe launch viral campaigns, where friends convince their own friends to vote. In many scenarios, the sets of voters we can add have some specific combinatorial structure, basically meaning that it is possible to add a combination (subset) of voters at a unit price. For instance, a TV advertisement appeals to a particular group of voters so that we can add all of them at the unit cost of airing the advertisement; a public speech in a given neighborhood will convince a particular group of people to vote at the unit cost of organizing the meeting; convincing an outgoing person to vote will "for free" also convince her friends to vote.

The goal of this chapter is to formally define an appropriate computational problem that models a combinatorial variant of control by adding voters, and to study its computational complexity. Specifically, we focus on the plurality rule and the Condorcet rule, mainly because the plurality rule is the most widely used rule in practice and the Condorcet rule models a large family of Condorcet-consistent rules. Moreover, for the plurality rule the standard variant of control by adding voters is solvable in linear time [BTT92]. This makes it one of only a few rules for which this problem is solvable in polynomial time. For the Condorcet rule, the standard variant of the control by adding voters problem changes from being NP-complete [BTT92] to polynomial-time solvable if we assume that the profiles are single-peaked [Bra+15] or single-crossing [MF14] (see Section 4.4.1 and Section 3.4 for the definition of single-peaked and single-crossing profiles). For both the plurality and the Condorcet rules and for the combinatorial variant, where adding a voter may cause some additional voters being added at a unit cost, we obtain hardness results for very restricted scenarios (Theorem 6.6 and Theorem 6.25) . All these hardness results for the Condorcet rule directly translate to all Condorcet-consistent voting rules, a large and important family of voting rules.

We defer the formal details, definitions, and concrete results to the following sections. Instead, we state the high-level main messages of our work. Herein, we assume that adding an unregistered voter means adding a *bundle* (subset) of unregistered voters; in this way, it is easy to see that the standard variant of control by adding voters is a special case of the combinatorial variant (set the bundle of each unregistered voter to be a singleton consisting of this single voter):

1. Many typical variants of combinatorial control by adding voters are intractable, but there is also a rich landscape of tractable cases. For instance, the problem is NP-complete already when there are only two alternatives or when the bundle size is at most two. However, with bundle sizes up to two, the problem is either fixed-parameter tractable for the parameter "the number k of bundles to add" or even polynomial-time solvable when requiring the bundling function to be full-d (see Section 6.3 for the definition; informally, this means that only voters with roughly the same preference orders can be bundled together).

2. Assuming that voters have single-peaked preferences does not lower the complexity of the problem (even though it does so in many other voting problems [Bra+15, Con09, Fal+11]). In contrast, assuming single-crossing preferences does lower the complexity of the problem.

We provide a detailed summary of our contributions at the end of Section 6.4. We believe that our setting of combinatorial control, and—more generally—of combinatorial problems that model manipulative attacks, offers a very fertile ground for future research and we intend the current chapter to be an initial step.

Related work. Bartholdi III, Tovey, and Trick [BTT92] were the first to study the concept of voting control by adding/deleting voters/alternatives to a given preference profile. They considered the constructive variant of the problem, where the goal is to ensure a given alternative's victory (and we focus on this variant of the problem as well). The destructive variant, where the goal is to prevent someone from winning, was introduced by Hemaspaandra, Hemaspaandra, and Rothe [HHR07]. These papers focused on the plurality rule and the Condorcet rule (and also the Approval rule, for the destructive case of Hemaspaandra, Hemaspaandra, and Rothe [HHR07]). Since then, many other researchers extended these results to a number of other rules and models [EHH15, Erd+15, Fal+09a, Fal+15, FHH11a, FHH15, Mei+08, PX12].

We study combinatorial control problems using the tools and methods of parameterized complexity theory (see Section 2.5 for more information). Most frequently, parameterized complexity of control problems is studied for the parameter "number of alternatives" [Fal+09a, FHH11a, HLM13]. The number of voters has received far less attention as a parameter (for the case of control, the parameter appears, for example, in the work of Betzler and Uhlmann [BU09] and, very recently, in the work of Chen et al. [Che+15a]; Brandt et al. [Bra+13] considered it in the context of winner determination). Several authors have also considered other parameters, such as the solution size (for example, the number of voters one can add). Papers focusing on this parameter include, for example, those of Liu et al. [Liu+09], Liu and Zhu [LZ10], and Erdélyi et al. [Erd+15].

Some of our results describe the complexity of voting control for the case where the voters' preference orders are either single-peaked [Bla48] or single-crossing [Mir71, Rob77] (see also Section 3.4 and Section 5.5.2 for more information on these two domains). The complexity of control for single-peaked profiles was first studied by Faliszewski et al. [Fal+11] and later by Brandt et al. [Bra+15], Faliszewski, Hemaspaandra, and Hemaspaandra [FHH14], and others. The case of control for single-crossing profiles was investigated by Magiera and Faliszewski [MF14]. Generally speaking, the control problems often drop from being NP-complete to being polynomial-time solvable when one of these domain restrictions is assumed. Naturally, single-peakedness and single-crossingness were studied algorithmically in many other contexts as well. Perhaps the first authors who observed that they may lower the complexity of voting problems were Walsh [Wal07] and Conitzer [Con09].

In most work on voting control, the authors assume that one could affect each entity of the preference profile at unit cost only. For example, one could add a voter at unit cost and adding two voters always is twice as expensive as adding a single voter. There is one exception known to us: Faliszewski, Hemaspaandra, and Hemaspaandra [FHH15] studied control with weighted voters, that is, adding a voter of weight w means adding a group of w voters, each with unit weight. On the one hand, this weighted voter model does not allow one to express rich combinatorial structures such as those studied here, while on the other hand, in our study we consider unweighted voters only (though adding weights to our model would be seamless). Very recently, Chen et al. [Che+15a] studied the combinatorial variant of both constructive and destructive control by either adding or deleting alternatives. They discovered that, with few voters, the complexity of the corresponding control problem for different voting rules ranges from polynomial-time solvable to NP-hard even for a constant number of voters.

Erdélyi, Hemaspaandra, and Hemaspaandra [EHH15] also studied a variant of combinatorial control by adding, deleting, or partitioning of the voters. They used a slightly different—though also very natural—model of bundling voters, where each voter has a label and each bundle consists exactly of the voters with a given label. Formally, our models are incomparable and, indeed, we show hardness results for the case of combinatorial control by adding voters with bundles of size two, whereas in their model this case is easily seen to be polynomial-time solvable. Our model is loosely related to the safe strategic voting model introduced by Slinko and White [SW14], where a voter v with preference order $\succ_v$ announcing to vote $\succ_v^*$ instead may motivate several other voters with the same preference order as $\succ_v$ to also vote $\succ_v^*$. Since the outcome of the manipulated election depends on the voters who eventually vote $\succ_v^*$ and thus, may be worse for v, the corresponding question is whether it is ever safe to vote strategically. Adapted to our bundling function, the potential manipulators in that model are exactly those whose swap distances are zero to the preference order of voter v.

The specific combinatorial flavor of our model is inspired by the seminal work of Rothkopf, Pekeč, and Harstad [RPH98][1] on *combinatorial auctions* (see, for example, the work of Sandholm [San06] for additional information). There, bidders can place bids on combinations of items such that the bid on the combination of a set of items might be less than, equal to, or greater than the sum of the individual bids on each element from the same set of items. While in combinatorial auctions one “bundles” items to bid on, in our scenario one bundles voters.

In the computational social choice literature, combinatorial voting is typically associated with scenarios where voters express opinions over a set of alternatives that themselves have a specific combinatorial structure (typically, one uses CP-nets to model preferences over such sets of alternatives [Bou+04]). For example, Conitzer, Lang, and Xia [CLX09] studied a form of control in this setting and Mattei et al. [Mat+12] studied bribery problems. In contrast, we use the standard model of preference profiles where all alternatives and preference orders are given explicitly, but we have a combinatorial structure on the sets of voters which can be added.

6.2 Chapter outline

In Section 6.3, we introduce additional notations relevant to this chapter (see also Chapter 2 for basic notations including preference profiles, voting rules, etc.). In Section 6.4, we formally define our central problem and summarize our contributions. In the following sections, we go on to study the complexity of combinatorial voter control problems where voters may have arbitrary preference orders: in Section 6.5, we focus on the so-called canonical parameters, namely the solution size, the number of alternatives, and the number of unregistered voters. In Section 6.6 we focus on parameters arising from the combinatorial structure, that is, the maximum bundle size and the swap distance. In Section 6.7, we analyze situations where the

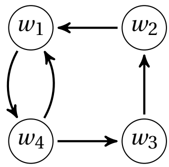
[1] According to Google Scholar, accessed September 2016, cited more than 1000 times.

$w_1 : a \succ b \succ c$
$w_2 : b \succ a \succ c$
$w_3 : a \succ c \succ b$
$w_4 : c \succ a \succ b$

(a)

$w_1 \mapsto \{w_1, w_4\}$
$w_2 \mapsto \{w_1, w_2\}$
$w_3 \mapsto \{w_2, w_3\}$
$w_4 \mapsto \{w_1, w_3, w_4\}$

(b)



(c)

Figure 6.1.: (a): Unregistered voters' preference orders. (b): A bundling function. (c): The corresponding bundling graph with the names of the voters as vertex labels.

voters' preference orders are either single-peaked or single-crossing. We conclude in Section 6.8 with several future research directions.

6.3 Definitions, notations, and examples

We refer to Sections 2.3 to 2.5 for the definitions of preference profiles, the Condorcet and the plurality rules, and relevant concepts from computational and parameterized complexity theory. We introduce one new notion regarding the most preferred alternative of a voter. We call a voter $v \in V$ a *c-voter* if alternative c is in the first position of v's preference order. In the following, we use V to refer to a set of registered voters and W to refer to a set of unregistered voters. Accordingly, we use v to refer to a registered voter and w to refer to an unregistered voter.

In the following, we focus on the concepts of combinatorial voter control.

Definition 6.1 (Bundling function)**.** Given a voter set W, a combinatorial bundling function $\kappa\colon W \to 2^W$ (abbreviated as *bundling function*) is a function assigning to each voter a subset of voters. For $x \in W$, $\kappa(x)$ is called x's *bundle* and x is called the *leader* of this bundle. We assume that $x \in \kappa(x)$ so that $\kappa(x)$ is never empty. We typically write b to denote the maximum bundle size under a given κ (which will always be clear from context). For each subset $W' \subseteq W$, we write $\kappa(W') := \bigcup_{x \in W'} \kappa(x)$ to denote the union of the bundles $\kappa(x)$ of each voter $x \in W'$.

Intuitively, we use combinatorial bundling functions to describe the sets of voters that we can add to the preference profile at unit cost. For example, one can think of $\kappa(x)$ as the group of voters that are going to vote under x's influence. Bundling functions can be represented explicitly: for each voter x, simply list the voters in $\kappa(x)$.

Example 6.1. Consider a profile $\mathscr{P}$ with three alternatives a, b, c, such that the plurality score of a is zero, $\mathsf{score}_{\mathscr{P}}(a) = 0$, the score of b is one, and the score of c is two. Let $W = \{w_1, w_2, w_3, w_4\}$ be a set of four unregistered voters whose preference orders are depicted in Figure 6.1a. Now, consider a bundling function κ (as depicted in Figure 6.1b) for W. The maximum bundle size is three. In order to make alternative a a plurality winner, we have to let a gain two points and c gain zero points. Thus, the only way to make a win is to add the bundles of voters w_2 and w_3 to the profile.

It is worthwhile to briefly discuss the model of bundling functions here. We could have defined our problem differently, by having sets of voters as the bundles, without a distinguished leader (somewhat similarly to the model of Erdélyi, Hemaspaandra, and Hemaspaandra [EHH15], but with each voter having possibly many labels). We chose our approach based on the idea that upon convincing a single voter to vote, the friends of this convinced voter would likely follow. We mention that most of our results transfer to this other model as well.

We are interested in various special cases of bundling functions.

Definition 6.2 (Properties of bundling functions)**.** We say that κ is *leader-anonymous* if each two voters x and y with the same preference orders have the same bundle, that is, $\kappa(x) = \kappa(y)$. We say that κ is *follower-anonymous* if each two voters x and y with the same preference orders follow the same leaders, that is, for each voter z, it holds that $x \in \kappa(z)$ if and only if $y \in \kappa(z)$. We call κ *anonymous* if it is both leader-anonymous and follower-anonymous.

One possible way of thinking about an anonymous bundling function is that it is a function assigning to each preference order appearing in the input a subset of the preference orders appearing in the input. For example, anonymous bundling functions naturally model scenarios such as airing TV advertisements that appeal to particular groups of voters.

To model the situation where only voters with roughly the same preference orders can be bundled to together, we introduce the notion of swap distances.

Definition 6.3 (Swap distances and full-d bundling functions)**.** For two voters w_i and w_j with preference orders $\succ_i$ and $\succ_j$, we define their *swap distance* as the number of (unordered) pairs of alternatives that $\succ_i$ and $\succ_j$ order differently. Given a number $d \in \mathbb{N}$, we call $\kappa\colon W \to 2^W$ a *full-d bundling function* if for each voter $x \in W$, bundle $\kappa(x)$ consists of all voters $y \in W$ such that x and y have swap distance at most d, that is,

$$\kappa(x) := \{y \in W \mid x \text{ and } y \text{ has swap distance at most } d\}.$$

The number d in a full-d bundling function describes the similarity of the preference orders of a voter and its followers; a smaller value of d means a higher degree of similarity.

We introduce the concept of the *bundling graph* of a bundling function, which, roughly speaking, models how the bundles of two voters interact with each other.

Definition 6.4 (Bundling graphs)**.** Given a bundling function κ (over the set W of voters), the *bundling graph* is a directed graph $G = (V(G), E(G))$ constructed as follows. For each voter $x \in W$, there is one vertex u_x in $V(G)$. For each voter y and each follower $z \in \kappa(z)$ in his bundle with $z \neq y$, there is an arc (u_y, u_z) from u_y to u_z in the arc set $E(G)$.

Given an alternative c, we say that a vertex is a c*-vertex* if the corresponding voter is a c-voter; otherwise we call it a *non-*c*-vertex*. Accordingly, we say that an arc is a (c_1, c_2)*-arc* if the source of this arc is a c_1-vertex and the target is a c_2-vertex.

For a positive integer z, a z*-star* is a directed graph consisting of $(z+1)$ vertices and z arcs such that there is a vertex with z (in- or out-) neighbors.

A *maximum matching* is a largest subset of endpoint-disjoint arcs.

Example 6.2. Let us look at the profile and the bundling function from Example 6.1 again. The bundling function is anonymous as no two voters have the same preference order. It is not full-d because, for instance, the bundle of voter w_1 contains voter w_4 which has a swap distance of two to w_1 but it does not contain voter w_2 which has a swap distance of one to w_1. The corresponding bundling graph is depicted in Figure 6.1c, where each vertex is labeled with its corresponding voter.

For arbitrary bundling functions, the bundling graph is a directed graph. However, if bundling function κ is full-d, then we can consider it as an undirected one. The reason is that in this case for every two unregistered voters x and y we have that $y \in \kappa(x)$ if and only if $x \in \kappa(y)$. In consequence, for each arc (u_x, u_y) in the bundling graph, the reverse arc (u_y, u_x) is also present.

Proposition 6.1. *If a bundling function κ is full-d, then for each unregistered voter x and each voter $y \in \kappa(x)$ it holds that $x \in \kappa(y)$.*

Proof. For each two voters x and y, if $y \in \kappa(x)$, then the swap distance between x and y is at most d. Therefore, since κ is a full-d bundling function and the swap distance is clearly symmetric, x must belong to the bundle $\kappa(y)$. □

Note that the "mutual containment" property, as stated above, does not hold for every bundling function. For example, κ with $\kappa(x) = \{x, y\}$ and $\kappa(y) = \{y\}$ is a valid bundling function. The following is easy to observe, as full-d bundling functions depend only on the preference orders and not on the specific voters:

Proposition 6.2. *If a bundling function κ is full-d, then it is also anonymous.*

Proof. To show that κ is anonymous we need to show both leader-anonymity and follower-anonymity. Suppose that κ is a full-d bundling function. Let x, y, z be three voters such that x and y have the same preference order. If $z \in \kappa(x)$ (resp. if $x \in \kappa(z)$), then z has a swap distance of at most d to x, and hence, to y. By the definition of full-d bundling functions, $z \in \kappa(y)$ (resp. $y \in \kappa(z)$). This shows the leader-anonymity (resp. follower-anonymity) of κ. □

6.4 Central problem and results

We define our central problem of combinatorial constructive control by adding voters as follows.

> COMBINATORIAL CONSTRUCTIVE CONTROL BY ADDING VOTERS (C-CC-AV)
> **Input:** A preference profile (A, V) with linear preference orders, a set W of unregistered voters with $V \cap W = \emptyset$ such that each unregistered voter has a linear preference order over A, a bundling function $\kappa\colon W \to 2^W$, a preferred alternative $p \in A$, and a non-negative integer $k \in \mathbb{N}$.
> **Question:** Is there a subset of voters $W' \subseteq W$ of size at most k such that p is a winner in the profile with voter set $V \cup \kappa(W')$?

We investigate two popular voting rules, the plurality rule and the Condorcet rule, which are defined in Section 2.4. We use the so-called non-unique-winner model. That is, for a control action to be successful it suffices for p to be one of the co-winners. Throughout this chapter, we refer to each subset $W' \subseteq W$ of voters such that p is a co-winner in the profile $(A, V \cup \kappa(W'))$ and $|W'| \leq k$ as a *solution*. We refer to k as the *solution size* (formally, k is an upper bound on the allowed solution size, but this notation makes the discussion a bit simpler). For the plurality rule, we also assume that the score difference between the current winner and p does not exceed the total number of p-voters in W.

COMBINATORIAL CONSTRUCTIVE CONTROL BY ADDING VOTERS (C-CC-AV) is a generalization of the well-studied problem CONSTRUCTIVE CONTROL BY ADDING VOTERS (CC-AV) (where, for each voter $x \in W$, we have $\kappa(x) = \{x\}$). CC-AV is linear-time solvable for the plurality rule by a simple calculation [BTT92] (also see the remark at the beginning of Section 6.6), but it is NP-complete for the Condorcet rule [Liu+09]. This immediately implies two observations:

Observation 6.3. *For the plurality rule,* C-CC-AV *is solvable in linear time if the maximum bundle size* b *is one.*

Observation 6.4. *For the Condorcet rule,* C-CC-AV *is* NP*-hard even if the maximum bundle size* b *is one.*

Our contributions. We introduce a new model for combinatorial control in voting. Our results show that COMBINATORIAL CONSTRUCTIVE CONTROL BY ADDING VOTERS (C-CC-AV) is NP-hard even for the plurality rule. For this reason, we complement our study by focusing on a number of different parameters, showing both fixed-parameter tractability results and parameterized hardness results (see Section 2.5 for more information on parameterized complexity concepts). We almost completely resolve the computational complexity status of C-CC-AV for the plurality rule and the Condorcet rule as a function of the maximum bundle size b and the maximum swap distance d from a voter v to the farthest element of v's bundle. For the plurality rule, the complexity of the problem depends on b in the following way:

1. If $b = 1$, then C-CC-AV is linear-time solvable (this is due to Bartholdi III, Tovey, and Trick [BTT92]; see Observation 6.3).

2. If $b = 2$, then the complexity of C-CC-AV depends on the bundling function. If the bundling function is full-d, then the problem is polynomial-time solvable (Theorem 6.20). Otherwise, the problem is NP-hard (Theorem 6.13), but is in FPT for the parameter "solution size" (Theorem 6.17).

3. If $b = 3$, then C-CC-AV is W[1]-hard even for anonymous bundling functions (Theorem 6.8), and it is NP-hard for full-d bundling functions, even if $d = 3$ (Theorem 6.21). The case of the maximum swap distance d equaling two remains open.

4. For any constant size $b \geq 4$, C-CC-AV is NP-hard already for full-d bundling functions with $d = 1$ (Theorem 6.22); for $d = 0$, which in essence means looking at the weighted control case for the special case of unary-encoded weights, the problem is polynomial-time solvable [FHH15].

For the Condorcet rule, we obtain NP-hardness even when the input profile has only two alternatives (of course, this result applies to the plurality rule as well; for two alternatives the two rules are identical). Furthermore, we show that for the case of full-d bundling functions, for both the plurality rule and the Condorcet rule, C-CC-AV remains hard even when restricting the profiles to be single-peaked, but

Arbitrary function κ	Complexity	Reference
In general	NP-c	Thm 6.6
	XP (parameter: k)	Pro 6.5
Singe-peaked & single-crossing		
$m = 2$, b unbounded,	W[2]-h (parameter: k)	Thm 6.6, Obs. 6.23
$m = 3$, $b = 3$	W[1]-h (parameter: k)	Pro 6.9, Obs. 6.24
m unbounded, $b = 2$	FPT (parameter: k)	Thm 6.17

Anonymous function κ, m unbounded	Complexity	Reference
In general	ILP-FPT (parameter: m)	Cor 6.11
Non-full-d function κ		
$b = 2$	NP-c	Thm 6.13
$b = 3$	W[1]-h (parameter: k)	Thm 6.8
Full-d function κ		
$b \leq 2$	P	Thm 6.20
$b = 3, d = 3$	NP-c	Thm 6.21
$b = 4, d = 1$	NP-c	Thm 6.22
Single peaked	NP-c, W[1]-h (parameter: k)	Thm 6.25
Single-crossing	P	Thm 6.29

Table 6.1.: Computational complexity classification of C-CC-AV for the plurality rule. Since the non-combinatorial problem CC-AV is already NP-hard for Condorcet's rule, we focus on the plurality rule. The parameters shown in the table are "the number m of alternatives", "the solution size k", "the maximum bundle size b", and "the maximum swap distance d between the leader and its followers in a bundle". We distinguish between arbitrary (the upper table) bundling functions and anonymous bundling functions (the lower table, including full-d). ILP-FPT means FPT based on a formulation as an integer linear program (see [Len83] and Theorem 2.2). "NP-c" means "NP-complete" while "W[1]-h (resp. W[2]-h)" means "W[1]-hard (resp. W[2]-hard)". The definitions of the parameterized complexity classes are given in Section 2.6.

that it is polynomial-time solvable when the given profile is single-crossing. Our results for the plurality rule are summarized in Table 6.1.

Finally, we remark that the combinatorial variants of the voter control problems that we study are clearly contained in NP. Thus, our NP-hardness results in fact imply NP-completeness results.

6.5 Canonical parameterizations

In this section we provide our results for profiles with unrestricted domains, that is, for the case where voters may have arbitrary preference orders. Later on, in Section 6.7, we will look at single-peaked and single-crossing profiles that only allow "reasonably restricted" preference orders.

6.5.1 Parameterization by the number of unregistered voters and by the solution size

We start our discussion by considering the parameters "number n of unregistered voters" and "solution size k". A simple brute-force algorithm, checking all possible combinations of k bundles, proves that for both the plurality and the Condorcet rules, C-CC-AV is in XP for parameter k, and in FPT for parameter n (the latter holds because $k \leq n$). Indeed, the same result holds for all voting rules with polynomial-time winner-determination procedures.

Proposition 6.5. *Let m be the number of alternatives, n' be the number of registered voters, n be the number of unregistered voters, and k be the solution size. For the plurality rule,* C-CC-AV *is solvable in $O(n^k \cdot k \cdot (n+n'))$ time. For the Condorcet rule,* C-CC-AV *is solvable in $O(n^k \cdot k \cdot (n+n') \cdot m^2)$ time. Thus, for both the plurality and the Condorcet rule,* CC-AV *is in* XP *for parameter k and in* FPT *for parameter n.*

Proof. We can solve C-CC-AV by considering all preference profiles resulting from adding one of the $\sum_{j=0}^{k} \binom{n}{j} \leq k \cdot n^k + 1$ possible combinations of (up to) k bundles of unregistered voters. For each combination of (up to) k bundles of voters, we use the standard winner determination algorithm for the given voting rule. For the plurality rule, the winner can be computed in time $O(n+n')$; for the Condorcet rule, the winner can be computed in time $O((n+n') \cdot m^2)$: for each pair of alternatives c and c', we compute whether a strict majority of voters prefers c to c' or c' to c (we can do so on a voter-by-voter basis by storing the results for each pair of alternatives). □

The above XP result parameterized by the solution size k probably cannot be improved to fixed-parameter tractability. Indeed, for parameter k we show that the problem is W[2]-hard, even for profiles with only two alternatives. This is quite remarkable because typically voting problems with a small number of alternatives are easy (they can be solved either by brute-force or by integer linear programming employing the famous FPT algorithm of Lenstra [Len83]; see Theorem 2.2 and see the surveys of Betzler et al. [Bet+12] and Bredereck et al. [Bre+14a] for examples). Furthermore, since our proof uses only two alternatives, it applies to almost all natural voting rules: for two alternatives almost all of them (including the Condorcet rule) are equivalent to the plurality rule. Also, every preference profile with two alternatives is trivially single-peaked and single-crossing, thus the next result extends to these domain restrictions as well.

Theorem 6.6. *For the plurality rule,* C-CC-AV *is* NP*-complete and* W[2]*-hard when parameterized by the solution size k, even for two alternatives.*

Proof. We first show the W[2]-hardness result by providing a parameterized reduction from the W[2]-complete problem Set Cover parameterized by the set cover size h [DF13] (also see Section 2.6.2 for more information):

> Set Cover
> **Input:** A family $\mathscr{F} = (S_1, \ldots, S_r)$ of sets over a universe $\mathscr{U} = \{u_1, \ldots, u_s\}$ of elements and a non-negative integer $h \geq 0$.
> **Question:** Is there a size-at-most-h *set cover*, that is, a collection $\mathscr{F}'$ of h sets in $\mathscr{F}$ whose union is $\mathscr{U}$?

Let $(\mathscr{U}, \mathscr{F}, h)$ be an instance of Set Cover, where $\mathscr{U} = \{u_1, \ldots, u_r\}$, and where $\mathscr{F} = (S_1, \ldots, S_s)$ is a collection of subsets of $\mathscr{U}$, and h is a non-negative integer. Without loss of generality, we assume that no set $S_i \in \mathscr{F}$ is empty and every element in U appears in at least one set $S_i \in \mathscr{F}$. We construct a preference profile (A, V) as follows. We let the set of alternatives be $A = \{p, g\}$, where p is our preferred alternative and g is the original winner. Since A has only two alternatives, when we speak of a p-voter (resp. a g-voter), we mean a voter with preference order $p \succ g$ (respectively, preference order $g \succ p$).

The registered voter set V consists of $|\mathscr{U}| + h$ voters with preference order $g \succ p$ (and no p-voters). The unregistered voter set W consists of two groups of voters which are all p-voters. The first group consists of a p-voter for each element $u_j \in \mathscr{U}$, called the element voter $w(u_j)$. We let the bundle of element voter $w(u_j)$ be a singleton, that is, $\kappa(w(u_j)) := \{w(u_j)\}$. The second group consists of a p-voter for each set $S_i \in \mathscr{F}$, called the set voter $w(S_i)$. We let the bundle of each set voter $w(S_i)$ consist of himself and all element voters $w(u_j)$ with $u_j \in S_i$, that is, $w(S_i) := \{w(S_j)\} \cup \{w(u_j) \mid u_j \in S_i\}$.

Observe that the plurality score of g is $|\mathscr{U}| + h$ while the plurality score of p is zero. In order to let p win, we have to add bundles to the original preference profile that will give $|\mathscr{U}| + h$ points. Finally, we set $k := h$.

It is clear that our construction runs in FPT time for the parameter k and it even runs in polynomial time. It remains to show that $(\mathscr{U}, \mathscr{F})$ has a size-at-most-h set cover if and only if there is a size-k subset $W' \subseteq W$ of unregistered voters such that p is a plurality winner in the profile $(A, V \cup \kappa(W'))$.

For the "if" part, suppose that there is a voter subset W' of size at most k such that p is a plurality winner of the profile $(A, V \cup \kappa(W'))$. Obviously, $\kappa(W') \geq |\mathscr{U}| + h$. Furthermore, if W' contains an element voter $w(u_j)$, then we can as well replace it with a set voter $w(S_i)$ that "contains" it, that is, $u_j \in S_i$. Thus, without loss of generality, we assume that W' consists of only set voters. Define $\mathscr{F}'$ to be the collection of sets that correspond to the set voters from W', that is, $\mathscr{F} := \{S_i \mid w(S_i) \in W'\}$. Then, $|\mathscr{F}'| \leq k = h$. Furthermore, for each element $u_j \in \mathscr{U}$ there must be a set $S_i \in \mathscr{F}$ which contains u_j since otherwise p will not obtain enough points to become a winner. This shows that $\mathscr{F}'$ is a set cover.

For the "only if" part, given a set cover $\mathscr{F}$ of size at most h we define W' to be the corresponding voter set, that is, $W' := \{w(S_i) \mid S_i \in \mathscr{F}\}$. It is easy to verify that $|W'| \leq h = k$, and p as well as g both win with $|\mathscr{U}| + h$ points.

As already mentioned, the parameterized reduction we presented is indeed a polynomial reduction. Since the problem we reduce from is NP-hard, we can conclude that our problem is NP-hard even for two alternatives. □

For two alternatives, the Condorcet rule is equivalent to the majority rule (see Definition 2.2). That is, one needs to add enough p-voters such that p is preferred to the second alternative by more than half of the voters. Thus, we can adapt the above reduction for the plurality rule to also work for the Condorcet rule.

Proposition 6.7. *For the Condorcet rule,* C-CC-AV *is* NP*-complete and* W[2]*-hard when parameterized by the solution size k, even for two alternatives.*

Proof. We use the same unregistered voters as described in the proof of Theorem 6.6, and we construct the input preference profile with $(|\mathscr{U}| + h - 1)$ g-voters. The correctness proof is along the same lines as the one for Theorem 6.6. □

As we can see, the bundles constructed in the proof of Theorem 6.6 are unbounded. If we require that each bundle has at most three voters and that the bundling function is anonymous, then for the plurality rule and when parameterized by the solution size k, C-CC-AV changes from being W[2]-hard to being W[1]-hard. It would be, however, interesting to know whether it is also contained in W[1].

Theorem 6.8. *For the plurality rule,* C-CC-AV *parameterized by the solution size k is* W[1]*-hard, even for anonymous bundling functions with bundles of size at most three.*

Proof. To show the W[1]-hardness result, we provide a parameterized reduction from the W[1]-complete problem Clique parameterized by the clique size h [DF13] (also see Section 2.6.2 for more information).

CLIQUE
Input: An undirected graph $G = (V(G), E(G))$ and a non-negative integer $h \in \mathbb{N}$.
Question: Does G admit a size-h *clique*, that is, a size-h vertex subset $U \subseteq V(G)$ such that $G[U]$ is complete?

We also call the number h of vertices in a clique the order of the clique.

Let (G, h) be a CLIQUE instance with $V(G)$ being the set of vertices and $E(G)$ being the set of edges. Without loss of generality, we assume that G is connected, $h \geq 3$, and each vertex in G has degree at least $h-1$. We construct a preference profile (A, V) as follows. We let the set A consist of three alternatives p, f, g, and an *edge alternative* c_e for each edge $e \in E(G)$. That is,

$$A := \{p, f, g\} \cup \{c_e \mid e \in E(G)\}.$$

Alternative p is our preferred alternative. We will construct voters such that f is the original plurality winner, and that there are sufficiently many unregistered g-voters to force us to add a "clique solution" to the profile. We use the edge alternatives from $\{c_e \mid e \in E(G)\}$ to ensure that all the unregistered voters have different preference orders, making the bundling function anonymous.

We introduce registered voters such that initially, f wins with $\binom{h}{2} + h$ points, g has $\binom{h}{2}$ points, our preferred alternative p has h points, and all edge alternatives have zero points. Formally, the registered voter set V consists of

- h voters, each with preference order $p \succ \langle A \setminus \{p\} \rangle$,
- $\binom{h}{2} + h$ voters, each with preference order $f \succ \langle A \setminus \{f\} \rangle$, and
- $\binom{h}{2}$ voters, each with preference order $g \succ \langle A \setminus \{g\} \rangle$.

In this way, we enforce that p needs at least $\binom{h}{2}$ points to become a winner. By carefully constructing the preference orders of the unregistered voters, we can enforce that the added voters correspond to a clique of size h. To this end, for each vertex $u \in V(G)$, we define $A(u) := \{c_e \mid e \in E(G) \wedge u \in e\}$. That is, $A(u)$ contains all edge alternatives that correspond to the incident edges of u. Now, we construct the set W of unregistered voters as follows (recall that for each set A, $\langle A \rangle$ denotes an arbitrary but fixed order):

(1) For each vertex $u \in V(G)$, we add an unregistered g-voter w_u with preference order

$$g \succ \langle A(u) \rangle \succ p \succ \langle A \setminus (\{g, p\} \cup A(u)) \rangle.$$

We call these unregistered voters *vertex voters*. We set $\kappa(w_u) := \{w_u\}$.

(2) For each edge $e = \{u, u'\} \in E(G)$, we add an unregistered p-voter w_e with preference order

$$p \succ c_e \succ g \succ \langle A \setminus \{p, g, c_e\} \rangle.$$

We call these unregistered voters *edge voters*. We set $\kappa(w_e) := \{w_u, w_{u'}, w_e\}$.

Note that all unregistered voters have different preference orders. This implies that our bundling function κ is anonymous (when all the unregistered voters have different preference orders, then every bundling function is anonymous). To complete our construction, we set $k := \binom{h}{2}$.

We show that graph G has a size-h clique if and only if there is a subset $W' \subseteq W$ of unregistered voters of size at most k such that p is a plurality winner in the profile $(A, V \cup \kappa(W'))$.

For the "if" part, suppose that there is a subset W' of at most k voters such that p is a plurality winner of the profile $(A, V \cup \kappa(W'))$. We show that the vertex set $U' := \{u \in V(G) \mid u \in e$ for some $w_e \in W'\}$ is a size-h clique for G. First, we observe that p needs at least $\binom{h}{2}$ points to become a winner because the initial winner f has exactly that amount of points more than p. By our construction, only bundles which include an edge voter can increase the score of p by adding one p-voter, while adding two additional g-voters. Since we can add at most $k = \binom{h}{2}$ bundles, we must add exactly k bundles of the edge voters. This means that $E(G[U'])$ contains k edges. However, in order to ensure p's victory, $\kappa(W')$ may only give at most h additional points to g. By the construction of the bundles of the edge voters, this means that U' contains at most h vertices. With $|E(G[U'])| \geq k$, we conclude that U' is of size h and, hence, is a size-h clique for G.

For the "only if" part, suppose that $U' \subseteq V(G)$ is a size-h clique for G. We construct the subset W' by adding to it all edge voters w_e with $e \in E(G[U'])$. Obviously, $|W'| = k$. We can verify that p co-wins with both f and g in the profile $(A, V \cup \kappa(W'))$ with score $\binom{h}{2} + h$. □

We conclude this section by mentioning that if we drop the anonymity, then we can adapt the proof of Theorem 6.8 to show that even for only two alternatives, our problem parameterized by the solution size remains W[1]-hard. Hence, we obtain the following.

Proposition 6.9. *For the Plurality rule,* C-CC-AV *is* W[1]*-hard when parameterized by the solution size k, even for three alternatives and bundle size at most three.*

Proof. We use almost the same reduction as given in the proof of Theorem 6.8 and we restrict the constructed profile to the three alternatives p, g, and f. The correctness proof for the plurality rule works analogously. □

6.5.2 Parameterization by the number of alternatives

From Theorem 6.6 and Proposition 6.7, we know that our central problem for both the plurality rule and the Condorcet rule is already NP-hard when the input profile has only two alternatives. The corresponding proofs use the non-anonymity of the bundling function in a crucial way. Indeed, if we require the bundling function to be anonymous, then C-CC-AV can be formulated as an integer linear program (ILP) where the number of variables and the number of constraints are bounded by some function dependent only on the number m of alternatives. Finding feasible solutions for such integer linear programs is in FPT, parameterized by the number of variables, due to the famous result of Lenstra [Len83].

Theorem 6.10. *Let m denote the number of alternatives, n' denote the number of registered voters, and n denote the number of unregistered voters. Let $\mathsf{ilp}(\rho_1, \rho_2, \rho_3)$ denote the running time[2] of the feasibility problem of an integer linear program which has ρ_1 variables and ρ_2 constraints, and where the maximum of the absolute values of the coefficients and the constant terms is ρ_3, $\rho_1, \rho_2, \rho_3 \in \mathbb{N}$. For the plurality and the Condorcet rules,* C-CC-AV *can be solved in $\mathsf{ilp}(2m!, 3n + m, \max(n, n'))$ time.*

Proof. In a given profile $\mathcal{P}$ with a set A of m alternatives, there are at most $m!$ voters with pairwise different preference orders. Since the bundling function is anonymous and, hence, follower-anonymous, there are at most $m!$ different bundles. Furthermore, we can assume that all voters in a solution W' have pairwise different preference orders (this is because, due to (leader) anonymity, there is no additional gain in adding two voters with the same preference order). Due to this, we can use an integer linear program (ILP) to check whether such a solution W' exists.

Let n' and n denote the number of registered and unregistered voters, respectively. Let Π be the set of the linear preference orders of the unregistered voters. For each alternative $a \in A$, we write $F(a)$ to denote the set of preference orders in which a is ranked first. For each linear preference order $\succ$, we define the following.

- Let $N(\succ)$ be the number of voters in W that have preference order $\succ$.

[2]As already noted in Theorem 2.2, the best known running time for this setting is $O(\rho_1{}^{2.5 \cdot \rho_1 + o(\rho_1)} \cdot \rho_1 \cdot \rho_2 \cdot \log(\rho_3 + 2))$ [FT87, Kan87, Len83].

- We introduce two Boolean variables, $x(\succ)$ and $y(\succ)$. The intended meaning of $x(\succ) = 1$ is that the sought solution W' contains voters with preference order $\succ$. The intended meaning of $y(\succ) = 1$ is that $\kappa(W')$ contains voters with preference order $\succ$. In our ILP, we use the values of the variables $x(\succ)$ to enforce the correct values of the variables $y(\succ)$.
- In what admittedly is a slight abuse of notation, we write $\kappa(\succ)$ to denote the set of preference orders of the voters included in the bundle of a voter with preference order $\succ$. Further, we define $\kappa^{-1}(\succ) = \{\succ' \mid \succ \in \kappa(\succ')\}$ to be the set of preference orders that "include" $\succ$ in their bundles.

Now we are ready to state our integer linear program (note that it suffices to find a feasible solution and, thus, we do not specify any function to minimize). Recall that for each alternative a, $\text{score}_{\mathscr{P}}(a)$ denotes the number of voters in profile $\mathscr{P}$ that rank a in the first position.

$$\sum_{\succ \in \Pi} x(\succ) \leq k, \tag{6.1}$$

$$0 \leq x(\succ) \leq 1, \qquad \forall \succ \in \Pi, \tag{6.2}$$

$$0 \leq y(\succ) \leq 1, \qquad \forall \succ \in \Pi, \tag{6.3}$$

$$x(\succ) \leq N(\succ), \qquad \forall \succ \in \Pi, \tag{6.4}$$

$$\sum_{\succ' \in \kappa^{-1}(\succ)} x(\succ') \leq n \cdot y(\succ), \qquad \forall \succ \in \Pi, \tag{6.5}$$

$$y(\succ) \leq \sum_{\succ' \in \kappa^{-1}(\succ)} x(\succ'), \qquad \forall \succ \in \Pi, \tag{6.6}$$

$$\text{score}_{\mathscr{P}}(a) + \sum_{\succ \in F(a)} N(\succ) \cdot y(\succ) \leq \text{score}_{\mathscr{P}}(p) + \sum_{\succ \in F(p)} N(\succ) \cdot y(\succ), \qquad \forall a \in A \setminus \{p\}. \tag{6.7}$$

We explain the meaning of the different constraint sets. First, constraint set (6.1) ensures that at most k voters are added to the solution W'. constraint sets (6.2) and (6.3) ensure that the introduced variables are Boolean. constraint set (6.4) ensures that the voters added to W' are indeed present in W. constraint sets (6.5) and (6.6) ensure that variables $y(\succ)$, $\succ \in \Pi$, have correct values. Indeed, if for some preference order $\succ'$ with $\succ' \in \kappa^{-1}(\succ)$, we have $x(\succ') = 1$, then constraint set (6.5) ensures that $y(\succ) = 1$. Otherwise, that is, for each preference order $\succ'$ with $\succ' \in \kappa^{-1}(\succ)$ it holds that $x(\succ') = 0$. Thus, constraint set (6.6) ensures that $y(\succ) = 0$. Finally, constraint set (6.7) ensures that p has plurality score at least as high as every other alternative (and, thus, is a winner). Clearly, there is a solution for this integer linear program if and only if there is a solution for the input instance.

For the case of the Condorcet rule, we modify constraint set (6.7). Instead of comparing the plurality scores of the alternatives, we rewrite it to compare how many voters prefer p to each given alternative a. To this end, for each alternative $a \in A$, let $B(p,a)$ denote the set of all preference orders in which p is ranked in front of a and let $\#_{\mathscr{P}}(p,a)$ be the number of voters in the original profile $\mathscr{P}$ that prefer p to a.

The following constraint set

$$\#_{\mathscr{P}}(p,a) + \sum_{\succ \in B(p,a)} N(\succ) \cdot y(\succ) \ > \ \frac{1}{2}\Big(n' + \sum_{\succ \in \Pi} N(\succ) \cdot y(\succ)\Big), \qquad \forall a \in A \setminus \{p\} \tag{6.8}$$

ensures that for each alternative $a \in A \setminus \{p\}$, more than half of the voters from the original profile and from the added voter set $\kappa(W')$ must prefer p to a, which means that p is a Condorcet winner. If we use this constraint to replace constraint (6.7), then we obtain an ILP for the Condorcet rule.

To analyze the running time, we only need to analyze the size of our ILPs: both ILPs have at most $2m!$ variables and at most $3n + 5m$ constraints, and in our ILPs, the absolute value of each coefficient and each constant term is at most $\max(n', n)$. □

As already stated in Footnote 2 and noted in Theorem 2.2, using the famous result of Lenstra [Len83] (later improved by Kannan [Kan87] and Frank and Tardos [FT87]), we can derive from Theorem 6.10 the following tractability result because an integer linear program with ρ_1 variables and ρ_2 constraints, whose coefficients and constant terms are between $-\rho_3$ and ρ_3, can be encoded $O(\rho_1 \cdot \rho_2 \cdot \log(\rho_3 + 2))$ bits.

Corollary 6.11. *For the plurality and the Condorcet rules,* C-CC-AV *is in* FPT *for the parameter "number m of alternatives".*

We close this section by mentioning that unless some complexity-theoretic assumption (that is, $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$) fails, for the case that the bundling function is anonymous, C-CC-AV parameterized by the number m of alternatives does not admit a polynomial-size kernel (see Section 2.6.1 for the definition). That is, no polynomial-time algorithm can, given any instance of C-CC-AV with m alternatives, compute an equivalent instance whose size is polynomial in m. The reason for the absence of such polynomial kernels is that adapting the proof of Theorem 6.6, we can construct a polynomial-parameter transformation from SET COVER parameterized by the size of the universe: we simply make the preferences of each element voter unique by adding an alternative with the same name, that is ranked between p

and g, and we proceed similarly with the set voters. Since unless $\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$, Set Cover parameterized by the size of the universe does not admit a polynomial kernel [DLS14], by Proposition 2.1 we obtain that C-CC-AV also does not admit a polynomial kernel for the parameter m.

6.6 Parameterization by the maximum bundle size and by the swap distance

We now focus on the complexity of C-CC-AV for the plurality rule as a function of two combinatorial parameters:

(a) the maximum size b of each voter's bundle, and

(b) the maximum swap distance d between the leader and her followers in one bundle.

By Observation 6.4, for the Condorcet rule the problem is NP-hard already for $b = 1$, so we do not consider the Condorcet rule in this section. Throughout this section, when we speak of C-CC-AV, we mean C-CC-AV for the plurality rule.

Specifically, we show that for the plurality rule, C-CC-AV is polynomial-time solvable if the maximum bundle size is one (that is, if we are in the non-combinatorial setting already studied by Bartholdi III, Tovey, and Trick [BTT92]), but it is NP-hard already when the maximum bundle size is two and the bundling function κ is anonymous. We also show that when the maximum bundle size is two, the problem for arbitrary bundling functions, parameterized by the solution size, is in FPT. In contrast, if the bundling function κ is a full-d (that is, if each bundle contains all the voters at swap distance at most d from the leader), then the problem is polynomial-time solvable if the maximum bundle size is two, but it is NP-hard already when the maximum bundle size is three and the maximum swap distance is three.

6.6.1 Bundle size at most two—an intractability result

First, if $b = 1$, then as mentioned in Observation 6.3, C-CC-AV reduces to CC-AV and, thus, can be solved in linear time [BTT92]. Indeed, one only needs to calculate the score difference of the preferred alternative and the current winner and check whether k, the maximum number of voters one may add, is at least as large as this difference and whether there are enough p-voters from the unregistered voter set to add to the preference profile.

C-CC-AV becomes intractable as soon as the maximum bundle size b is two, even for anonymous bundling functions. To show this we reduce from the following restricted variant of 3SAT.

(2-2)-3SAT
Input: A collection $\mathcal{C}$ of clauses over the variable set $\mathcal{X} = \{x_1, \ldots, x_n\}$, where each clause has either two or three literals, and each variable appears exactly four times, twice as a positive literal and twice as a negative literal.
Question: Is there a truth assignment that satisfies all clauses in $\mathcal{C}$?

As we will show next, this variant remains NP-hard.

Lemma 6.12. (2-2)-3SAT *is* NP*-complete.*

Proof. Clearly, the problem belongs to NP. We provide a polynomial reduction from a variant version of the NP-complete 3SAT problem, where each clause has either *two* or *three* literals, each variable occurs either two or three times, and at most one time as a negative literal [Tov84, Theorem 2.1].

First, we assume that no variable appears only as a positive literal: if this was the case for some variable, then we could set it to true and simplify the formula. For each variable x_i that appears three times (two times positively and one time negatively), we add one new variable y_i, and two new clauses $\{\neg x_i, \neg y_i, y_i\}$ and $\{\neg y_i, y_i\}$. For each variable x_i that appears two times (one time positively and one time negatively), we add one new clause $\{\neg x_i, x_i\}$. In this way, we obtain a new instance, where each variable appears two times positively and two times negatively. It is easy to see that the original instance is a yes-instance if and only if the newly constructed instance is a yes-instance for (2-2)-3SAT. ☐

Theorem 6.13. *For the plurality rule,* C-CC-AV *is* NP*-complete even if the bundling function is anonymous and the maximum bundle size b is two.*

Proof. We reduce from the NP-complete problem (2-2)-3SAT (Lemma 6.12). Given a (2-2)-3SAT instance $(\mathcal{C}, \mathcal{X})$, where $\mathcal{C}$ is as set of clauses and $\mathcal{X}$ is a set of variables, we construct an input preference profile (A, V) as follows. We add to the set A our preferred alternative p and an initial winner, called g. Then, for each clause $C_j \in \mathcal{C}$, we add to A one *clause alternative* c_j and four *dummy alternatives*, $d_j^{(1)}, d_j^{(2)}, d_j^{(3)}, d_j^{(4)}$. That is,

$$A := \{p, g\} \cup \{c_i \mid C_i \in \mathcal{C}\} \cup \{d_j^{(1)}, d_j^{(2)}, d_j^{(3)}, d_j^{(4)} \mid x_j \in \mathcal{X}\}.$$

We use the clause alternatives to make sure that the solution to an input instance encodes a satisfying truth assignment, and the dummy alternatives to ensure that all unregistered voters have distinct preference orders (this implies that our bundling function is anonymous).

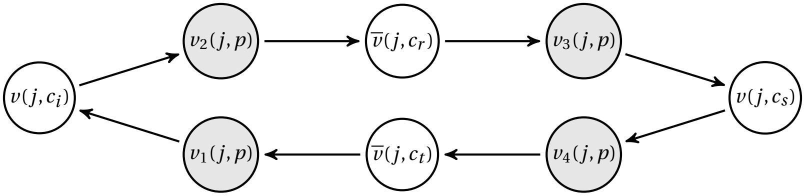


Figure 6.2.: Part of the construction used in Theorem 6.13. Specifically, we show the cycle corresponding to variable x_j which occurs as a positive literal in clauses C_i and C_s, and as a negative literal in clauses C_r and C_t.

We construct the set V of registered voters so that the initial score of g is $4|\mathcal{X}|$, the initial score of each clause alternative c_i is $4|\mathcal{X}| - |C_i| + 1$ (where $|C_i|$ is the number of literals that clause C_i contains), and the initial score of p is zero. We assume without loss of generality that no clause contains the same literal more than once.

We construct the set W of unregistered voters as follows (throughout the rest of the proof, we will often write ℓ_j to refer to a literal that contains variable x_j; depending on the context, ℓ_j will mean either x_j or $\neg x_j$ and the exact meaning will always be clear). For each variable $x_j \in \mathcal{X}$ that occurs as a positive literal x_j in clauses C_i and C_s, $i < s$, and as a negative literal $\neg x_j$ in clauses C_r and C_t, $r < t$, we create

(1) four p-voters, denoted by $v_1(j,p), v_2(j,p), v_3(j,p), v_4(j,p)$, with the following preference orders:

$$\text{voter } v_1(j,p)\colon\ p \succ d_j^{(1)} \succ \langle C \setminus \{p, d_j^{(2)}\}\rangle,$$
$$\text{voter } v_2(j,p)\colon\ p \succ d_j^{(2)} \succ \langle C \setminus \{p, d_j^{(3)}\}\rangle,$$
$$\text{voter } v_3(j,p)\colon\ p \succ d_j^{(3)} \succ \langle C \setminus \{p, d_j^{(4)}\}\rangle,$$
$$\text{voter } v_4(j,p)\colon\ p \succ d_j^{(4)} \succ \langle C \setminus \{p, d_j^{(1)}\}\rangle;$$

we call these voters *variable* voters, and

(2) four *clause voters*, denoted by $v(j,c_i), \overline{v}(j,c_r), v(j,c_s), \overline{v}(j,c_t)$, with the following preference orders:

$$\text{voter } v(j,c_i)\colon\ c_i \succ d_j^{(1)} \succ \langle C \setminus \{c_i, d_j^{(1)}\}\rangle,$$
$$\text{voter } \overline{v}(j,c_r)\colon\ c_r \succ d_j^{(2)} \succ \langle C \setminus \{c_r, d_j^{(2)}\}\rangle,$$
$$\text{voter } v(j,c_s)\colon\ c_s \succ d_j^{(3)} \succ \langle C \setminus \{c_s, d_j^{(3)}\}\rangle,$$
$$\text{voter } \overline{v}(j,c_t)\colon\ c_t \succ d_j^{(4)} \succ \langle C \setminus \{c_t, d_j^{(4)}\}\rangle.$$

Note that each clause C_i has exactly $|C_i|$ corresponding clause voters.

We now describe our bundling function κ whose corresponding bundling graph will contain a cycle for each variable (see Figure 6.2 for an illustration). Formally, we define κ as follows:

$$\begin{aligned}
\kappa(v_1(j,p)) &:= \{v_1(j,p), v(j,c_i)\}, & \kappa(v(j,c_i)) &:= \{v(j,c_i), v_2(j,p)\},\\
\kappa(v_2(j,p)) &:= \{v_2(j,p), \overline{v}(j,c_r)\}, & \kappa(\overline{v}(j,c_r)) &:= \{\overline{v}(j,c_r), v_3(j,p)\},\\
\kappa(v_3(j,p)) &:= \{v_3(j,p), v(j,c_s)\}, & \kappa(v(j,c_s)) &:= \{v(j,c_s), v_4(j,p)\},\\
\kappa(v_4(j,p)) &:= \{v_4(j,p), \overline{v}(j,c_t)\}, & \kappa(\overline{v}(j,c_t)) &:= \{\overline{v}(j,c_t), v_1(j,p)\}.
\end{aligned}$$

We complete the reduction by setting the number of unregistered voters which will be added to the profile, to four times the number of variables in $\mathcal{X}$, $k := 4|\mathcal{X}|$. Obviously, the reduction runs in polynomial time.

To show the correctness of the construction, the general idea is that in order to let p win, all unregistered p-voters must be in $\kappa(W')$ and no clause alternative c_i should gain more than $(|C_i|-1)$ points. Formally, we show that $(\mathcal{C},\mathcal{X})$ has a satisfying truth assignment if and only if there is a size-k voter subset $W' \subseteq W$ such that p wins the election $(A, V \cup \kappa(W'))$ (recall that $k = 4|\mathcal{X}|$).

For the "if" part, let $\beta \colon \mathcal{X} \to \{\text{true}, \text{false}\}$ be a satisfying truth assignment function for $(\mathcal{C},\mathcal{X})$. Intuitively, β will guide us through constructing the voter set W' in the following way: First, for each variable x_j and for each clause that contains the literal ℓ_j but β sets it to false, we add to W' the corresponding clause voter. For instance, if $\beta(x_j) = \text{false}$ and $x_j \in C_i$, then we add $v(j,c_i)$ to W'. This way, we include $2|\mathcal{X}|$ p-voters in $\kappa(W')$, and, for each clause C_i, we include at most $(|C_i|-1)$ c_i-voters as well. The former is true because exactly $|\mathcal{X}|$ literals are set to false by β, each literal is included in exactly two clauses, and adding the corresponding clause voter to W' also includes a unique p-voter in $\kappa(W')$; the latter is true because if β is a satisfying truth assignment, then each clause C_i contains at most $(|C_i|-1)$ literals set to false. Then, for each clause voter v already in W', we also add the voter $v_z(j,p)$, $1 \le z \le 4$, that contains v in his bundle. This way, we include $2|\mathcal{X}|$ additional p-voters in $\kappa(W')$ without including additional clause voters. Formally, we define W' as follows:

$$\begin{aligned}
W' := &\big\{\overline{v}(j,c_i), v_z(j,p) \mid \beta(x_j) = T \wedge \neg x_j \in C_i \wedge v(j,c_i) \in \kappa\big(v_z(j,p)\big)\big\} \cup \\
&\big\{v(j,c_i), v_z(j,p) \mid \beta(x_j) = F \wedge x_j \in C_i \wedge v(j,c_i) \in \kappa\big(v_z(j,p)\big)\big\}.
\end{aligned}$$

Following our intuitive argument, one can verify that all p-voters are contained in $\kappa(W')$, and that each clause alternative c_i gains at most $(|C_i|-1)$ points.

For the "only if" part, let W' be a subset of voters such that p is a plurality winner in the profile $(A, V \cup \kappa(W'))$.

We say that a literal ℓ_j is *selected* if at least one voter $v(j, c_i)$ is *not* in $\kappa(W')$. If, for some variable x_j, all four voters $v(j, c_i)$ are in $\kappa(W')$, then we arbitrarily set variable x_j to be selected. Intuitively, selecting a literal means that it should be set to true to satisfy the formula. Just as not all literals in a clause C_i can be set to false, it is not allowed to include all corresponding clause voters $v(j, c_i)$ in $\kappa(W')$.

Before we define the truth assignment, we first prove that for each variable x_j, literals x_j and $\neg x_j$ cannot be selected simultaneously. This is clear in the special case where all four clause voters $v(j, c_i)$ are in $\kappa(W')$. Now we focus on the case where not all four clause voters $v(j, c_i)$ are in $\kappa(W')$. Recall that $\kappa(W')$ must contain all unregistered p-voters in order to let p win. This implies that for each two clauses that contain the same variable but not the same literal, at least one corresponding clause voter must be added to $\kappa(W')$. Thus, if one clause voter is not contained in $\kappa(W')$, then both of his "neighboring" (in the sense of having distance two to this clause voter in the bundling graph, as depicted in Figure 6.2) clause voters must be included in $\kappa(W')$. That is, for each variable x_j, bundle $\kappa(W')$ must contain at least the two voters of the form $v(j, c_i)$ or the two voters of the form $\overline{v}(j, c_i)$. In turn, this means that only one of x_j and $\neg x_j$ is selected.

We now define the truth assignment $\beta\colon \mathcal{X} \to \{\text{true}, \text{false}\}$ such that $\beta(x_j) := \text{true}$ if literal x_j is selected and $\beta(x_j) := \text{false}$ if $\neg x_j$ is selected. Following the previous arguments, function β is well-defined. It is a satisfying truth assignment function for $(\mathcal{C}, \mathcal{X})$ because for each clause C_i, by the fact that p is a plurality winner in the profile $(A, V \cup \kappa(W'))$, we have that for each clause alternative c_i, $\kappa(W')$ contains at most $(|C_i| - 1)$ c_i-voters. There must be some p-voter that corresponds to a literal ℓ_j from c_i which is not contained in $\kappa(W')$. Hence, ℓ_j is selected. Thus, this literal is set to true via β, and clause C_i is satisfied.

Overall, the formula is satisfiable if and only if it is possible to make p win by adding $2|\mathcal{X}|$ bundles, which completes the reduction. □

Taking a closer look at the proof of Theorem 6.13, we observe that the bundling graph we obtain in the reduction consists only of cycles (disregarding the directions of the arcs). One natural question is whether this is crucial for the hardness result. In fact, in a collaboration with Gerhard Woeginger (TU Eindhoven) at the 2015 Dagstuhl seminar on "Computational Social Choice: Theory and Applications", we could show that C-CC-AV remains NP-complete even if the corresponding bundling graph consists of trees (disregarding the directions of the arcs) of height at most two. We achieve this by reducing from another restricted variant of 3SAT, where each clause contains at most three literals and each variable appears exactly three times, twice as a positive literal and once as a negative literal. This problem variant is also NP-complete. We omit the corresponding proof.

6.6.2 Bundle size at most two—two tractability results

While we have just shown that C-CC-AV is NP-hard for the plurality rule even if each bundle has at most two voters, intuitively it seems plausible that some tractability should exist for this case, as the setting is very restrictive. We prove this intuition right by showing the following results.

1. We give a fixed-parameter algorithm for the problem, parameterized by the solution size k (Theorem 6.17).

2. We give a polynomial-time algorithm for the case where the bundling function is full-d (Theorem 6.20).

Both results rely on the fact that, for $b = 2$, we can work with the corresponding bundling graph in the following way. Intuitively, we should select arcs containing as many p-vertices as possible. Hence, we first find a largest subset of disjoint arcs whose both endpoints are p-vertices. We then update the bundles (and the corresponding bundling graph) and add to the solution all p-voters whose bundles are singletons. Finally, in the case of a non-full-d bundling function, we brute-force search through the bundling graph structure corresponding to the remaining part of the solution in "FPT"-time. In the case where the bundling function is full-d, we can solve the remaining problem greedily in polynomial time.

To implement our ideas, we first need some notions and observations regarding the structure of the bundling graphs in our instances. Throughout the remainder of the discussion of the case of $b = 2$, let $I = ((A, V), W, \kappa, p, k)$ be a C-CC-AV instance, and let $G = (V(G), E(G))$ be the bundling graph for I. We assume that this is a yes-instance and we let W' be a solution for I (that is, a subset of unregistered voters) of a size not greater than k. We say that a solution is minimal if it has the smallest size among all solutions, and we focus on such solutions.

Since $b = 2$, each bundle corresponds to one of the following four different bundle types:

(1) bundles consisting of two p-voters; the corresponding bundling graph notion for this type is a (p, p)-arc;

(2) bundles consisting of exactly one p-voter; the corresponding bundling graph notion for this type is a p-vertex with no outgoing arcs;

(3) bundles consisting of one p-voter and one non-p-voter; the corresponding bundling graph notion for this type is either a $(p,\text{non-}p)$-arc or a $(\text{non-}p,p)$-arc; and

(4) bundles not containing any p-voter.

Clearly, we never need to add bundles of the last type to our solution. Thus, we only need to take care of the bundles of the first three types and, without loss of generality, we assume that G *does not* contain any arc between two non-p-voters. Further, we can consider the first three bundle types independently, in the same order as they are listed above. The next two lemmas formalize this observation.

Lemma 6.14. *If a solution W' contains a c-voter w' whose bundle $\kappa(w')$ is* not *of type (1) but there is at least one p-voter $w \in W \setminus \kappa(W')$ whose bundle $\kappa(w)$ is of type (1), then $W'' := (W' \setminus \{w'\}) \cup \{w\}$ is also a solution of the same size.*

Proof. Clearly, $|W'| = |W''|$. Let $\mathscr{P}' := (A, V \cup \kappa(W'))$ and let $\mathscr{P}'' := (A, V \cup \kappa(W''))$. Now observe that the plurality score of p in profile $\mathscr{P}''$ is at least as high as the plurality score of p in profile $\mathscr{P}'$, and that every other alternative has at most the same plurality score in $\mathscr{P}''$ as in $\mathscr{P}'$. □

Using the same reasoning as for Lemma 6.14, we can show a similar result regarding bundles of type (3) and type (2).

Lemma 6.15. *If a solution W' contains a p-voter w' whose bundle $\kappa(w')$ is of type (3) but there is at least one p-voter $w \in W \setminus \kappa(W')$ whose bundle $\kappa(w)$ is of type (2), then $W'' := (W' \setminus \{w'\}) \cup \{w\}$ is also a solution of the same size.*

Proof. The proof of this result works in the same way as the proof of Lemma 6.14. □

We need the following result to preprocess trivial yes-instances and to upper-bound by k the number of p-vertices adjacent to a given vertex.

Lemma 6.16. *Suppose that all the bundles are of type (3). If a bundling graph G contains (as a subgraph) a k-star whose center is a c-vertex for some alternative $c \in C \setminus \{p\}$ and whose leaves are p-vertices, then the following holds.*

1. *If c has at least k points more than p, then no solution can contain the leader of any bundle corresponding to an arc of the star.*

2. *Otherwise, provided that p can become a winner at all, this star corresponds to a solution of size k.*

Proof. Suppose that c has at least k points more than p. Since all bundles are of type (3), adding at most k bundles increases the score of p by at most k points. Adding even one bundle corresponding to an arc from the star increases the score of c by one, making c have more points than p again. Thus, if it is only allowed to add at most k bundles, then adding a bundle corresponding to an arc from the star makes it impossible for p to be a plurality winner.

In contrast, if c has at most $(k-1)$ points more than p, then the best we can do is to add all leaders of the k bundles corresponding to the k arcs of the star. This increases the score of p by k (the highest possible value given that all bundles are of type (3)) and keeps the score of c less than or equal to that of p. If it is possible to make p win, then adding these k bundles makes p win. □

Using these observations, we are ready to show our two main results for the case with maximum bundle size two (Theorems 6.17 and 6.20).

Theorem 6.17. *Let m be the number of alternatives, n' be the number of registered voters, n be the number of unregistered voters, and k the solution size. If the maximum bundle size b is two, then for the plurality rule,* C-CC-AV *parameterized by the solution size k can be solved in $O(m \cdot n' + m \cdot n^2 + k^{4k} \cdot m \cdot n)$ time.*

Proof. Following the discussion preceding this proof, our algorithm starts by picking as many disjoint bundles of types (1) and (2) as possible. To do so, we first find a maximum matching for the corresponding bundling graph restricted to the (p, p)-arcs, and add the bundles corresponding to this matching (we also update the bundling function to take this into account; indeed, some bundles may change type). Then, we add as many bundles with exactly one p-voter as possible (that is, bundles of type (2)). By Lemmas 6.14 and 6.15, this greedy approach is correct. From now on, we assume that our bundles are only of type (3) (and we assume that our value k is modified, taking into account all the bundles we have added so far).

We now make several observations regarding our updated bundling graph G. We assume that it is possible to ensure p's victory, and we consider a specific minimum-size solution, denoted by W'. We let G' be the subgraph of G that corresponds to the solution W'. Formally, $G' = (V(G'), E(G'))$, where $V(G') = V(G)$, and

$$E(G') := \{(u, v) \in E(G) \mid u \text{ corresponds to a leader } x \text{ in } W' \text{ such that } v \text{ corresponds to a follower in } \kappa(x)\}.$$

Claim 6.18. *For each p-vertex in G', the sum of its in-degree and its out-degree is at most one.*

Proof of Claim 6.18. Towards a contradiction, assume that there is a p-vertex u for which the sum of its in-degree and its out-degree is at least two. Let u_1 and u_2 be two neighbors of u (note that at least one of them is an in-neighbor). Since all the bundles are of type (3), neither u_1 nor u_2 is a p-vertex because their common neighbor u is already a p-vertex. Let $v_1, v_2 \in W$ be the voters to which u_1 and u_2 correspond, respectively. At least one of them must be a leader (that is, must belong to W'). It is easy to verify that removing this voter from W' results in a correct solution of smaller size, which contradicts the assumption that W' is minimal. (of Claim 6.18) $\diamond$

By Claim 6.18, we can conclude that G' consists of stars (ignoring the directions of the arcs).

Claim 6.19. *Each connected component of G' has the following properties:*

1. *It is a star.*
2. *The center of the star corresponds to a non-p-voter and all leaves to p-voters.*
3. *The center of the star has at most one out-arc and at most k in-arcs.*

Proof of Claim 6.19. Let F be a connected component of G' (see Section 2.2.2 for the definition of connected components in directed graphs). We first show that component F contains at most one non-p-vertex. Towards a contradiction, suppose that F contains two non-p-vertices, u and u'. These two vertices cannot be adjacent because we only have bundles of type (3). Thus, there is a path between u and u' that contains at least one p-vertex (we ignore the direction of the arcs on this path). However, the sum of the in-degree and the out-degree of a p-vertex on such a path would be two, which by Claim 6.18 is impossible. Thus, F contains at most one non-p-vertex. Since for every p-vertex the sum of its in-degree and its out-degree is at most one (Claim 6.18), F is a star and we can consider the p-vertices as the leaves. Hence, we consider the non-p-vertex as the center (if the star has only two vertices, one p-vertex and one non-p-vertex, then we define the non-p-vertex as the center). The last part of our claim follows from Lemma 6.16 and the fact that every bundle has size two. (of Claim 6.19) $\diamond$

Based on these two claims, we derive a search-tree algorithm for the case where we are left with bundles of type (3) only. That is, the bundling graph corresponding to a minimum solution consists only of stars where the centers are non-p-voters. We start with an empty graph and we keep adding "good" stars to it, one by one. To formalize this idea, we need one more notion: Given two graphs, G_1 and G_2, such

Algorithm 6.1: The main part of our algorithm for the case that all bundles consist of exactly one p-voter and one non-p-voter (Type (3)).

1 Find a best i^{th} star S if it exists; otherwise backtrack to the last step where the $(i-1)^{\text{th}}$ star has not yet been added to M.

2 There are three possibilities of the relation between S and a M-extending bundling graph F; we guess which one of them actually applies:

(a) F contains S. Let S be the i^{th} star and add it to M, decrease the budget r by the number of arcs in S.

(b) There is a star with index j, $j > i$, that has not needed any alternative so far. Guess the value of j and mark alternative c as needed by the j^{th} star. If the number of stars that need any alternative exceeds the budget, then backtrack to the beginning of Step (2) and follow the the other branches.

(c) There is a star with index j, $j > i$, that will use some *non-reserved* p-vertex v from S. Guess the value of j and guess the vertex v from S. Reserve v for the j^{th} star. If all p-vertices from S have been reserved, then backtrack to the beginning of Step 2 and branch into other possibilities.

that G_2 is a subgraph of G_1, by $G_1 \setminus G_2$ we mean the directed graph obtained from G_1 by deleting all arcs from G_2 followed by deleting all isolated vertices.

Now, we are ready to describe our algorithm. First, we guess the size k' ($0 \le k' \le k$) of a minimum-size solution (this value equals the additional score that p will gain). Then, we begin with an empty graph M and a budget r, initialized to k'. We will repeatedly add "good" stars to the graph M. Thus, when we speak of the j^{th} star, we mean the j^{th} star that is, or will be, added to M. During the algorithm, we occasionally mark some of the p-vertices as *reserved* for some of the stars that are yet to be added to M. Similarly, we occasionally mark some alternatives as *needed* by some of the to-be-added stars. Intuitively, if a p-vertex is reserved for a star, it means that this star uses this vertex as a leaf. When an alternative is needed by a star, it means that this star's center must correspond to this alternative.

The main part of our algorithm executes two steps in a loop, until the whole budget has been spent. We describe this approach in Algorithm 6.1: After having found a "good" star in the first step, the algorithm uses brute-force to search each of three possible branchings, depending on the relation between the found star and the partial solution.

Initially our graph M is empty and the budget r is set to k'. Let i be the number of stars in M added so far, plus one (so, initially, $i = 1$). Let us introduce two more concepts for the "partial solution" M we computed so far. The first one describes a property of a star.

Definition 6.5 (Best i^{th} stars)**.** A star S is a *best i^{th}* star regarding M if it satisfies the following; let the center of S correspond to the alternative c, $c \neq p$.

(1) S contains all p-vertices that have been reserved for the i^{th} star but no p-vertices that have been reserved for any j^{th} star with $j > i$. No alternative $c' \neq c$ is marked as needed for the i^{th} star.

(2) The original score of c plus the number of occurrences of the c-vertices in M plus the number of times that c is needed for some j^{th} stars, $j > i$, is less than the original score of p plus k', and

(3) S has the largest—but not larger than $(r - t)$—number of leaves among all stars in $G \setminus M$ fulfilling the first two conditions, where t is the number of p-vertices that are reserved for some j^{th} stars with $j > i$.

The next concept describes a property of a bundling graph.

Definition 6.6 (M-extending solution graphs)**.** Given two bundling graphs M and F, we say that F is an M-extending solution graph if it satisfies the following.

(1) F contains M.

(2) F is the bundling graph of a size-k' solution.

(3) F respects the current *reservation requirements*, that is, every reserved p-vertex is contained in G', no star in G' contains two vertices that are reserved for different stars, and all p-vertices that have been reserved for the same star j are also contained in the same star of G'.

(4) F respects the current *center requirements*, that is, for each alternative c with $c \neq p$, it holds that the number of times that c is needed is at most the number of c-vertices in G'.

We show the correctness of our algorithm by the following inductive argument about S. Let the center of S be a c-vertex with $c \neq p$ and let s denote the number of leaves in S. Suppose that there is an M-extending bundling graph F. We show that there is also an M-extending bundling graph H such that one of the three branchings in Step (2) applies to H. Obviously, if F contains the center of S, then we can verify that there is a bundling graph H satisfying Conditions (1)–(4) and containing S (Branching (a) in Algorithm 6.1).

Now assume that bundling graph F does not contain the center of S. We distinguish three cases regarding the intersection between F and S (we use the following notation: If a star has some b-vertex as the center, $b \neq p$, then we say that this is a b-centered star):

1. Suppose that bundling graph F does not contain any vertex from S. Let S_1 be a c-centered star in $F \setminus M$ if $F \setminus M$ contains one. Otherwise, let S_1 be an empty graph.

   a) If no p-vertex in S_1 is reserved or if S_1 is empty, then let S_2 be a supergraph of S_1 and a subgraph of $F \setminus M$ such that the number of p-vertices in S_2 is s. that are not reserved for any j^{th} star with $j > i$ (note that such subgraph exists because S satisfies Condition (3) from Definition 6.5). We can verify that the graph $F' := (F \setminus S_2) \cup S$ contains S and satisfies Conditions (1)–(4) from Definition 6.6. Thus, Branching (a) applies to graph F'.

   b) Otherwise, S_1 contains a reserved p-vertex, denote by u. By assumption, F does not contain any vertex of S. Since S satisfies Condition (1) of Definition 6.5, we know that no p-vertex in F is reserved for the i^{th} star. This implies that u is reserved for a star j with $j > i$ and that we can guess one value $j > i$ and mark the alternative c as needed for the star j. Thus, Branching (b) applies to graph F.

2. Suppose that the graph $F \setminus M$ contains two stars S_1 and S_2, $S_1 \neq S_2$, such that each star S_ℓ, $\ell \in \{1,2\}$, contains a vertex v_ℓ from S.

   a) If v_1 (resp. if v_2) is reserved, then v_1 (resp. if v_2) is reserved for the i^{th} star (Condition (1) of Definition 6.5). This implies that v_2 (resp. v_1) is not reserved. Then, we can guess one value j, $j > i$, and guess one not yet reserved vertex v from S, and mark v as reserved for the j^{th} star.

   b) Otherwise, both v_1 as well as v_2 are not reserved. Since S_1 and S_2 cannot both be the i^{th} star, we can guess one value j, $j > i$, and guess one not yet reserved vertex v from S, and mark v as reserved for the j^{th} star.

   In both cases, Branching (c) applies to graph F.

3. In this last case, suppose that the graph $F \setminus M$ contains only one star S_1 which contains some p-vertex u from S. Let the center of S_1 be a c_1-vertex.

a) If $c = c_1$ and if no p-vertex in S_1 is reserved for a star j with $j > i$, then let S_2 be a supergraph of S_1 and a subgraph of $F \setminus M$ such that the number of p-vertices in S_1 that are not reserved for the j^{th} star with $j > i$ is s. Such subgraph exists because S satisfies Condition (3) We can verify that the bundling graph $F' := (F \setminus S_2) \cup S$ contains S and satisfies Conditions (1)–(4) from Definition 6.6. Thus, Branching (a) applies to graph F'.

b) If S_1 contains a p-vertex which is reserved for some star j', $j' > i$, then this vertex cannot be u because S satisfies Condition (1) from Definition 6.5. Thus, we can guess one value j, $j > i$, and guess one not yet reserved vertex v from S, and mark v as reserved for the j^{th} star. This implies that Branching (c) applies to F.

c) Otherwise, $c \neq c'$ and no p-vertex in S_1 is reserved. Then, S_1 contains at most s arcs since S satisfies Condition (3) from Definition 6.5.

   i. If c's original score plus the number of occurrences of the c-vertices in F is less than p's original score plus k', then let S_2 be a supergraph of S_1 and a subgraph of $F \setminus M$ such that S_2 consists of exactly s p-vertices that are not reserved for any star j with $j > i$ (note that such subgraph exists because S satisfies Condition (3) of Definition 6.5). We compare graph F with the following graph $F' := (F \setminus S_2) \cup S$. Both graphs, F and F', contain the same number of p-vertices. The number of c-vertices in F is one less than the number of c-vertices in F'. This means by assumption that adding the bundles corresponding to F' will not make c have more points than p. That is, F' is also an M-extending graph and contains S. Thus, Branching (a) applies to F'.

   ii. Otherwise, $F \setminus M$ contains at least one c-centered star S_2 as S satisfies Condition (2) of Definition 6.5. Since S_1 and S_2 cannot both be the i^{th} star, we can guess one value j with $j > i$, and do the following. We mark c as needed for the j^{th} star or we guess one not yet reserved vertex v from S and mark v as reserved for the j^{th} star. Then, Branching (b) or Branching (c) applies to F.

We have shown the correctness of our algorithm, including the part shown in Algorithm 6.1. To see the running time, recall that m denotes the number of alternatives, n the number of registered voters, n' the number of unregistered voters, and k the number of unregistered voters we are allowed to add. Constructing the bundling graph for a given instance runs in $O(m \cdot (n')^2)$ time. The constructed graph has at most n' vertices and at most n' arcs. Computing the original score of each alternative runs in $O(m \cdot n)$ time. Preprocessing bundles of type (1) by finding a maximum matching runs in $O((n')^{3/2})$ time [MV80]. Preprocessing bundles of type (2) runs in $O(n')$ time. For the running time of the search-tree algorithm, note that S has at most k arcs and there are at most $k-1$ additional stars that need an alternative. This leads to a total of $1 + k \cdot (k-1) + (k-1) = k^2$ possible guesses in Step (2). Moreover, after guessing the minimum solution size k', we build a search-tree algorithm which has depth at most $2k$ (the number of stars in M plus the number of stars that need an alternative is at most k while the number of reserved p-vertices is at most k) and has branching factor k^2. This means that our search tree has size at most $(k^2)^{2k}$. In each branching node, we can find a best i^{th} star S in $O(m \cdot n')$ time. Thus, the combined running time is $O(m \cdot n' + m \cdot n^2 + k^{4k} \cdot m \cdot n)$. □

If we require the bundling function to be full-d, then we obtain a polynomial-time algorithm by extending the greedy algorithm of Bartholdi III, Tovey, and Trick [BTT92].

Theorem 6.20. *Let m be the number of alternatives, n' be the number of registered voters, and n be the number of unregistered voters. If the maximum bundle size b is two and the bundling function κ is full-d, then* C-CC-AV *for the plurality rule is solvable in $O(m \cdot (n + n'))$ time.*

Proof. As in the proof of Theorem 6.17, we first select as many p-vertices as possible without any non-p vertex. Since in our C-CC-AV instance $((A, V), W, \kappa, p, k)$ every bundle has at most two voters and κ is a full-d bundling function, every two bundles are either equal or disjoint. Thus, by Lemmas 6.14 and 6.15, we can greedily select all disjoint bundles of type (1) and then all disjoint bundles of type (2) (note that by selecting a bundle we mean to add the leader of the bundle to the solution; since both voters corresponding to a bundle are leaders of the same bundle, we can choose either of them). By Lemma 6.14 and Lemma 6.15 and by the fact that all bundles are either equal or disjoint, this greedy approach is correct. Thus, from now on, we assume that our bundles are only of type (3).

Afterwards, the algorithm sorts (in ascending order) the remaining bundles by the score of the non-p-voter in each bundle (remember that we have only bundles of type (3)) and then adds the bundles in this order.

The correctness of the second part of the algorithm follows because all bundles are either equal or disjoint.

For the running time, calculating the scores of all alternatives costs $O(m \cdot n')$ time. Adding bundles with two p-voters or with one p-voter and throwing away all irrelevant bundles costs $O(m \cdot n)$. Sorting the bundles can be done in $O(m \cdot (n + n'))$

time as the values are upper-bounded by the instance size. Looping over the sorted alternatives, for each bundle possibly shifting the current alternative higher in the sorted sequence, can be done in $O(m \cdot n')$ time. Thus, the total running time is $O(m \cdot (n + n'))$. □

6.6.3 Bundle Size Three or More

For a bundle size of $b = 3$, we obtain NP-completeness even for full-d bundling functions with $d \leq 3$. The proof is similar to the one of Theorem 6.13.

Theorem 6.21. *For the plurality rule,* C-CC-AV *is* NP*-complete even for full-d bundling functions with constant value $d \geq 3$ and maximum bundle size $b = 3$.*

Proof. We give a polynomial reduction from (2-2)-3SAT. This reduction is almost the same as the one given in the proof of Theorem 6.13. The main difference is that we carefully construct the voters' preference orders so that the bundling function is full-3 and each bundle consists of at most three voters. This leads to some technical differences, but the main idea of the construction remains the same.

Let $(\mathscr{C}, \mathscr{X})$ be our input instance of (2-2)-3SAT, where $\mathscr{C}$ is the set of clauses over the variables from the set $\mathscr{X}$. We construct the same set of alternatives as in the proof of Theorem 6.13. That is, we set $A := \{p, g\} \cup \{c_i \mid C_i \in \mathscr{C}\} \cup \{d_j^{(1)}, d_j^{(2)}, d_j^{(3)}, d_j^{(4)} \mid x_j \in \mathscr{X}\}$. As in the proof of Theorem 6.13, we construct the set of registered voters so that the initial score of alternative g is $8|\mathscr{X}|$, the initial score of each clause alternative c_i is $8|\mathscr{X}| - |C_i| + 1$ (where $|C_i|$ is the number of literals in C_i), and the initial scores of all the other alternatives are zero.

We construct the set of unregistered voters as follows. For each variable $x_j \in \mathscr{X}$ that occurs as a positive literal x_j in some clauses C_i and C_s, $i < s$, and as a negative literal $\neg x_j$ in some clauses C_r and C_t, $r < t$, we introduce eight unregistered p-voters, $v_z(j,p)$, $1 \leq z \leq 8$, and four unregistered clause voters, $v(j,c_i), \overline{v}(j,c_r), v(j,c_s), \overline{v}(j,c_t)$. Their preference orders are defined as follows, where $R(j)$ denotes the set $A \setminus \{p, c_i, c_r, c_s, c_t, d_j^{(1)}, d_j^{(2)}, d_j^{(3)}, d_j^{(4)}\}$:

$$\text{voter } v_1(j,p)\colon\ p \succ c_t \succ c_i \succ c_r \succ c_s \succ d_j^{(1)} \succ d_j^{(2)} \succ d_j^{(3)} \succ d_j^{(4)} \succ \langle R(j) \rangle,$$

$$\text{voter } v_2(j,p)\colon\ p \succ c_i \succ c_r \succ c_s \succ c_t \succ d_j^{(1)} \succ d_j^{(2)} \succ d_j^{(3)} \succ d_j^{(4)} \succ \langle R(j) \rangle,$$

$$\text{voter } v_3(j,p)\colon\ p \succ c_i \succ c_r \succ c_s \succ c_t \succ d_j^{(2)} \succ d_j^{(4)} \succ d_j^{(3)} \succ d_j^{(1)} \succ \langle R(j) \rangle,$$

$$\text{voter } v_4(j,p)\colon\ p \succ c_r \succ c_s \succ c_t \succ c_i \succ d_j^{(2)} \succ d_j^{(4)} \succ d_j^{(3)} \succ d_j^{(1)} \succ \langle R(j) \rangle,$$

$$\text{voter } v_5(j,p)\colon\ p \succ c_r \succ c_s \succ c_t \succ c_i \succ d_j^{(3)} \succ d_j^{(4)} \succ d_j^{(1)} \succ d_j^{(2)} \succ \langle R(j) \rangle,$$

$$\text{voter } v_6(j,p)\colon\ p \succ c_s \succ c_t \succ c_i \succ c_r \succ d_j^{(3)} \succ d_j^{(4)} \succ d_j^{(1)} \succ d_j^{(2)} \succ \langle R(j)\rangle,$$

$$\text{voter } v_7(j,p)\colon\ p \succ c_s \succ c_t \succ c_i \succ c_r \succ d_j^{(4)} \succ d_j^{(2)} \succ d_j^{(1)} \succ d_j^{(3)} \succ \langle R(j)\rangle,$$

$$\text{voter } v_8(j,p)\colon\ p \succ c_t \succ c_i \succ c_r \succ c_s \succ d_j^{(4)} \succ d_j^{(2)} \succ d_j^{(1)} \succ d_j^{(3)} \succ \langle R(j)\rangle,$$

$$\text{voter } v(j,c_i)\colon\ c_i \succ p \succ c_r \succ c_s \succ c_t \succ d_j^{(2)} \succ d_j^{(1)} \succ d_j^{(4)} \succ d_j^{(3)} \succ \langle R(j)\rangle,$$

$$\text{voter } \overline{v}(j,c_r)\colon\ c_r \succ p \succ c_s \succ c_t \succ c_i \succ d_j^{(4)} \succ d_j^{(3)} \succ d_j^{(2)} \succ d_j^{(1)} \succ \langle R(j)\rangle,$$

$$\text{voter } v(j,c_s)\colon\ c_s \succ p \succ c_t \succ c_i \succ c_r \succ d_j^{(4)} \succ d_j^{(1)} \succ d_j^{(3)} \succ d_j^{(2)} \succ \langle R(j)\rangle,$$

$$\text{voter } \overline{v}(j,c_t)\colon\ c_t \succ p \succ c_i \succ c_r \succ c_s \succ d_j^{(1)} \succ d_j^{(4)} \succ d_j^{(2)} \succ d_j^{(3)} \succ \langle R(j)\rangle.$$

In this way, we obtain the following properties of the swap distances.

(1) For each number $z \in \{1,2,\ldots,4\}$, it holds that:
   (a) Voters $v_{2z-1}(j,p)$ and $v_{2z}(j,p)$ have a swap distance of three.
   (b) For each number $z' \in \{1,2,\ldots,8\} \setminus \{2z-1,2z\}$, voters $v_{2z-1}(j,p)$ and $v_{2z}(j,p)$ have a swap distance of at least four to $v_{z'}(j,p)$, respectively.

(2) For each literal ℓ_j and each clause C_y that contains ℓ_j, let v be the clause voter that corresponds to C_y and ℓ_j. Then the following holds:
   (a) The swap distance between voter v and voter $v_z(j,p)$ who ranks alternative c_y at the second place is exactly three (one swap between p and c_y and two swaps among alternatives $d_j^{(1)}$, $d_j^{(2)}$, $d_j^{(3)}$, and $d_j^{(4)}$).
   (b) The swap distance between voter v and some other clause voter is at least four (because of the clause alternatives c_i, c_r, c_s, and c_t, and p).
   (c) The swap distance between voter v and every p-voter $v_z(j,p)$, $z \in \{1,2,\ldots,8\}$ who ranks c_y below the second place is at least four (because of the clause alternatives c_i, c_r, c_s, and c_t, and p).

Note that the swap distance between two voters who correspond to two different variables is much larger than three. We define our bundling function to be a full-3 bundling function whose bundling graph is depicted in Figure 6.3. We obtain the following values of the bundling function for variable x_j:

$$\kappa(v_1(j,p)) := \{v_1(j,p), v_2(j,p), \overline{v}(j,c_t)\}, \quad \kappa(v_2(j,p)) := \{v_2(j,p), v_1(j,p), v(j,c_i)\},$$
$$\kappa(v_3(j,p)) := \{v_3(j,p), v_4(j,p), v(j,c_i)\}, \quad \kappa(v_4(j,p)) := \{v_4(j,p), v_3(j,p), \overline{v}(j,c_r)\},$$

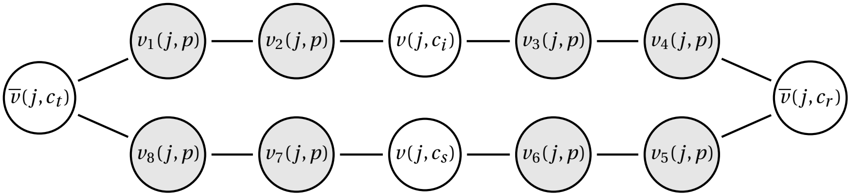


Figure 6.3.: Part of the construction used in the proof of Theorem 6.21. Specifically, we show the cycle corresponding to variable x_j which occurs as a positive literal in clauses C_i and C_s, and as a negative literal in clauses C_r and C_t.

$$\begin{aligned}
\kappa(v_5(j,p)) &:= \{v_5(j,p), v_6(j,p), \overline{v}(j,c_r)\}, & \kappa(v_6(j,p)) &:= \{v_6(j,p), v_5(j,p), v(j,c_s)\},\\
\kappa(v_7(j,p)) &:= \{v_7(j,p), v_8(j,p), v(j,c_s)\}, & \kappa(v_8(j,p)) &:= \{v_8(j,p), v_7(j,p), \overline{v}(j,c_t)\},\\
\kappa(v(j,c_i)) &:= \{v(j,c_i), v_2(j,p), v_3(j,p)\}, & \kappa(\overline{v}(j,c_r)) &:= \{\overline{v}(j,c_r), v_4(j,p), v_5(j,p)\},\\
\kappa(v(j,c_s)) &:= \{v(j,c_s), v_6(j,p), v_7(j,p)\}, & \kappa(\overline{v}(j,c_t)) &:= \{\overline{v}(j,c_t), v_8(j,p), v_1(j,p)\}.
\end{aligned}$$

Finally, we set $k := 4|\mathscr{X}|$.

The proof of the correctness is, in essence, the same as the one of Theorem 6.13. If there is a satisfying truth assignment $\beta\colon \mathscr{X} \to \{\text{true}, \text{false}\}$ for our input instance, then we can derive a solution to our problem as follows. We say that a clause is *failed* by literal ℓ_j if either this clause contains ℓ_j and $\beta(\ell_j) = \text{false}$, or this clause contains $\neg\ell_j$ and $\beta(\ell_j) = \text{true}$. For each literal ℓ_j, we include in our solution W' these two clause voters who correspond to the clauses failed by ℓ_j. Further, we include those four p-voters whose bundles each contain at least one of these two clause voters. Since each literal is contained in exactly two clauses, we can easily verify that the defined solution W' contains exactly $4|\mathscr{X}|$ voters and that it gives $8|\mathscr{X}|$ additional points to p. Since every clause C_i is failed by at most $|C_i|-1$ literals, each clause alternative obtains at most $|C_i|-1$ additional points. Thus, altogether, p, g, and all clause alternatives tie for victory.

The proof for the other direction is the same as in the case of Theorem 6.13. In particular, given a solution $\mathscr{P}'$, we declare a literal ℓ_j to be selected if at least one of the corresponding clause voter is not included in $\kappa(W')$; the proof that, if W' is a solution for our control problem, then setting the selected literals to true leads to a satisfying truth assignment, proceeds as the one of Theorem 6.13. □

If we relax the bundle size to be at most four, then we find that C-CC-AV is NP-hard even if the maximum swap distance is one. This stands in contrast to the case where $d = 0$, where C-CC-AV reduces to the CC-AV problem for weighted elections [FHH15], which, for the plurality rule, is polynomial-time solvable by a simple greedy algorithm.

Theorem 6.22. *For the plurality rule,* C-CC-AV *is* NP*-complete even for full-d bundling functions with constant value $d \geq 1$ and maximum bundle size $b = 4$.*

Proof. We can use the same construction as in the W[1]-hardness proof of Theorem 6.25 (see the next section) to provide a polynomial-time reduction from the Vertex Cover problem instead of the Partial Vertex Cover problem; see more details right after the proof of Theorem 6.25. The maximum swap distance remains the same; that is, it is one. Since the maximum bundle size is exactly one higher than the maximum vertex degree of the Vertex Cover instance, and since Vertex Cover is NP-hard [GJS76] already when the maximum vertex degree is three, the correctness of Theorem 6.22 follows. □

6.7 Single-peaked and single-crossing cases

In this section, we consider the computational complexity of C-CC-AV for the restricted case where the profiles are single-peaked or single-crossing. We note that the W[2]-hardness result for non-anonymous bundling functions and for two alternatives (Theorem 6.6) also extends to these restricted domains because every profile with two alternatives is single-peaked and single-crossing.

Observation 6.23. *For the plurality rule and the Condorcet rule, for single-peaked and single-crossing profiles,* C-CC-AV *parameterized by the solution size is* W[2]*-hard, even for two alternatives.*

Moreover, by carefully crafting the preference orders for three alternatives, the W[1]-hardness result (Proposition 6.9) for three alternatives and a maximum bundle size of three also extends to these restricted domains. Thus we obtain the following.

Observation 6.24. *For the plurality rule, for single-peaked and single-crossing profiles,* C-CC-AV *parameterized by the solution size is* W[1]*-hard, even for two alternatives and with bundles size of at most three.*

In the remainder of this section, we focus on instances with full-d bundling functions; via Proposition 6.2, our hardness results extend to anonymous and hence, also to arbitrary bundling functions. Our results for the combinatorial variant of control with anonymous bundling functions under our domain restrictions are quite different from those for the non-combinatorial case. Indeed, both for the plurality rule and the Condorcet rule, the non-combinatorial voter control problems for single-peaked (resp. single-crossing) profiles are solvable in polynomial time [Bra+15, Fal+11, MF14] (for the case of plurality, this is true even in the unrestricted case). For the combinatorial case, we show hardness for both voting rules even when the input profile is single-peaked, but we give polynomial-time algorithms for the single-crossing case. We begin with the case of single-peaked profiles.

Theorem 6.25. *For both the plurality rule and the Condorcet rule, even when the input profile is single-peaked and the bundling function κ is full-d with constant $d = 1$, C-CC-AV is NP-complete and when parameterized by the solution size k, it is W[1]-hard.*

Proof. We consider the plurality rule first. We show both NP-hardness and W[1]-hardness results by providing a single reduction from the NP-complete and W[1]-hard problem PARTIAL VERTEX COVER (PVC). The parameter for PARTIAL VERTEX COVER is the size h of the vertex cover [GNW07]:

> PARTIAL VERTEX COVER (PVC)
> **Input:** An undirected graph $G = (V(G), E(G))$ and two non-negative integers $h, \ell \in \mathbb{N}$.
> **Question:** Does G admit a size-h vertex subset $U \subseteq V(G)$ which intersects at least ℓ edges in G?

To obtain the desired results, we will prove that the reduction can be done in polynomial time and in FPT time.

Let (G, h, ℓ) be our input instance for PVC where $V(G)$ denotes the vertex set and $E(G)$ denotes the edge set. We set the parameter $k := h$, and construct an input preference profile (A, V) as follows. The alternative set A contains our preferred alternative p and an initial winner, called g. For each vertex u_i in $V(G)$, we add to set A two *vertex alternatives*, called a_i and $\overline{a}_i$. The introduction of these two vertex alternatives is to ensure that restricting the bundling function to be full-3 causes relevant voters to be "bundled" together while each bundle still has at most three voters. Formally, we have

$$A := \{p, g\} \cup \{a_i, \overline{a}_i \mid u_i \in V(G)\}.$$

Before constructing the voters and their preference orders, we first define the following *canonical preference order*:

$$\langle A \rangle := p \succ g \succ a_1 \succ \overline{a}_1 \succ \ldots \succ a_{|V(G)|} \succ \overline{a}_{|V(G)|}.$$

Based on this order we define further preference orders. For each set P of disjoint pairs of alternatives such that the alternatives in each pair are consecutive with respect to the canonical preference order, we define the preference order diff-order(P) to be the preference order that differs from our canonical order by exactly the pairs in P.

We are now ready to construct the registered and unregistered voters. The registered voter set V consist of $(h+\ell)$ voters, each ranking the alternatives according to the canonical preference order. In this way, the initial score of g is $(h+\ell)$ and the scores of all the other alternatives are zero. Thus, g is a unique winner.

We define the unregistered voter set W as follows:

(1) For each edge $e=\{u_i,u_j\}\in E(G)$, we construct an *edge voter* $w(e,p)$ with preference order diff-order$(\{\{a_i,\overline{a}_i\},\{a_j,\overline{a}_j\}\})$. We say that $w(e,p)$ corresponds to edge e. Note that all edge voters are p-voters.

(2) For each edge $e=\{u_i,u_j\}\in E(G)$, we construct a *dummy voter* $d(e,g)$ with preference order diff-order$(\{\{p,g\},\{a_i,\overline{a}_i\},\{a_j,\overline{a}_j\}\})$. We also say that $d(e)$ corresponds to edge e. Note that all dummy voters are g-voters.

(3) For each vertex $u_i\in V(G)$, we construct a *vertex voter* $w(i,p)$ with preference order diff-order$(\{\{a_i,\overline{a}_i\}\})$. We say that $w(i,p)$ corresponds to u_i. Note that all vertex voters are p-voters.

We have constructed $(h+\ell)$ registered and $2|V(G)|+|E(G)|$ unregistered voters. Their preference orders are single-peaked with respect to the following order:

$$\overline{a}_{|V(G)|}\succ\overline{a}_{|V(G)|-1}\succ\ldots\succ\overline{a}_1\succ p\succ g\succ a_1\succ a_2\succ\ldots\succ a_{|V(G)|}.$$

Finally, we define the bundling function κ to be full-1. To understand how κ works, we calculate the swap distances between the preference orders of all possible pairs of voters in W. We see the following.

(a) The distance between each two edge voters is at least two.

(b) The distance between each edge voter and each dummy voter is exactly one if they correspond to the same edge, and is at least three otherwise.

(c) The distance between each edge voter $w(e,p)$ and each vertex voter $w(i,p)$ is one if $u_i\in e$, and three otherwise.

(d) The distance between each two dummy voters is at least two.

(e) The distance between each dummy voter and each vertex voter is at least two.

(f) The distance between each two vertex voters is two.

To make κ full-1, we define κ as follows.

- For each edge $e = \{u_i, u_j\} \in E(G)$, we define the bundles of the corresponding edge voter and the dummy voter as $\kappa(w(e,p)) := \{w(e,p),\ w(i,p),\ w(j,p),\ d(e,g)\}$, $\kappa(d(e,g)) := \{d(e,g), w(e,p)\}$.
- For each vertex $u_i \in V(G)$, we define the bundle of the corresponding vertex voter as $\kappa(w(i,p)) := \{w(i,p)\} \cup \{w(e,p) \mid u_i \in e \in E(G)\}$.

Note that by the construction of the bundling function, adding a dummy voter is never better than adding his "corresponding" edge voter.

Obviously, the construction runs in $O((h+\ell+|V(G)|+|E(G)|)\cdot|V(G)|)$ time. It remains to show that graph G has a size-h vertex subset $U \subseteq V(G)$ which intersects at least ℓ edges in G if and only if there is a size-k voter subset $W' \subseteq W$ such that p is a plurality winner of the profile $(A, V \cup \kappa(W'))$. Recall that all unregistered voters except for the dummy voters are p-voters and that p needs at least $(h+\ell)$ points in order to win.

For the "only if" part, suppose that $X \subseteq V(G)$ is a size-h vertex set and $Y \subseteq E(G)$ is a size-ℓ edge set such that for every edge $e \in Y$ it holds that $e \cap X \neq \emptyset$. We set $W' := \{w(u_i) \mid u_i \in X\}$. It is easy to verify that $\kappa(W')$ consists of h vertex voters and at least ℓ edge voters. Each of them gives p one point if added to the profile. This results in p being a winner of the resulting profile.

For the "if" part, suppose that there is a size-k subset $W' \subseteq W$ such that p is a plurality winner of the profile $(A, V \cup \kappa(W'))$. Observe that if W' contains some dummy voter $d(e,g)$, then we can replace him with voter w_e. If w_e is already in set W', then we can simply remove $d(e,g)$ from W'. Thus, we can assume that set W' does not contain any dummy voters. Now, assume that W' contains some edge voter $w(e,p)$ with $e = \{u_i, u_j\}$. Since, by the previous argument, W' does not contain $d(e,g)$, we know that $d(e,g)$ is not in $\kappa(W' \setminus \{w(e,p)\})$. This means that if both $w(i,p)$ and $w(j,p)$ belong to $\kappa(W' \setminus \{w_e\})$, then we can safely remove $w(e,p)$ from W'; p will still be a winner of the profile $(A, V \cup \kappa(W' \setminus \{w(e,p)\}))$. On the contrary, assume that exactly one of $w(i,p)$ and $w(j,p)$ is not in $\kappa(W' \setminus \{w(e,p)\})$; without loss of generality, let $w(i,p)$ be the voter not in the mentioned set. It is easy to verify that p is a winner of profile $(A, V \cup \kappa((W' \setminus \{w(e,p)\}) \cup \{w(i,p)\}))$ (the net effect of including the bundle of $w(e,p)$ is that p's score increases by at most one, whereas the net effect of including the bundle of $w(i,p)$ is that p's score increases by at least one). Similarly, if neither $w(i,p)$ nor $w(j,p)$ with $i < j$ are in $\kappa(W' \setminus \{w(e,p)\})$, then it is easy to verify that p

is still a winner of the profile $(A, V \cup \kappa((W' \setminus \{w(e,p)\}) \cup \{w(i,p)\}))$. All in all, we can assume that W' contains vertex voters only. Since all vertex voters are p-voters, without loss of generality we can assume that W' contains exactly $k = h$ of them.

We define vertex set $X := \{u_i \mid w(i,p) \in W'\}$ and edge set $Y := \{e \in E(G) \mid e \cap X \neq \emptyset\}$. By the construction of the edge voters' preference orders, $\kappa(W')$ consists of k vertex voters and $|Y|$ edge voters. This must add up to at least $(h+\ell)$ voters. Therefore, $|Y| \geq \ell$, implying that at least ℓ edges are covered by X. This completes the proof for the case of plurality.

For the Condorcet rule, we use the same unregistered voter set W and construct the original profile with $(h+\ell)-1$ registered voters whose preference orders are diff-order($\{g,p\}$). We set $k := h$. Since all voters rank either p or g in the first position, the Condorcet rule equals the plurality rule for the unique-winner model. Thus, using the same reasoning as used for the plurality rule, we can verify that (G,h,ℓ) is a yes-instance of PVC if and only if there is a size-k subset $W' \subseteq W$ such that p is a Condorcet winner of the profile $(A, V \cup \kappa(W'))$.

In fact, our reductions also a polynomial-time reduction. Since PARTIAL VERTEX COVER is NP-complete, we simultaneously obtain NP-hardness results for both voting rules. □

We can easily verify that VERTEX COVER is a special case of PARTIAL VERTEX COVER where the number ℓ of edges to "cover" is the total number of edges $|E(G)|$ of the input graph G. Thus, we can use the same construction given in the proof of Theorem 6.25 for the case of $\ell = |E(G)|$ to provide a polynomial reduction from VERTEX COVER to our combinatorial voter control problem. In the obtained instance, the maximum bundle size equals the maximum vertex degree of the reduced graph G plus one. Since VERTEX COVER is already NP-hard when the maximum vertex degree is three, we can conclude that C-CC-AV is NP-hard even when the maximum bundle size is four.

We now consider the single-crossing case (see Section 3.4 for more information on this property). First, we make some observations about the single-crossing property.

Lemma 6.26. *If a preference profile is single-crossing, then for each alternative c, all c-voters appear consecutively in each single-crossing order of the voters.*

Proof. Consider a single-crossing profile with n voters and let $\sigma\colon x_1 \succ x_2 \succ \ldots \succ x_n$ be a single-crossing voter order. Suppose for the sake of contradiction that there are three consecutive voters x_i, x_{i+1}, and x_{i+2}, $1 \leq i \leq n-2$ such that x_i and x_{i+2} are

c-voters but x_{i+1} is a c'-voters with $c' \neq c$. Then, the order σ is not single-crossing with respect to the couple $\{c, c'\}$—a contradiction. $\square$

If we also require the bundling function to be full-d, then all voters in a bundle must appear consecutively and any solution for the plurality rule has a nice property we will use later on.

Lemma 6.27. *Let* $I = ((A, V), W, \kappa, p, k)$ *be a* C-CC-AV *instance such that* $(A, V \cup W)$ *is single-crossing and* κ *is full-*d*. The following holds.*

(1) The voters of each bundle appears consecutively in each single-crossing order of the voters from W*.*

(2) If I *is a yes-instance for the plurality rule, then there is a subset* $W' \subseteq W$ *of size at most* k *such that (a)* p *is a plurality winner in profile* $(A, V \cup \kappa(W'))$*, and (b) all bundles of voters* $w \in W'$ *contain only* p*-voters, except at most two bundles which may contain some non-*p*-voters.*

Proof. Consider a follower w' in the bundle of an unregistered voter w and consider a third voter w'' which is positioned between w and w' along a single-crossing order. Then, by the definition of the single-crossing property, we know that every couple that separates voters w from w'' also separates voters w from w'. By the definition of swap distances, this implies that the swap distance between w and w'' is at most as large as the one between w and w'. Since κ is full-d, we have $w'' \in \kappa(w)$, implying the first statement.

We show the second statement by modifying an existing solution for I. Let σ be a single-crossing order of the voters in W'. Let $W' \subseteq W$ be a size-k subset of the unregistered voters such that p is a plurality winner in the profile $(A, V \cup \kappa(W'))$. Without loss of generality, we assume that the bundle of each voter in W' contains at least one p-voter and no two voters in W' have the same bundle.

Suppose that there are two voters w and w' with different bundles from the solution W' such that their bundles each contains a non-p-voter and the first p-voter along the order σ.

Let $\kappa(w, 1)$ and $\kappa(w', 1)$ denote the first voter in $\kappa(w)$ and in $\kappa(w')$ along the order σ, respectively. Without loss of generality, we assume that $\kappa(w, 1)$ is *not* behind $\kappa(w', 1)$ in the order σ. We distinguish two cases regarding the number of p-voters in the bundles $\kappa(w)$ and $\kappa(w')$. If $\kappa(w)$ contains more p-voters than $\kappa(w')$, then by the first statement we know that $\kappa(w) \supseteq \kappa(w')$. We can safely remove w' from W' and obtain a smaller solution. Otherwise, $\kappa(w)$ does not contain more p-voters than $\kappa(w')$. By the first statement and by the fact that $\kappa(w, 1)$ is not behind $\kappa(w', 1)$, we know that each non-p-voter contained in $\kappa(w')$ is also contained in $\kappa(w)$. Thus, removing w from W' does not decrease the number of points given to p and does not increase the number of points given to any other alternative $c \neq p$. Using the same reasoning repeatedly, we can conclude that W' contains at most one voter w whose bundle $\kappa(w)$ contains a non-p-voter and the first p-voter along the single-crossing order.

Along the same lines, we can show that W' contains at most one voter w whose bundle $\kappa(w)$ contains a non-p-voter and the last p-voter along the single-crossing order. To complete the proof, it remains to observe that for each unregistered voter $w \in W$, if $\kappa(w)$ contains both some p-voters and some non-p-voters, then it also must contain either the first p-voter or the last p-voter along the single-crossing order. Altogether, this implies that W' contains at most two voters whose bundles contain non-p-voters. □

For the Condorcet rule, we use the concept of median voters and we apply the well-known median-voter theorem due to Black [Bla48] (we provide the proof for the sake of completeness). Given a voter order $\sigma := \langle v_1, v_2, \ldots, v_n \rangle$, let the *median voter set* of σ, denoted as Med(σ), be the set $\{v_{\ell+1}, v_{\ell+2}, \ldots, v_{n-\ell} \mid \ell = \lfloor \frac{n-1}{2} \rfloor\}$. Note that the size of Med(()σ) is one for odd n and two for even n.

Lemma 6.28 ([Bla48])**.** *Let $\mathcal{P}$ be a single-crossing profile and let σ denote such a single-crossing order of the voters in $\mathcal{P}$. An alternative c is a Condorcet winner of $\mathcal{P}$ if and only if all voters in Med(σ) are c-voters.*

Proof. By the definition of median voters, observe that along the order σ, the number of voters ordered in front of Med(σ) and the number of voters ordered behind Med(σ) are the same; they are both $\lfloor \frac{n-1}{2} \rfloor$.

For the "only if" part, suppose that c is a Condorcet winner. Assume towards a contradiction that there is a voter in Med(σ), called w, which prefers another alternative c' to c. Since c is a Condorcet winner which implies that c beats c', it must hold that some voter in front of w and some other voter behind w prefer c to c'—a contradiction to σ being single-crossing regarding couple $\{c, c'\}$.

For the "if" part, suppose that all voters in Med(σ) are c-voters. Now consider an arbitrary alternative $c' \neq c$. Then, all voters in Med(σ) prefer c to c'. Since σ is a single-crossing order, couple $\{c, c'\}$ separates σ at most once. That is, it cannot happen that some voter in front of Med(σ) and some other voter behind Med(σ) prefer c' to c. Thus, at most $\lfloor \frac{n-1}{2} \rfloor$ voters from the whole voter set prefer c' to c, implying that c beats c' for every alternative $c' \neq c$. Thus, c is a Condorcet winner. □

With these three lemmas available, we are ready to describe the following polynomial-time algorithms for both the plurality rule and the Condorcet rule.

Theorem 6.29. *For the single-crossing case with full-d bundling functions,* C-CC-AV *is solvable in* $O((n+n')\cdot m^2 + n^2)$ *for the plurality rule, and it is solvable in* $O((n+n')\cdot m^2 + (n+n')^2 \cdot n^4)$

Proof. First, we find a single-crossing order for profile $(A, V \cup W)$ in $O((n+n')\cdot m^2)$ time ([EFS12] and Theorem 3.9). By Lemma 6.27 (2) and Lemma 6.28, we only need to store the most preferred alternative of each voter in order to find a solution. Thus, the running time from now on only depends on the number n of unregistered voters. We start with the plurality rule and let $\sigma\colon x_1 \succ x_2 \succ \ldots \succ x_n$ be a single-crossing order of the unregistered voters.

Due to Lemma 6.27 (2), we may assume that our solution contains at most two voters whose bundles each contain non-p-voters. We first guess these two voters, and after this initial guess, the bundles of all remaining voters in the solution contain only p-voters (Lemma 6.27 (1)). Thus, the remaining task is to find the maximum score that p can gain by selecting k' bundles, $k' \le k$, containing only p-voters. By Lemma 6.26, all p-voters are consecutive along the order σ. By Lemma 6.27 (1), the voters of each bundle are also consecutive along σ. Thus, our problem is equivalent to Maximum Interval Cover with total interval being n; this problem is solvable in $O(n^2)$ time using dynamic programming, as described by Golab et al. [Gol+09, Section 3.2].

For the Condorcet rule, we use a slightly different algorithm. From now on, we represent each voter by his bundle. When we speak of a solution, we mean a set of bundles of size at most k. The goal is to find a set Δ of bundles of minimum size whose addition to the original profile makes p a Condorcet winner. First, let $\beta\colon x_1 \succ x_2 \succ \ldots \succ x_{n+n'}$ be a single-crossing order of all voters in $V \cup W$, and let β_{res} be the order derived from β by considering only the voters in $V \cup \bigcup_{\delta \in \Delta} \delta$.

From Lemma 6.28 we know that the voters in $\text{Med}(\beta_{\text{res}})$ must be p-voters. Thus, we begin by guessing at most two bundles δ_1 and δ_2 that may contain p-voters from $\text{Med}(\beta_{\text{res}})$ as median voters (for simplicity, we define the bundle of each registered voter to be his singleton). Since by Lemma 6.27 (1), the voters of every bundle are consecutive along the single-crossing order, we can update all bundles by excluding those voters from $\delta_1 \cup \delta_2$ afterwards. Let $V' := V \cup \delta_1 \cup \delta_2$. Now observe that to make p win, by Lemma 6.28, we only have to find appropriate bundles ordered in front of and behind $\delta_1 \cup \delta_2$ such that the median voters (with respect to the restricted single-crossing order) are p-voters. To formalize this idea, we guess two additional sizes z_1 and z_2 which will denote the numbers of unregistered voters from $\kappa(W')$ ordered in front of and behind $\delta_1 \cup \delta_2$. Note that with these two values z_1 and z_2, we can already check whether p has any chance to become a Condorcet winner because we know the order of the voters in β.

It remains to find the minimum number of unregistered bundles such that

- their union consists of exactly z_1 unregistered voters ordered in front of $\delta_1 \cup \delta_2$ and exactly z_2 unregistered voters ordered behind $\delta_1 \cup \delta_2$, and
- adding these bundles to V' makes some p-voters median voters.

As for the case of the plurality rule, finding bundles fulfilling the first requirement can be done in $O(n^2)$ time, using the algorithm of Golab et al. [Gol+09, Section 3.2]. Checking whether p-voters are median voters for some order can be done in $O(n + n')$ time. Altogether, we can find in $O((n + n') \cdot m^2 + (n + n')^2 \cdot n^4)$ time a set of bundles of minimum size, which make p a Condorcet winner when added. □

6.8 Concluding remarks

We introduced a new notion of combinatorial control in voting. Specifically, we considered the problem of constructive control by adding voters: instead of being able to add voters one-by-one, we can add some bundles of voters at unit cost each. In our study, we have focused on the plurality rule and the Condorcet rule. We saw how the specific structures—be it preference structures or parameters of the input instance—influence the computational complexity of our new model.

Our work provides several opportunities for future research. First, we draw an almost complete picture of the multivariate complexity analysis for three parameters: the maximum bundle size b, the number k of bundles allowed to add, the maximum swap distance d between the preference order of a voter and the preference order of any voter in its bundle (Table 6.1). The case with bundle size at most three and swap distance two remains open.

Second, for the plurality rule, for anonymous bundling functions and for maximum bundle size two, we provide a fixed-parameter algorithm for our C-CC-AV problem parameterized by the solution size k (Theorem 6.17). The running time of the algorithm, however, is *not* upper-bounded single-exponentially in k. It would be interesting to know whether the problem can be solved in a running time which has a single exponential bound. Moreover, it would also be interesting to know whether it admits a polynomial kernel (see Section 2.6.1 for the definition).

Third, we did not discuss destructive control and the related problem of combinatorial deletion of voters. For plurality, we expect that combinatorial addition of voters for destructive control, and combinatorial deletion of voters for either constructive or destructive control behave similarly to combinatorial addition of voters for constructive control. For instance, we can adapt the parameterized reduction of our W[2]-hardness result from Theorem 6.6 to show that these three variants of combinatorial voter control problems are W[2]-hard (and thus NP-hard). We conjecture that Proposition 6.5, Theorem 6.8, and Corollary 6.11 can be transferred to these cases. Understanding these other variants, and also considering other voting rules which are not already hard for the non-combinatorial model, is a promising route for future work.

Another field of future research is to study other combinatorial voting models—this may include controlling the swap distance, "probabilistic bundling" where the bundle of each voter is chosen at random, or using other distance measures besides the swap distance. Naturally, it would also be interesting to consider problems other than voting control as for instance, SHIFT BRIBERY (also see Chapter 7 where we study this problem); a combinatorial variant of SHIFT BRIBERY has very recently been started by Bredereck et al. [Bre+15e].

Finally, instead of studying combinatorial voter control, one might also be interested in combinatorial control by adding or deleting alternatives. As the problem is already NP-hard for the plurality rule and for the non-combinatorial variant, Chen et al. [Che+15a] conducted a parameterized complexity analysis for the parameter "number of the voters" for several popular voting rules.

CHAPTER 7

Shift Bribery

Don't buy a single vote more than necessary.
Joseph P. Kennedy, 1958

In the previous chapter, we have studied how to manipulate voting results by adding voters in a combinatorial setting, where adding a voter also means adding a bundle of other voters at unit price. In this chapter, we continue with the study of voting manipulation, but investigate a different strategy. Here, we are allowed to shift our preferred alternative p higher in some voters' preference orders. Each such shift, however, comes at a price and we must not exceed a given budget. We study the parameterized complexity with respect to several parameters, pertaining to the nature of the solution sought and the size of the input profile, and for several classes of price functions. When we parameterize by the number of affected voters, then for each of our voting rules (Borda, maximin, Copeland) the problem is W[2]-hard. If, instead, we parameterize by the number of positions by which p is shifted in total, then the problem is fixed-parameter tractable for Borda and maximin, and it is W[1]-hard for Copeland. If we parameterize by the budget, then the results depend on the price function class.

7.1 Introduction

In the previous chapter, we have studied how, by organizing a constructive campaign, convincing some unregistered voters to vote may influence the voting outcome. However, we did not consider the fact that such a campaign may also affect the opinion of already registered voters. In this chapter, we take a closer look at this effect. We are particularly interested in the case where possible changes of

This chapter is based on "Prices Matter for the Parameterized Complexity of Shift Bribery" by R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier, *Information and Computation* [Bre+16a] .

a preference order always involve shifting the preferred alternative higher in the specific order. To illustrate this concept, let us first describe three typical scenarios in rank aggregation.

- There are product rankings based on comparing prices, features, and different tests performed by various institutions such as foundations, journals, etc. These rankings are then aggregated into the final one or a winner may be defined according to the rankings.
- Universities are judged based on multiple different criteria, including the number of students per faculty member, availability of particular facilities, the number of Nobel prize winners employed etc.
- Sport competitions involve multiple rankings. For example, a Formula 1 season consists of about twenty races, each resulting in a ranking of the drivers.

A company intending to make its product a winner or a university aiming to increase its prestige or a racing team performing a post-season analysis may want to influence the component rankings obtained from different sources (different product tests, different judgment criteria, different races, different voters). Clearly, this comes at a cost which may differ from source to source and, indeed, can sometimes be quite high. To understand how to achieve a better ranking or even to become a winner with a limited budget, we study the SHIFT BRIBERY problem which, given a preference profile, some preferred alternative p, and some budget, asks whether it is possible to make p win by bribing voters to shift p higher in their preference orders while spending no more than the given budget. SHIFT BRIBERY was introduced by Elkind, Faliszewski, and Slinko [EFS09]; we describe related work in more detail below. Naturally, the effect or the number of positions by which p is shifted in each voter's preference order depends on the voter's character and situation, and on the amount of effort we invest into convincing the voter. This "effort" could, for example, be the amount of time spent, the cost of implementing a particular change, or, in the bribery view of the problem, the payment to the voter. Thus, the problem and, therefore, its computational complexity, depends on the voting rule, on various voting parameters such as the numbers of alternatives and voters, and on the type of price functions describing the efforts needed to shift p up by a given number of positions in the voters' preference orders. Our goal is to unravel the nature of these dependencies.

Related work and background. The computational complexity of bribery in voting was first studied by Faliszewski, Hemaspaandra, and Hemaspaandra [FHH09]. They considered the BRIBERY problem, where one asks if it is possible to make a given alternative win by changing at most a given number of preference orders. Its priced variant, $BRIBERY, is the same except that each voter has a possibly different price for the change of his preference order. These problems were studied for various voting rules, including the Borda rule [Bre+08, FHH09], the maximin rule [FHH11b], and the Copeland$^{\alpha}$ rule [Fal+09b] (see Section 7.4 for definitions of these rules). Recently, Gertler et al. [Ger+15] studied the bribery problem for linear ranking systems. Notably, the destructive variant of BRIBERY, known under the name MARGIN OF VICTORY [MRS11, Xia12], where the goal is to make an alternative lose, has a surprisingly positive motivation—it can be used in efficient post-election audits which aim to detect errors or fraud in election. This is because audits need to be more vigilant in a close election, where making an initial winner lose requires only to change a few voters' preference orders, than in an election with landslide victory. The *margin of victory* measures exactly how close a winner is to losing.

The above problems, however, do not take into account that the price of bribing a voter may depend on what preference order we wish the "bribed" voter to cast. For example, a voter might be perfectly happy to swap the two least preferred alternatives but not the two most preferred ones. To model such situations, Elkind, Faliszewski, and Slinko [EFS09] introduced the SWAP BRIBERY problem. They assumed that each voter has a swap-bribery price function which gives the cost of swapping each two alternatives (provided they are consecutive in the voter's preference order; one can perform a series of swaps to transform the voter's preference order in an arbitrary way). They found that SWAP BRIBERY is both NP-hard and hard to approximate for most well-known voting rules. The reason for these results is, essentially, that the POSSIBLE WINNER problem [BD10, BR12, KL05, XC11] (see Section 8.7 for more information of the definition), which is NP-hard for almost all natural voting rules, is a special case of SWAP BRIBERY with each swap costing either zero or infinity. Motivated by this, Dorn and Schlotter [DS12] considered the parameterized complexity of SWAP BRIBERY for the k-Approval voting rule (where each voter gives a point to his or her top k alternatives). In addition, Elkind, Faliszewski, and Slinko [EFS09] also considered SHIFT BRIBERY, a variant of SWAP BRIBERY where all the swaps have to involve the preferred alternative p. They show that SHIFT BRIBERY remains NP-hard for the Borda, maximin, and Copeland rules but that there is a 2-approximation algorithm for Borda and a polynomial-time algorithm for the k-Approval voting rule. Recently, SHIFT BRIBERY has become more popular as a research subject:

- Elkind and Faliszewski [EF10] gave a 2-approximation algorithm for all scoring rules (generalizing the result for Borda) and other approximation algorithms for Copeland and maximin.

- Schlotter, Faliszewski, and Elkind [SFE15] showed that Shift Bribery is polynomial-time solvable for the Bucklin and the Fallback voting rules.

- Bredereck et al. [Bre+15d] considered the combinatorial variant of Shift Bribery. They showed that, in general, the combinatorial variant of the problem is highly intractable (NP-hard, hard in the parameterized sense, and hard to approximate), but they provided some (approximation) algorithms for natural restricted cases.

- Bredereck et al. [Bre+16c, Bre+16d] solved two open problem we proposed in the conference version [Bre+14b] of this chapter. Using mixed integer linear programming, Bredereck et al. [Bre+16d] showed that for the Borda, maximin, and Copeland$^\alpha$ rules, Shift Bribery parameterized by the number of alternatives is fixed-parameter tractable. Bredereck et al. [Bre+16c] showed that for the Borda and maximin rules, Shift Bribery parameterized by the number of voters is W[1]-hard.

Based on the idea of modeling campaign management as bribery problems, other researchers introduced additional types of bribery problems. For example, Schlotter, Faliszewski, and Elkind [SFE15] introduced Support Bribery and Baumeister et al. [Bau+12] introduced Extension Bribery. Both problems model the setting where voters cast partial preference orders that rank some of their top alternatives only, and the briber completes these preference orders; they differ in that Support Bribery assumes that the voters know their complete preference orders but do not report them completely and Extension Bribery assumes that the voters have no preferences regarding the unreported alternatives.

7.2 Results

For the Borda, maximin, and Copeland$^\alpha$ rules, Shift Bribery has high worst-case complexity as to exact algorithms, but admits polynomial-time approximation algorithms [EF10, EFS09]. To better understand causes of intractability of Shift Bribery in different special cases, we use parameterized complexity analysis. For instance, almost tied voting situations are tempting targets for Shift Bribery. An exact algorithm which is efficient for this special case may be more attractive than a general approximation algorithm. In close-to-tied voting situations it might suffice, for example, to contact only a few voters or, perhaps, to shift the preferred alternative by only a few positions in total. Similarly, it is important to solve the problem exactly if we only have a small budget at disposal. We cover this by using various problem parameterizations and performing a parameterized complexity analysis.

It is natural to expect that the complexity of SHIFT BRIBERY depends on the nature of the voters' price functions and, indeed, there is some evidence for this: For example, if we assume that the price functions are *convex*, then we can verify that the 2-approximation algorithm of Elkind and Faliszewski [EF10] boils down to a greedy procedure that picks the cheapest available single-position shifts until it ensures the designated alternative's victory. Such an implementation would be much faster than the expensive dynamic programming algorithm that they use, but would guarantee a 2-approximate solution for convex prices only. On the contrary, the hardness proofs of Elkind, Faliszewski, and Slinko [EFS09] all use a very specific form of price functions which we call *all-or-nothing* prices, where the cost of bribing a voter is independent of the number of shifts that p will be put forward in this voter's preference order. See Section 7.4.3 for the definitions of the different price functions that we study.

We combine these two observations and study the parameterized complexity of SHIFT BRIBERY for Borda, maximin, and Copeland$^{\alpha}$, for parameters describing the number of affected voters, the number of unit shifts, the budget, the number of alternatives, and the number of voters, under price functions that are either all-or-nothing, sortable, arbitrary, convex, or have a unit price for each single shift. The three voting rules we select are popular in different kinds of voting apart from political elections. For instance, Borda is used by the X.Org Foundation to elect its board of directors. A modified variant is used for the Formula 1 World Championship (and numerous other competitions including ski-jumping and song contests). A slightly modified version of Copeland is used to elect the Board of Trustees for the Wikimedia Foundation.

We summarize our results in Table 7.1; for the sake of completeness we also include some results from the literature [Bre+15e, DS12]. In short, it turns out that indeed both the particular parameter(s) used and the nature of the price functions have a strong impact on the computational complexity of SHIFT BRIBERY. We also present novel FPT *approximation schemes* (see Definition 7.3) exploiting the parameters "number of voters" and "number of alternatives" (such schemes are rare in the literature and may be of significant practical interest) and a *partial kernelization* [Bet+11] (polynomial-time data reduction) result for the parameter "number of unit shifts".

parameter $\mathscr{R}$		SHIFT BRIBERY unit prices	convex prices	arbitrary prices	sortable prices	all-or-nothing prices
budget (B)						
B/M	[Cor 7.13 & 7.14]	FPT	FPT	W[2]-h	W[2]-h	W[2]-h
C	[Cor 7.13 & 7.14]	W[1]-h	W[1]-h	W[2]-h	W[2]-h	W[2]-h
#shifts (t)						
B/M	[Thm 7.3 & 7.6]	FPT	FPT	FPT	FPT	FPT
C	[Cor 7.10]	W[1]-h	W[1]-h	W[1]-h	W[1]-h	W[1]-h
#affected voters (n_a)						
B/M/C	[Thm 7.11 & 7.12]	W[2]-h	W[2]-h	W[2]-h	W[2]-h	W[2]-h
#alternatives (m)						
B/M/C	[Thm 7.17]	FPT$^{\spadesuit}$	XP	XP	FPT$^{\diamondsuit}$	FPT$^{\diamondsuit}$
#voters (n)						
B/M	[Pro 7.19]	W[1]-h$^{\bigstar}$	W[1]-h$^{\bigstar}$	W[1]-h$^{\bigstar}$	W[1]-h$^{\bigstar}$	FPT
C	[Thm 7.20 & Pro 7.19]	W[1]-h	W[1]-h	W[1]-h	W[1]-h	FPT
		SHIFT BRIBERY(O)				
#alternatives (m)						
B/M/C	[Thm 7.15]	FPT-AS for sortable prices				
#voters (n)						
B/M/C	[Thm 7.18]	FPT-AS for all considered price function families				

Table 7.1.: Parameterized complexity of SHIFT BRIBERY and SHIFT BRIBERY(O) for Borda (B), maximin (M), or Copeland$^{\alpha}$ (C) (for each rational number α, $0 \le \alpha \le 1$). Note that SHIFT BRIBERY(O) is the optimization variant of SHIFT BRIBERY, which seeks to minimize the budget spent. “W[1]-h” (resp. “W[2]-h”) stands for W[1]-hard (resp. W[2]-hard). Definitions for FPT, W[1], W[2], XP are provided in Section 2.6. The FPT result marked with “$\spadesuit$” follows from the work of Dorn and Schlotter [DS12]. The FPT results marked with “$\diamondsuit$” follow from the work of Bredereck et al. [Bre+15e]. The W[1]-hardness results marked with “$\bigstar$” follow from the work of Bredereck et al. [Bre+16c]

7.3 Chapter outline

In Section 7.4 we present specific additional notions and we formally define the SHIFT BRIBERY problem, together with its parameterizations and definitions of price functions. Our results are in Section 7.5 (parameterization by solution cost) and Section 7.6 (parameterization by preference profile measures). We discuss future research perspectives in Section 7.7. For the sake of readability, we refer the reader to our journal version [Bre+16a] for some of the proofs which we have omitted because this chapter already includes a proof using a similar idea. We clearly point out where we do this.

7.4 Definitions, notations and examples

Before we provide additional notions we need in this chapter, we remark that to describe the running times of our FPT algorithms, we often use the $O^*(\cdot)$ notation. It is a variant of the standard $O(\cdot)$ notation where polynomial factors are omitted. For example, if the algorithm's running time is $f(\ell)\cdot|I|^{O(1)}$, where f is superpolynomial, then we would say that it is $O^*(f(\ell))$. We use this notation to emphasize the superpolynomial part of the running time.

7.4.1 Borda, maximin, and Copeland rules

Given a preference profile $\mathcal{P}$, for each two alternatives c and d, we define $N_{\mathcal{P}}(c,d)$ to be the number of voters in $\mathcal{V}(\mathcal{P})$ who prefer c to d.

We consider the Borda rule, the maximin rule, and the Copeland$^\alpha$ family of rules. These rules assign points to every alternative and pick as winners those who get most; we write $\mathsf{score}_{\mathcal{P}}(c)$ to denote the number of points alternative $c \in A$ receives in profile $\mathcal{P}$—the particular voting rule used to compute the score will always be clear from the context.

Definition 7.1 (Borda and maximin rules)**.** Let $\mathcal{P}$ be a profile with complete preference orders. Under *Borda*, each alternative $c \in A$ receives from each voter $v \in \mathcal{V}(\mathcal{P})$ as many points as there are alternatives that v ranks lower than c. Formally, the Borda score of alternative c is $\mathsf{score}_{\mathcal{P}}(c) := \sum_{d\in C\setminus\{c\}} N_{\mathcal{P}}(c,d)$. Similarly, the *maximin* score of a alternative c is the number of voters who prefer c to his "strongest competitor". Formally, for the maximin rule, we have $\mathsf{score}_{\mathcal{P}}(c) := \min_{d\in C\setminus\{c\}} N_{\mathcal{P}}(c,d)$.

Definition 7.2 (Copeland$^\alpha$)**.** Let $\mathcal{P}$ be a profile with complete preference orders. Given a rational number $\alpha \in [0,1]$, For each two alternatives $a,b \in A$ with $a \neq b$, recall that a beats b if a majority of voters prefers a to b. We say that a ties with b if exactly half of the voters prefer a to b. The *Copeland* score of a given alternative c equals the number of alternatives that c beats plus α times the number of alternatives that c ties with. Formally,

$$\mathsf{score}_{\mathcal{P}}(c) := \left|\{d \in A\setminus\{c\}\colon N_{\mathcal{P}}(c,d) > N_{\mathcal{P}}(d,c)\}\right| + \alpha\cdot\left|\{d \in A\setminus\{c\}\colon N_{\mathcal{P}}(c,d) = N_{\mathcal{P}}(d,c)\}\right|.$$

Typical values of α are 0, 1/2, and 1, but there are cases where other values are used [Fal+09b]. All our results on SHIFT BRIBERY for Copeland$^\alpha$ hold for each rational value of α. For the sake of brevity, we write "Copeland" instead of "Copeland$^\alpha$ for arbitrary rational number α" throughout this chapter.

7.4.2 FPT approximation scheme

Typically, approximation algorithms are asked to run in polynomial time. It is, however, not always possible to approximate the optimal solution to an arbitrary given factor in polynomial time. Relaxing the polynomial running time to FPT time, we can obtain the following notion (also see Marx [Mar08] for more information).

Definition 7.3 (FPT-approximation scheme (FPT-AS))**.** An FPT approximation scheme (FPT-AS) with parameter k for a minimization problem is an algorithm that, given an instance I and a value $\varepsilon > 0$, returns a $(1+\epsilon)$-approximate solution. This algorithm has running time $f(k,\varepsilon)\cdot|I|^{O(1)}$, where f is a computable function depending on k and ε.

When describing the running time of an approximation scheme, we treat ε as a fixed constant. Thus, a polynomial-time approximation scheme may, for example, include exponential dependence on $1/\varepsilon$.

7.4.3 Shift Bribery

Given a voting rule $\mathscr{R}$, in $\mathscr{R}$ SHIFT BRIBERY the goal is to ensure our preferred alternative p's victory. To this end, we can shift p forward in some of the voters' preference orders. Each shift may have a different price, depending on the voter and on the length of the shift. Here we follow the notation of Elkind and Faliszewski [EF10].

Definition 7.4 (Price functions)**.** A price function $\pi\colon \mathbb{N}\to\mathbb{N}$ for a given voter v gives, for each number ℓ of positions, the price $\pi(\ell)$ of shifting p forward in v's preference order by ℓ positions. We require that $\pi(0)=0$ and that $\pi(\ell)\le\pi(\ell+1)$ for each $\ell\in\mathbb{N}$. We also assume that if p is ranked at a position r in the preference order, then $\pi(\ell)=\pi(\ell-1)$ whenever $\ell\ge r$. In other words, it costs nothing to keep a voter's preference order as it is, it never costs less to shift p farther, and we cannot shift p beyond the top position in the preference order.

Given a voter set V, we write $\boldsymbol{\pi}$ (bold face) to denote a mapping of SHIFT BRIBERY price functions that describes a price function $\boldsymbol{\pi}(v)$ for each voter v in V.

Example 7.1. Let v be a voter with preference order $c_1 \succ c_2 \succ p \succ c_3$ and let π be v's price function. Then, by paying $\pi(1)$ we can change v's preference order to $c_1 \succ p \succ c_2 \succ c_3$, and by paying $\pi(2)$ we can change it to $p \succ c_1 \succ c_2 \succ c_3$.

It is clear that we need at most $|C|-1$ values to completely describe each SHIFT BRIBERY price function.

Definition 7.5 (Shift actions)**.** Given a profile $\mathscr{P}$ with a voter set V. A *shift action* SA is a function of type $V \to \mathbb{N}$, that describes how far p is shifted in the preference order of each voter $v \in V$. We define shift($\mathscr{P}$, SA) to be the profile $\mathscr{P}'$ identical to $\mathscr{P}$ except that p has been shifted forward in the preference order of each voter v by SA(v) positions. If that would mean moving p beyond the top position in some preference order, we shift p up to the top position only.

Given a mapping $\boldsymbol{\pi}$ of price functions for the voter set V, we write

$$\mathsf{price}(\boldsymbol{\pi}, \mathrm{SA}) := \sum_{v \in V} \boldsymbol{\pi}(v)(\mathrm{SA}(v))$$

to denote the price of a given shift action SA. A shift action SA is *successful* if p is a winner in the profile shift($\mathscr{P}$, SA). The term *unit shift* refers to shifting p by one position in one preference order.

Using this notation, the $\mathscr{R}$ SHIFT BRIBERY decision problem is defined as follows.

> $\mathscr{R}$ SHIFT BRIBERY
> **Input:** A profile $\mathscr{P}$ with voter set V, a mapping $\boldsymbol{\pi}$ of SHIFT BRIBERY price functions for V, a preferred alternative $p \in \mathscr{A}(\mathscr{P})$, and a budget bound $B \in \mathbb{N}$.
> **Question:** Is there a shift action SA with cost $\mathsf{price}(\boldsymbol{\pi}, \mathrm{SA}) \leq B$ and p is an $\mathscr{R}$-winner in shift($\mathscr{P}$, SA)?

The optimization variant SHIFT BRIBERY(O) is defined likewise, but does not include the budget B in the input. Instead, it asks for a shift action SA that ensures p's victory at minimum cost $\mathsf{price}(\boldsymbol{\pi}, \mathrm{SA})$.

> $\mathscr{R}$ SHIFT BRIBERY (O)
> **Input:** A profile $\mathscr{P}$ with voter set V, a mapping $\boldsymbol{\pi}$ of SHIFT BRIBERY price functions for V, and a preferred alternative $p \in \mathscr{A}(\mathscr{P})$.
> **Task:** Find a shift action SA such that $\mathsf{price}(\boldsymbol{\pi}, \mathrm{SA})$ is minimum and p is an $\mathscr{R}$-winner in shift($\mathscr{P}$, SA).

For an input instance I of SHIFT BRIBERY(O), we write $\mathsf{OPT}(I)$ to denote the cost of a cheapest successful shift action for I (and we omit I if it is clear from the context).

7.4.4 Price Functions

The price functions used in SHIFT BRIBERY instances may strongly affect the computational complexity of the problem. In this section we present the families of price functions we focus on.

All-or-nothing prices. A price function π is *all-or-nothing* if there is a value $c > 0$ such that $\pi(0) = 0$ and for each $\ell > 0$, $\pi(\ell) = c$ (this value c can be different for each voter). The NP-hardness proofs of Elkind, Faliszewski, and Slinko [EFS09] use exactly this family of price functions (without referring to them directly, though).

Convex prices. A price function π is *convex* if for each $\ell > 0$, $\pi(\ell+1) - \pi(\ell) \leq \pi(\ell+2) - \pi(\ell+1)$ (provided that it is possible to shift the preferred alternative by up to $\ell+2$ positions in the given preference order).

Unit prices. To model the setting where each unit shift has the same cost for each voter, we define the *unit price function* by setting $\pi(\ell) = \ell$ for each $\ell > 0$ such that p can be shifted by ℓ positions. Unit prices are an extreme example of convex price functions.

Sortable prices. Sortable price functions describe a property of voters with the same preference order. A mapping $\boldsymbol{\pi}$ of price functions for the voter set V is called *sortable* if for each two voters $v, v' \in V$ with the same preference order, it holds that

$$\forall 1 \leq \ell \leq m-2\colon\ \boldsymbol{\pi}(v)(\ell) > \boldsymbol{\pi}(v')(\ell) \implies \boldsymbol{\pi}(v)(\ell+1) > \boldsymbol{\pi}(v')(\ell+1).$$

Informally, this means that one can sort each set V' of voters with the same preferences such that between two voters, bribing the one who comes first in this ordering is always cheaper, no matter how many positions p should be shifted. Many natural price function families are sortable. For example, the exponential functions of the form $\boldsymbol{\pi}(v)(\ell) = b_v^{\ell}$ (where each voter v may have an individual base b_v) are sortable. The polynomials of the form $\boldsymbol{\pi}(v)(\ell) = b_v \cdot \ell^c$ (where the exponent c is the same for all voters having the same preference order, but each voter v may have an individual coefficient b_v) are sortable as well. We note that, for values ℓ greater or equal to the original position r of p in the voter's preference order, we simply set $\boldsymbol{\pi}(v)(\ell) = \boldsymbol{\pi}(v)(r-1)$.

Example 7.2. Table 7.2 shows seven voters with preference orders and corresponding price functions. For each price function π_i with $1 \leq i \leq 7$, the corresponding row in Table 7.2(a) lists the price of shifting p higher by j positions. For instance, the intersection of row π_6 and column 3 in Table 7.2(b) contains an entry with value 8; this means that shifting p higher by 3 positions in voter v_8's preference order costs 3 units. We can verify that π_1 is an all-or-nothing and convex price function, π_3 is a unit and convex price function, and π_5, π_6, π_7 are convex functions. The mapping of all voters except v_7 to their respective price functions is sortable.

voter v_1: $e \succ c \succ a \succ b \succ p \succ d$	
voter v_2: $c \succ d \succ a \succ p \succ e \succ b$	
voter v_3: $b \succ a \succ d \succ p \succ e \succ c$	
voter v_4: $e \succ d \succ b \succ a \succ c \succ p$	
voter v_5: $b \succ a \succ e \succ p \succ c \succ d$	
voter v_6: $b \succ a \succ e \succ p \succ c \succ d$	
voter v_7: $b \succ a \succ e \succ p \succ c \succ d$	

(a)

	1	2	3	4	5
π_1:	1	1	1	1	1
π_2:	2	3	3	3	3
π_3:	1	2	3	3	3
π_4:	2	2	2	5	5
π_5:	3	3	6	6	6
π_6:	3	4	7	7	7
π_7:	2	4	8	8	8

(b)

Table 7.2.: An illustration of the concept of price functions. (a) The preference orders of seven voters. (b) Each row is labeled with the corresponding price function π. We omit the the case of shifting p by zero positions. If a column is labeled x, then its entries show the prices of shifting p forward by x positions (if possible). We draw a boundary for each row to show the upper bound of the number of shifts p can move forward in the respective voter's preference order.

Given a preference profile $\mathcal{P}$, we write $\Pi_{\mathsf{arbitrary}}$ to denote the set of all price function mappings for voter set $\mathcal{V}(\mathcal{P})$, Π_{convex} to denote the set of the convex price function mappings for $\mathcal{V}(\mathcal{P})$, Π_{unit} to denote the set of the unit price function mappings for $\mathcal{V}(\mathcal{P})$, $\Pi_{\mathrm{all-or-nothing}}$ to denote the set of the all-or-nothing price function mappings for $\mathcal{V}(\mathcal{P})$, and Π_{sort} to denote the set of all sortable mappings for $\mathcal{V}(\mathcal{P})$. We observe the following straightforward relations between these sets (also see the Hasse diagram in Figure 7.1).

Proposition 7.1. *For each given profile, the following relations hold between the families of price functions.*

1. $\Pi_{\mathsf{unit}} \subset \Pi_{\mathsf{convex}} \subset \Pi_{\mathsf{arbitrary}}$,
2. $\Pi_{\mathrm{all-or-nothing}} \subset \Pi_{\mathsf{sort}} \subset \Pi_{\mathsf{arbitrary}}$,
3. $\Pi_{\mathsf{unit}} \subset \Pi_{\mathsf{sort}}$.

7.4.5 Parameters for Shift Bribery

So far, SHIFT BRIBERY has not been studied from the parameterized complexity point of view. Dorn and Schlotter [DS12] and Schlotter, Faliszewski, and Elkind

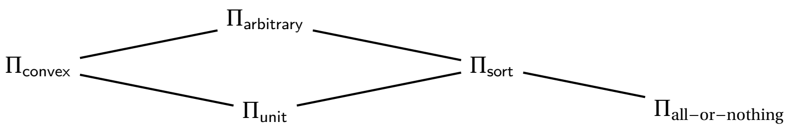


Figure 7.1.: Hasse diagram of the inclusion relationship among the price function families (for a given preference profile).

[SFE15] have, however, provided parameterized complexity results for SWAP BRIBERY and for SUPPORT BRIBERY. We consider two families of parameters. One is related to the properties of a successful shift action that we seek:

#shifts t	the total number of unit shifts in a shift action,
#voters-affected n_a	the total number of voters whose preference orders are changed,
the budget B	the sum of prices of bribing the voters.

The other parameters describe the preference profile:

#alternatives m	the number of alternatives,
#voters n	the number of voters.

We assume that the values of these parameters are passed explicitly as part of the input. Note that some parameters such as m and n clearly are incomparable whereas n (resp. t) clearly upper-bounds n_a, making the last parameter "stronger" [KN12] than the first two parameters. Similarly, B upper-bounds both t and n_a, provided that each shift has a price of at least one.

Observation 7.2. *Let m denote the number of alternatives, n the number of voters, n_a the number of affected voters, t the total number of unit shifts in a successful shift action, and B the sum of prices of the bribed voters. The following holds.*

1. $n_a \le \min(n, t)$.
2. $t \le n_a \cdot m$.
3. *For unit prices,* $t \le B$.
4. *For all-or-nothing prices,* $n_a \le B$.

7.5 Parameterizations by solution cost measures

In this section we present our results for parameters related to the cost of a successful shift action, that is, the number of unit shifts, the number of voters affected by at least one shift, and the budget. It turns out that parameterization by the number of unit shifts tends to yield presumably lower complexity (FPT algorithms for Borda and maximin and W[1]-hardness for Copeland) than parameterization by the number of affected voters (W[2]-hardness). Parameterization by the budget lies in between, and the complexity depends on each particular price function family as already shown in Observation 7.2 .

7.5.1 Unit shifts

For Borda and maximin, SHIFT BRIBERY parameterized by the number t of unit shifts in the solution is in FPT for arbitrary price functions. This is due the nature of these voting rules, which allows us to focus on a small number of alternatives. More precisely, we can shrink the number of alternatives as well as the number of voters so that they are upper-bounded by functions in t (in effect, achieving a partial kernelization [Bet+11]).

Theorem 7.3. BORDA SHIFT BRIBERY *parameterized by the number t of unit shifts is in* FPT *for arbitrary price functions. The running time of the algorithm is* $O^*((2^t \cdot (t+1) \cdot t)^t)$.

Proof. Let $\mathcal{P}$ be a preference profile with the alternative set A and the voter set V, and let $\boldsymbol{\pi}$ be a mapping of price functions for the voter set $\mathcal{V}(\mathcal{P})$. Suppose that there is a successful shift action SA for our preferred alternative $p \in A$ and it uses at most t unit shifts. Our algorithm iterates over all non-negative integers t' with $t' \leq t$, which we interpret as the exact number of shifts used in the successful shift action.

Under the Borda rule, applying a shift action that uses t' unit shifts increases the score of p to exactly $\mathsf{score}_{\mathcal{P}}(p) + t'$. This means that irrespective of which shift action with t' unit shifts we use, if some alternative c has initial score at most $\mathsf{score}_{\mathcal{P}}(p) + t'$, then after applying the shift action, p will certainly have score at least as high as c. On the contrary, if some alternative c has score greater than $\mathsf{score}_{\mathcal{P}}(p) + t'$, then a successful shift action must ensure that c loses at least $\mathsf{score}_{\mathcal{P}}(c) - (\mathsf{score}_{\mathcal{P}}(p) + t')$ points. Since every unit shift decreases the score of exactly one alternative, it follows that if there is a successful shift action that uses exactly t' shifts, then the number of alternatives with score greater than $\mathsf{score}_{\mathcal{P}}(p) + t'$ must be at most t'. We define the set containing such alternatives as

$$A(t') := \{c \in \mathcal{A}(\mathcal{P}) \mid \mathsf{score}_{\mathcal{P}}(c) > \mathsf{score}_{\mathcal{P}}(p) + t'\}. \qquad (\star)$$

Our algorithm first computes the set $A(t')$ according to ($\star$). If $|A(t')| > t'$, then the algorithm skips the current iteration and continues with $t' \leftarrow t' + 1$.

By the above argument, we can focus on a small group of alternatives, namely on $A(t')$. It turns out that we can also focus on a small group of voters. We first define these voters and then show why it is enough to focus on them. For each subset $A' \subseteq A(t')$ and for each non-negative integer j, $0 \le j \le t'$, we compute a subset $V' \subseteq V$ of at most t' voters such that the following holds.

(i) For each voter $v \in V'$, if we shift p by j positions in v's preference order, then p passes all alternatives from A' and $j - |A'|$ other alternatives from $A \setminus A(t')$.

(ii) $\sum_{v \in V'} \boldsymbol{\pi}(v)(j)$ is minimal.

We denote this set as $V(A', j)$. If there are several subsets of voters that satisfy these conditions, then we pick one arbitrarily. Finally, we set

$$V(t') := \bigcup_{\substack{A' \subseteq A(t') \\ 0 \le j \le t'}} V(A', j).$$

Claim 7.4. *If there is a successful shift action* SA *using exactly t' unit shifts and of cost at most B, then there is a successful shift action* SA$'$ *using exactly t' shifts and of cost at most B such that* SA$'$ *affects only the voters in $V(t')$.*

Proof. To see this, assume that there is a voter $v \not\in V(t')$ and an integer j such that SA shifts p by j positions in v's preference order, that is, SA$(v) = j$. Now consider the set A' of alternatives from $\{c \in \mathcal{A}(\mathcal{P}) \mid \mathsf{score}_{\mathcal{P}}(c) > \mathsf{score}_{\mathcal{P}}(p) + t'\}$ that p has passed after the shifting. By the definition of $V(A', j)$ and by simple counting arguments, there is a voter $v' \in V(A', j)$ for which SA does not shift p. Again, by definition of $V(A', j)$, we know that $\boldsymbol{\pi}(v')(j) \le \boldsymbol{\pi}(v)(j)$. Thus, if in the shift action SA we replace the shift by j positions in the preference order of v by a shift of j positions in the preference order of v', then we obtain a successful shift action using exactly t' shifts and of cost no greater than that of SA. We can repeat this process until we obtain a successful shift action using exactly t' shifts and of cost at most B that affects the voters in $V(t')$ only. (of Claim 7.4) $\diamond$

By Claim 7.4, to find a successful shift action using exactly t' unit shifts, it suffices to focus on the voters from $V(t')$. The size of $V(t')$ can be upper-bounded by

$$|V(t')| \le \sum_{\substack{A' \subseteq A(t') \\ 0 \le j \le t'}} t' \le 2^{t'} \cdot (t' + 1) \cdot t'.$$

Since in the preference order of each voter in $V(t')$ we can shift p by at most t' positions and we can do so for at most t' voters, there are at most $|V(t')|^{t'}$ shift actions that we need to try. We can consider them all in FPT time: $O^*((2^{t'} \cdot (t'+1) \cdot t')^{t'})$. $\square$

Example 7.3. We use the profile and price functions from Example 7.2 to illustrate the idea of Claim 7.4. Suppose that we are in the iteration with $t' = 3$ and we have $A(3) = \{a, b\}$. According to the the preference orders and price functions given in Table 7.2, we obtain the following.

- If $A' = \emptyset$, then $V(A', 1) = \{v_3, v_4, v_7\}$ and $V(A', 2) = V(A', 3) = \emptyset$.
- If $A' = \{a\}$, then $V(A', 1) = \{v_2\}$, $V(A', 2) = \{v_2, v_3, v_4\}$, and $V(A', 3) = \{v_2\}$.
- If $A' = \{b\}$, then $V(A', 1) = \{v_1\}$, $V(A', 2) = V(A', 3) = \emptyset$.
- If $A' = \{a, b\}$, then $V(A', 1) = \emptyset$, $V(A', 2) = \{v_1\}$, and $V(A', 3) = \{v_1, v_3, v_4\}$.

This implies that $V(3) = \{v_1, v_2, v_3, v_4, v_7\}$.

Note that, for the case of $A' = \{a\}$ and $j = 2$, both $\{v_2, v_3, v_4\}$ and $\{v_3, v_4, v_5\}$ fulfill Conditions (i)–(ii). We arbitrarily choose $V(A', j) = \{v_2, v_3, v_4\}$. Now, suppose that a successful shift action $\textsc{Sa}\colon V \to \mathbb{N}$ using 3 shifts has $\textsc{Sa}(v_1) = 2$, $\textsc{Sa}(v_5) = 3$. This action costs $\mathsf{price}(\boldsymbol{\pi}, \textsc{Sa}) = 5$. Applying it makes alternatives a, b, and e each lose one point. Since $\textsc{Sa}(v_5) = 2$, we shift p forward by two positions in voter v_5's preference order, passing exactly one alternative, e, not from A', and one alternative, a, from A'. By our constructions, we have that $v_5 \notin V(\{a\}, 2) = \{v_2, v_3, v_4\}$ and $v_5 \notin V(3)$. However, since $|V(\{a\}, 2)| = 3$ and $\textsc{Sa}$ can affect at most $t' = 3$ voters, we can find one voter in $V(\{a\}, 2) \setminus \{v_1, v_5\}$ to affect and make p win. For instance, if we replace shifting v_5's preference order with shifting v_3's preference order, then we obtain a shift action $\textsc{Sa}'$ with $\textsc{Sa}(v_1) = 2$ and $\textsc{Sa}'(v_3) = 2$, and $\textsc{Sa}'(v_i) = 0$ for other voter v_i. Applying $\textsc{Sa}'$ makes alternatives a, b, d each lose one point (note that d is not in our set $A(3)$). This action uses 3 shifts and even has a lower cost, 4.

Using an idea from the proof of Theorem 7.3, we can transform a given profile into one that consists of only $g_1(t)$ alternatives and $g_2(t)$ voters, where g_1 and g_2 are two computable functions. In this way, we obtain a so-called partial problem kernel [Bet+11, Kom15]. It is a *partial* problem kernel only and not a regular problem kernel, since the cost defined by the price function π and the budget are not necessarily upper-bounded by a function of t (see Section 2.6.1 for the definition and further discussion of problem kernels).

Theorem 7.5. *An instance of* BORDA SHIFT BRIBERY *parameterized by the number t of unit shifts can be reduced to an equivalent instance with the same budget, and with $O(t^4 \cdot 2^t)$ alternatives and $O(t^3 \cdot 2^t)$ voters.*

Proof. Let $\mathscr{P}$ be our initial preference profile with alternative set A and voter set V. By the proof of Theorem 7.3, we know that we only need to focus on a subset A_{crit} of alternatives whose size is upper-bounded by a function of the parameter t. This set A_{crit} is defined as follows:

$$A_{\text{crit}} := \bigcup_{t'=0}^{t} A(t'),$$

where $\{c \in \mathscr{A}(\mathscr{P}) \mid \mathsf{score}_{\mathscr{P}}(c) > \mathsf{score}_{\mathscr{P}}(p) + t'\}$ if the number of alternatives whose scores are greater than $\mathsf{score}_{\mathscr{P}}(p) + t'$ is at most t'; otherwise it will not be possible to decrease the score of every of those alternatives by at least one within t' unit shifts, thus, $A(t') := \emptyset$.

From the same proof, we also know that we only need to focus on a small group of voters from the set $\bigcup_{t'=0}^{t} V(t')$, where $V(t')$ is defined as in the proof of Theorem 7.3. Briefly put, $V(t')$ consists of the voters such that if we shift p in some of their preference orders by a total of t' positions, then we may make p a winner. As already shown in that proof, $V(t')$ has at most $2^{t'} \cdot (t'+1) \cdot t'$ voters. Thus, we can upper-bound the size of V_{crit} by $\sum_{t'=0}^{t} 2^{t'} \cdot (t'+1) \cdot t' \le t^3 \cdot 2^t$. Now, if the total number n of voters in initial voter set V is at most this bound $t^3 \cdot 2^t$, then we simply set $V_{\text{crit}} := V$ to be the set of all voters. Otherwise,

$$V_{\text{crit}} := \bigcup_{t'=0}^{t} V(t').$$

Since we only need to compute $\bigcup_{t'=0}^{t} V(t')$ when $n \ge t^3 \cdot 2^t$, it follows that V_{crit} can be computed in polynomial time using the straightforward algorithm in the previous proof. Clearly, $|A_{\text{crit}}| \le t \cdot (t-1)/2$ and $|V_{\text{crit}}| \le \min\{n, \sum_{t'=0}^{t} 2^{t'} \cdot (t'+1) \cdot t'\} \le \min\{n, t^3 \cdot 2^t\}$.

The remaining task is to construct an equivalent preference profile *containing* all alternatives from A_{crit} and the voters from V_{crit} along with their price functions such that the size of the profile is still upper-bounded by a function in t.

We construct three groups of sets of alternatives for the new profile as follows: First, we introduce a dummy alternative d_i for each alternative $c_i \in A_{\text{crit}}$. We use it to realize the original score difference between p and alternative c_i. We denote the set of all these dummy alternatives as D.

Second, for each voter $v_i \in V_{\text{crit}}$, we need to replace the alternatives in his original preference order that do not belong to A_{crit} but that are still relevant to the shift action. Let A_i be the set of alternatives from A_{crit} that are ranked in front of p by voter v_i, that is,

$$A_i := \{c' \in A_{\text{crit}} \mid c' \succ_i p\}.$$

Last, we introduce a new set F_j of filler alternatives whose size equals the number of irrelevant alternatives that ranked in front of p by no more than t positions. That is,

$$F_i := \{c \in A \setminus A_{\text{crit}} \mid c \text{ is ranked at most } t \text{ positions ahead of } p \text{ by } v_i\}.$$

The new alternative set A_{new} is defined as

$$A_{\text{new}} := A_{\text{crit}} \cup D \cup \bigcup_{v_i \in V_{\text{crit}}} A_i \cup F_i.$$

Clearly, $|A_{\text{new}}| \le 2|A_{\text{crit}}| + t \cdot (|A_{\text{crit}}| + |V_{\text{crit}}|) \le t \cdot (t^2+1) + \min\{t^4 \cdot 2^t, t \cdot n\}$.

We construct two groups of voters for the new voter set V_{new}.

1. We construct the first group of voters to retain the score difference between p and every alternative $c_j \in A_{\text{crit}}$ in the original profile. To this end, recall that $\mathsf{score}_{\mathscr{P}}(c_j)$ denotes this score difference. We introduce $\mathsf{score}_{\mathscr{P}}(c_j)$ pairs of voters with the following preference orders.

   $\mathsf{score}_{\mathscr{P}}(c_j)$ voters: $c_j \succ d_j \succ \langle A_{\text{new}} \setminus \{p, c_j, d_j\}\rangle \succ p$,
   $\mathsf{score}_{\mathscr{P}}(c_j)$ voters: $p \succ \langle A_{\text{new}} \setminus \{p, c_j, d_j\}\rangle \succ c_j \succ d_j$.

   In this way, we make d_j have $2 \cdot \mathsf{score}_{\mathscr{P}}(c_j)$ points more than d and $\mathsf{score}_{\mathscr{P}}(c_j)$ points more than all other alternatives, including p. Note that this preference order construction is a very common technique to achieve a certain Borda score difference between two alternatives. We set the cost for the first unit to be $B+1$; this is higher than our budget.

2. We construct the second group of voters to retain the important part of the voters from V_{crit}. That is, for each voter $v_i \in V_{\text{crit}}$, we set his new preference order as follows.

   voter v_i: $\langle A_i \cup F_i\rangle \succ p \succ \langle A_{\text{new}} \setminus (A_i \cup F_i \cup \{p\})\rangle$,

   where $\langle A_i \cup F_i\rangle$ corresponds to the order of the alternatives in v_i's original preference order and the price function remains unchanged. We also add a dummy voter v'_i with the reverse preference order of v_i and we set the cost for the first unit shift to be $B+1$; this is higher than our budget.

We obtain the following inequality for the size of the newly constructed voter set V_{new}:

$$2|V_{\text{crit}}| + \sum_{c_j \in A_{\text{crit}}} 2 \cdot \mathsf{score}_{\mathscr{P}}(c_j) \le t \cdot (t-1) + t^3 \cdot 2^{t+1}.$$

The equivalence of the two instances can be verified as follows: The newly added dummy and filler alternatives have at most the same score as p. Each alternative $c_j \in A_{\text{crit}}$ has exactly $\text{score}_{\mathscr{P}}(c_j)$ points more than p. Since it will be too expensive to bribe the voters from the first group of voters or the dummy voters v'_i ($B+1$ budget for one unit shift), the only possibility of shifting p within the budget B is to bribe the voters from the original profile who have the same preference orders up to renaming of the alternatives that do not belong to A_{crit}. □

The fixed-parameter tractability result for Maximin Shift Bribery follows by using a similar approach as in the proof of Theorem 7.3. Thus, we omit the corresponding proof and refer to the journal version [Bre+16a].

Theorem 7.6. Maximin Shift Bribery *parameterized by the number t of unit shifts is in* FPT *for arbitrary price functions. The running time of the algorithm is $O^*((2^t \cdot (t+1)\cdot t)^t)$.*

Using an analogous approach as in the proof of Theorem 7.5 to construct an equivalent profile, we obtain the following.

Corollary 7.7. *An instance of* Maximin Shift Bribery *parameterized by the number t of unit shifts can be reduced to an equivalent instance with the same budget, and with $O(t^4 \cdot 2^t)$ alternatives and $O(t^3 \cdot 2^t)$ voters.*

For Copeland, we do not get FPT membership, but we show W[1]-hardness even for all-or-nothing prices and for unit prices, which implies hardness for each of our price function families.

Theorem 7.8. Copeland Shift Bribery *parameterized by the number of unit shifts is* W[1]*-hard, even for all-or-nothing prices.*

Proof. For the sake of completeness, fix some rational number α, $0 \le \alpha \le 1$, for the Copeland "tie breaking"; we will see later that we will not use α since the number of voters in the constructed instance is odd.

We provide a parameterized reduction from the W[1]-hard problem Clique parameterized by the clique size h (see the beginning of the proof of Theorem 6.8 for more the definition of Clique). Let G be our input graph with $V(G)$ being the set of r vertices and $E(G)$ being the set of s edges, and let h be the clique order that we seek (we assume $h > 1$). Without loss of generality, assume that the number r of vertices is odd and that $r > 6$ (both assumptions can be guaranteed by adding several isolated vertices to the graph).

We set the available budget B to be $h \cdot (h-1)/2$, which is the number of edges in a clique with h vertices. We construct the preference profile $\mathcal{P} = (A, V)$ as follows. We first add our preferred alternative p, an initial winner d, and a set D of $2r$ *dummy alternatives* to our alternative set A. The introduction of the dummy alternatives is to ensure that p loses exactly h points to the initial voters and the budget is defined such that p can only gain points by passing vertex alternatives that form a clique. Then, for each vertex $u_i \in V(G)$, we add a *vertex alternative* c_i to A. We denote the set of vertex alternatives as A_{vertex}. Formally, we have

$$A := \{p, d\} \cup \{d_{2i-1}, d_{2i}, c_i \mid u_i \in V(G)\}.$$

We construct two groups of voters together with a mapping $\boldsymbol{\pi}$ to their price functions. The first group consists of voters v_e and v'_e for each edge $e = \{u_i, u_{i'}\} \in E(G)$, $i < i'$, with the following preference orders:

$$\begin{aligned}
&\text{voter } v_e\colon && c_i \succ c_{i'} \succ p \succ \langle A \setminus \{c_i, c_{i'}, p\}\rangle,\\
&\text{voter } v'_e\colon && \overleftarrow{c_i \succ c_{i'} \succ p \succ \langle A \setminus \{c_i, c_{i'}, p\}\rangle}.
\end{aligned}$$

They have all-or-nothing prices. That is, $\boldsymbol{\pi}(v_e)(0) = 0$ and $\boldsymbol{\pi}(v_e)(j) = 1$ for each $j > 0$, and $\boldsymbol{\pi}(v'_e)(j) = B + 1$ for each $j \geq 0$. We set $V_{\text{edges}} := \{v_e \mid e \in E(G)\}$.

The second group V_{struct} consists of polynomially many voters with the following properties, which is indeed possible due to a classic theorem by McGarvey [McG53]: Each vertex alternative c_i beats p (by $2h-3$ voters' preference orders) and beats d (by one voter's preference order). Alternative d beats each dummy alternative (by one voters' preference order). Our preferred alternative p beats $2r - h - 1$ dummy alternatives and loses to each remaining dummy alternative (by one preference order); it also beats d (by one preference order). Each alternative in $A_{\text{vertex}} \cup D$ beats exactly $\lfloor 3r/2 \rfloor$ other alternatives in $V(G) \cup D$ (recall that r is odd).

The formal definition of the preference orders of voters in V_{struct} reads as follows. Let $D' \subseteq D$ be a subset of $h+1$ (arbitrary) alternatives.

(a) There is one voter with preference order $\langle A_{\text{vertex}} \cup D\rangle \succ p \succ d$.

(b) There are $(h-2)$ pairs of voters with preference orders

$$\begin{aligned}
&h-2 \text{ voters:} && \langle A_{\text{vertex}}\rangle \succ p \succ d \succ \langle D\rangle,\\
&h-2 \text{ voters:} && \overleftarrow{\langle D\rangle} \succ d \succ \overleftarrow{\langle A_{\text{vertex}}\rangle} \succ p.
\end{aligned}$$

(c) There are two voters with preference orders

$$d \succ \langle D\rangle \succ p \succ \langle A_{\text{vertex}}\rangle \quad \text{and} \quad \overleftarrow{\langle A_{\text{vertex}}\rangle} \succ p \succ d \succ \overleftarrow{\langle D\rangle}.$$

(d) There are two voters with preference orders

$$p \succ \langle D \setminus D' \rangle \succ \langle D' \rangle \succ \langle A_{\text{vertex}} \rangle \succ d \text{ and } d \succ \overleftarrow{\langle A_{\text{vertex}} \rangle} \succ \overleftarrow{\langle D' \rangle} \succ p \succ \overleftarrow{\langle D \setminus D' \rangle}.$$

(e) For technical reasons, we rename the alternatives in A_{vertex} to $a_1, a_2, \ldots, a_r$ and the alternatives D to be $a_{r+1}, a_{r+2} \ldots, a_{2r}$. For the sake of readability, let $\theta := 3r$ and $\eta := \lfloor 3r/2 \rfloor$. For each alternative $a_i \in A_{\text{vertex}} \cup D$, let A_i denote a set of η alternatives,

$$A_i := \{a_{(i+1) \bmod \theta}, a_{(i+2) \bmod \theta}, \ldots, a_{(i+\eta) \bmod \theta}\}, \text{ and}$$

let $\langle A_i \rangle := a_{(i+1) \bmod \theta} \succ a_{(i+2) \bmod \theta} \succ \ldots \succ a_{(i+\eta) \bmod \theta}$. There are two voters with preference orders

one voter: $a_i \succ \langle A_i \rangle \succ \langle (A_{\text{vertex}} \cup D) \setminus A_i \rangle \succ p \succ d$,
one voter: $d \succ p \succ \overleftarrow{\langle (A_{\text{vertex}} \cup D) \setminus A_i \rangle} \succ a_i \succ \overleftarrow{\langle A_i \rangle}$.

Using this type of preference orders, we make each alternative from $A_{\text{vertex}} \cup D$ beat exactly $\lfloor 3r/2 \rfloor$ alternatives from the same set.

Note that each pair of preference orders constructed in (b)–(d) follows a common pattern as originally used by McGarvey [McG53]: The second preference order is almost the reverse of the first one, except that the relative order of two specific alternatives remains the same. This way, we give one alternative two more points than another alternative and one more point than the remaining alternatives.

All voters from V_{struct} have all-or-nothing prices: For each $v \in V_{\text{struct}}$, it holds that $\boldsymbol{\pi}(v)(0) = 0$ and $\boldsymbol{\pi}(v)(j) = B + 1$ for each $j > 0$. We will see that our parameter "number t of unit shifts" is $2B = h \cdot (h-1)$. This completes our construction which runs in polynomial time.

Note that due to the budget, we can only afford to bribe the voters in V_{edges}. Prior to any bribery, p has $2r - h$ points, d has $2r$ points, and each other alternative has at most $\lfloor 3r/2 \rfloor + 2$ points. This means that d is the unique Copeland winner of $\mathcal{P}$. We show that there is a shift action $\textsc{Sa}$ such that $\mathsf{price}(\boldsymbol{\pi}, \textsc{Sa}) \le B$ and p is a winner of $\text{shift}(\mathcal{P}, \textsc{Sa})$ if and only if G has a clique of order h.

For the "only if" part, assume that there is a successful shift action that ensures p's victory. Given our price functions, we can bribe up to $h \cdot (h-1)/2$ voters in V_{edges}. Further, it is clear that a successful shift action must ensure that p obtains h additional points by beating h additional vertex alternatives (it is impossible to decrease the score of d). To beat a vertex alternative, p has to pass it in the preference orders of at least $h-1$ voters. This means that there is some set $A' \subseteq A_{\text{vertex}}$ of at least h vertex alternatives such that p passes each alternative in A' at least $h-1$ times. If

$c \in A'$ and p passes c in $h-1$ preference orders, then the value of $N_{\mathcal{P}}(c,p) - N_{\mathcal{P}}(p,c)$ changes from $2h-3$ to -1, making p beat c. Given our budget, we can altogether shift p by $t = h \cdot (h-1)$ positions (each two positions correspond to an edge in G; note that t is the value of our parameter). Thus A' contains exactly h alternatives, p passes one of them in each unit shift, and p passes each alternative from A' exactly $h-1$ times. This is possible if and only if the alternatives in A' correspond to a clique in G.

For the "if" part, if Q is a set of h vertices from $V(G)$ that form a clique, then shifting p forward in the preference orders of the B voters from V_{edges} that correspond to the edges of the graph induced by Q ensures p's victory. □

The previous reduction can be adapted to the case of unit prices. The key trick is to insert sufficiently many "filler" alternatives directly in front of p in the preference orders of some voters who shall not be affected. This simulates the effect of all-or-nothing prices. Again, we refer the reader to our journal version [Bre+16a] for the proof.

Theorem 7.9. COPELAND SHIFT BRIBERY *parameterized by the number of unit shifts is* W[1]*-hard, even for unit prices.*

By Proposition 7.1, the W[1]-hardness result for the all-or-nothing price function (Theorem 7.8)) also holds for the case when the prices are convex, sortable, or arbitrary. Together with the W[1]-hardness for unit prices (Theorem 7.9) we derive the following.

Corollary 7.10. COPELAND SHIFT BRIBERY *parameterized by the number of unit shifts is* W[1]*-hard for each price function family that we consider.*

7.5.2 Number of affected voters and the budget

For the number of affected voters, SHIFT BRIBERY is W[2]-hard for each of the Borda, maximin, and Copeland$^\alpha$ rules. This holds for each family of price functions we consider: The result for all-or-nothing prices follows almost directly from the NP-hardness proofs due to Elkind, Faliszewski, and Slinko [EFS09]. Their reductions have to be adapted to work for SET COVER (see below for the definition) rather than its restricted variant, EXACT COVER BY 3-SETS [GJ79], but this can be done quite easily. To obtain the result for unit prices, with some effort, it is still possible to carefully modify their proofs, maintaining their main ideas. We present the proof for the Borda rule. The remaining proofs are included in the journal version [Bre+16a].

Theorem 7.11. BORDA SHIFT BRIBERY *parameterized by the number of affected voters is* W[2]*-hard for each of the price function families we consider.*

Proof. We consider BORDA SHIFT BRIBERY for unit prices first. We give a parameterized reduction from SET COVER parameterized by the set cover size h; this problem is defined in the beginning of the proof of Theorem 6.6. Let $(\mathcal{U}, \mathcal{F}, h)$ be our input instance, where $\mathcal{U} = \{u_1, \ldots, u_r\}$ and $\mathcal{F} = (S_1, \ldots, S_s)$ is a collection of subsets of $\mathcal{U}$, and h is a non-negative integer. We construct a set A of alternatives to include our preferred alternative p, an initial alternative d, and an auxiliary alternative g. For each element u_j from the universe $\mathcal{U}$, we add an *element-alternative* a_j to A. We use g to achieve a certain score difference between p and any element alternative a_j. Finally, we add a set D of $2h \cdot (r+1) + 2$ dummy alternatives to A. Thus,

$$A := \{p, d, g\} \cup \{a_j \mid u_j \in \mathcal{U}\} \cup D.$$

We use D to build a block between p and d in some preference orders to prevent p from shifting in those orders.

We construct two groups of voters for our voter set V. To this end, for each set $S_i \in \mathcal{F}$, we denote by $A(i)$ the set of all element alternatives corresponding to the elements in S_i and by $D(i)$ an arbitrarily chosen set of $r - |S_i|$ dummy alternatives from D. Thus, $|A(i)| + |D(i)| = r$.

The first group consists of two voters v_{2i-1} and v_{2i} for each set S_i with preference orders

$$\begin{aligned} v_{2i-1}&\colon\ d \succ \langle A(i) \rangle \succ \langle D(i) \rangle \succ p \succ \langle A \setminus (A(i) \cup D(i) \cup \{p, d\}) \rangle, \\ v_{2i}\ &\colon\ \overleftarrow{d \succ \langle A(i) \rangle \succ \langle D(i) \rangle \succ p \succ \langle A \setminus (A(i) \cup D(i) \cup \{p, d\}) \rangle}. \end{aligned}$$

Note that each pair of voters v_{2i-1}, v_{2i}, gives each alternative $|A| - 1$ points. The second group consists of the following voters:

(a) For each element $u_j \in \mathcal{U}$, we construct $B+1$ pairs of voters with preference orders

$$\begin{aligned} B+1 \text{ voters:}\ & p \succ a_j \succ g \succ \langle A \setminus (\{p, a_j, g, d\}) \rangle \succ d, \\ B+1 \text{ voters:}\ & d \succ \overleftarrow{\langle A \setminus (\{p, a_j, g, d\}) \rangle} \succ a_j \succ g \succ p. \end{aligned}$$

Using these preference orders, we make each element alternative a_j have $2B + 2$ points more than g and $B+1$ points more than the remaining alternatives, including p. Moreover, since the first $B+1$ voters already rank p at the first position and since there are more than B voters between p and d in the second $B+1$ voters' preference orders, we can conclude that it is never beneficial to affect these voters.

(b) We construct $B+1$ pairs of voters with preference orders

$$B+1 \text{ voters: } p \succ \langle D \rangle \succ d \succ g \succ \langle A \setminus (\{p,d,g\} \cup D) \rangle,$$
$$B+1 \text{ voters: } \overleftarrow{\langle A \setminus (\{p,d,g\} \cup D) \rangle} \succ d \succ g \succ \overleftarrow{\langle D \rangle} \succ p.$$

Again, by these preference orders, we make d have $2(B+h)$ points more than g and $B+h$ points more than the remaining alternatives, including p.

Since we are dealing with unit price functions, we set the price function of each voter to be $\pi(\ell) = \ell$. It is easy to verify that there is a value L such that the alternatives in the constructed profile have the following Borda scores:

- $\mathrm{score}_{\mathscr{P}}(d) = L + B + h$.
- $\mathrm{score}_{\mathscr{P}}(p) = L$.
- For each element $u_i \in \mathscr{U}$, $\mathrm{score}_{\mathscr{P}}(a_i) = L + B + 1$.
- For each alternative $c \in D \cup \{g\}$, $\mathrm{score}_{\mathscr{P}}(c) \le L$.

We set our parameter "maximal number n_a of affected voters" to h and set our budget B to $h \cdot (r+1)$. This completes our construction which obviously runs in polynomial time. We show that $(\mathscr{U}, \mathscr{F}, h)$ is a yes-instance of SET COVER if and only if the constructed instance of BORDA SHIFT BRIBERY is a yes-instance.

For the "only if" part, assume that there is a collection $\mathscr{F}'$ of h sets from $\mathscr{F}$ such that their union is $\mathscr{U}$. Without loss generality, let $\mathscr{F}' := \{S_1, S_2, \ldots, S_h\}$. Bribing the voters $v_1, \ldots, v_{2h-1}$ to shift p to the first position costs $h \cdot (r+1) = B$. This shift action makes p gain B points, d lose h points because p passes d in each of these h voters' preference orders, and each element alternative a_i lose at least one point because $S_1, S_2, \ldots, S_h$ cover all elements from $\mathscr{U}$. This makes p, d and every element alternative a_i have $L + h \cdot (r+1)$ points each and tie for victory.

For the "if" part, assume that there is a successful shift action SA that ensures p's victory and affects $n_a = h$ voters only. Since the budget is $B = h \cdot (r+1)$ and we use unit price functions, we can use at most B unit shifts and the score of p can increase to at most $L + B$. Since d has score $L + B + h$ and we can affect h voters only, for each affected voter, p must pass d in his preference order.

As already discussed while describing the construction, in the preference orders of the voters from the voter set $V \setminus \{v_{2i} \mid 1 \le i \le r\}$, either p is already in front of d, or d is at least $B+1$ positions in front of p. Given our budget and unit prices, this implies that our shift action SA involves only the voters from $\{v_{2i} \mid 1 \le i \le r\}$.

Further, since the score of p can increase to at most $L + B$ and each element alternative a_i, prior the shifts, has score $L + B + 1$, p must pass each u_i at least once.

This means that the voters affected under SA correspond to a cover of $\mathscr{U}$ by h sets from $\mathscr{F}$.

The above argument applies to unit prices but, by Proposition 7.1, it also immediately covers the case of convex prices, sortable prices, and thus arbitrary prices. To deal with all-or-nothing price functions, it suffices to use the same construction as above and set the budget B to h, but with the following prices:

- For each element $u_i \in \mathscr{U}$, we set the price function π of voter v_{2i+1} to be $\pi(0) = 0$ and $\pi(j) = 1$ for each $j > 0$.
- For all the remaining voters, we use price function π', such that $\pi'(0) = 0$ and $\pi'(j) = B + 1$ for each $j > 0$.

It is easy to see that the construction remains correct with these price functions. □

As before, we refer the reader to our journal version [Bre+16a] for the proof of the following theorem.

Theorem 7.12. *For the maximin rule and the Copeland rule,* SHIFT BRIBERY *parameterized by the number of affected voters is* W[2]*-hard for each of the price function families we consider.*

When we parameterize SHIFT BRIBERY by the available budget B, the results fall between those from Section 7.5.1 and Theorems 7.11 and 7.12. This is because, depending on the price function family, the budget is upper-bounded by different parameters. These are either the number of unit shifts or the number of affected voters (see Observation 7.2), for which we have different parameterized complexity results. In essence, the hardness proofs for all-or-nothing prices carry over from the parameterization by the number of affected voters to the parameterization by the budget (and this implies hardness for arbitrary prices and sortable prices), while the results for the number of unit shifts carry over to the setting with convex/unit prices.

For all-or-nothing prices, the W[2]-hardness results for the parameter "number of affected voters" (Theorems 7.11 and 7.12) translate because, as already mentioned in Observation 7.2, the budget puts an upper bound on the number of affected voters.

Corollary 7.13. *For Borda, maximin, and Copeland,* SHIFT BRIBERY *parameterized by the budget B is* W[2]*-hard for arbitrary, sortable, and all-or-nothing prices.*

For the Borda rule and the maximin rule, and given convex/unit prices, the FPT results for parameter "the number t of unit shifts" (Theorem 7.3 and Corollary 7.7) translate because for convex/unit prices the budget B is an upper bound on the number of possible unit shifts. For the Copeland rule, the W[1]-hardness result for the same parameter t (Corollary 7.10) translate because for unit prices, $t = B$.

Corollary 7.14. *For convex and unit prices and when parameterized by the budget B, for Borda and maximin,* Shift Bribery *is in* FPT *with running time $O^*((2^B \cdot (B+1) \cdot B)^B)$, while for Copeland it is* W[1]*-hard.*

7.6 Parameterizations by the preference profile measures

In this section we consider Shift Bribery parameterized by either the number of alternatives or the number of voters. Voting with few alternatives is a natural scenario in politics. For example, there is typically only a handful of alternatives in presidential voting. Voting with few voters arises naturally in multiagent systems. For example, Dwork et al. [Dwo+01] suggested voting-based methods for aggregating results from several web search engines. Betzler, Guo, and Niedermeier [BGN10], Brandt et al. [Bra+13], and Fellows et al. [Fel+10] considered winner determination problems with few voters, while Chen et al. [Che+15a] considered voting control by adding alternatives with few voters.

7.6.1 Number of alternatives

As opposed to almost all other (unweighted) voting problems ever studied in computational social choice, for Shift Bribery (and for bribery problems in general) the parameterization by the number of alternatives is one of the most difficult ones. In other voting problems, the natural, standard attack is to use the integer-linear programming (ILP) technique as discussed in Section 2.6.1. For instance, this has been applied successfully to winner determination [BTT89], control [Fal+09b] (also see Chapter 6 for our control model), possible winner [BHN09] (also see Chapter 8), and lobbying [Bre+14c] problems. This works because with m candidates there are at most $m!$ different preference orders, and we can have a variable for each of them in the ILP. However, in our setting this approach fails. The reason is that in bribery problems the voters are not only described by their preference orders, but also by their price functions. This means that we cannot lump together a group of voters with the same preference orders anymore, and we have to treat each of them individually. However, for the case of sortable and all-or-noting price function families, Bredereck et al. [Bre+15d] found a novel way of employing the ILP approach to obtain FPT algorithms.

Dorn and Schlotter [DS12] considered the complexity of Swap Bribery parameterized by the number of alternatives. However, their proof implicitly assumes that each voter has the same price function and, thus, it implies that Shift Bribery parameterized by the number m of alternatives is in FPT for unit prices, but not necessarily for the other families of price functions. Whenever the number of different prices or different price functions is upper-bounded by some constant or, at least, by some function only depending on m, Dorn and Schlotter's approach can be adapted.

For the more general price function families, we were neither able to find alternative FPT attacks nor to find hardness proofs (which, due to the limited number of alternatives one can use and the fact that the voters are unweighted, seem particularly difficult to design). For sortable prices, however, we can show that there is an FPT approximation scheme for SHIFT BRIBERY(O) when parameterized by the number of alternatives.

Theorem 7.15. *Let $\mathcal{R}$ be a voting rule for which winner determination parameterized by the number m of alternatives is in* FPT. *There is a factor-$(1+2\varepsilon+\varepsilon^2)$ approximation algorithm solving $\mathcal{R}$* SHIFT BRIBERY *for sortable prices in time $O^*(M^{M\cdot\lceil \ln(M/\varepsilon)\rceil+1})$ (where $M = m \cdot m!$) times the cost of $\mathcal{R}$'s winner determination.*

We refer the reader to the journal version [Bre+16a] for the proof of Theorem 7.15. Since the publication of our original findings [Bre+14b], Bredereck et al. [Bre+15d] have improved upon our FPT-AS result stated in Theorem 7.15, using mixed integer linear programs to show fixed-parameter tractability for SHIFT BRIBERY; their result also applies to SHIFT BRIBERY(O) because the optimization variant of the feasibility problem of integer linear programs (ILP FEASIBILITY) is also fixed-parameter tractable. Their approach relies on the fact that the fixed-parameter tractability result of ILP FEASIBILITY parameterized by the number of variables [Len83] (adapted to our setting) does not only hold for integer linear programs with $g_1(m)$ integer-valued variables and $g_2(m)$ constraints, but also holds when the program has polynomially many (in the input size) additional *real-valued* variables.

Adopting some of the ideas from the proof of Theorem 7.15, we obtain the following XP result for SHIFT BRIBERY with arbitrary price functions. The corresponding algorithm relies on the fact that for a set of voters with the same preference orders, finding a subset of q voters who shift p higher by exactly the same number j of positions and who have the minimum price is polynomial-time solvable.

Lemma 7.16. *Let $(q_0, q_1, \ldots, q_m) \in \mathbb{N}^{m+1}$ be an $(m+1)$-dimensional vector of non-negative integers. For a set V' of $\sum_{j=0}^{m} q_j$ voters with the same preference orders, it is polynomial-time solvable to compute a cheapest shift action for V' such that for each j with $0 \le j \le m$, the number of voters in V' that shift p higher in their preference orders by j positions is exactly q_j.*

Proof. We show this by reducing our problem to the MINIMUM-COST FLOW problem which, given a *network* $(s,t,U,E,\mathrm{cap},\mathrm{cost})$ and a flow requirement $d \in \mathbb{N}$, asks whether there is a minimum-cost flow f with at least d units. This problem is polynomial-time solvable [AMO93].

We construct the following instance of MINIMUM-COST FLOW. Let $U := \{s,t\} \cup \{u_v \mid v \in V'\} \cup \{w_0, w_1, \ldots, w_m\}$. The vertices s and t are, respectively, the source and the sink. The vertices from $\{u_v \mid v \in V'\}$ form the first layer in the flow network and correspond to the voters in V'. The vertices $w_0, w_1, \ldots, w_m$ form the second layer in the network and correspond to the number j of positions that p should be shifted higher. The set E of arcs, together with their capacities cap and cost, is constructed as follows:

1. For each vertex $v \in V'$, there is an arc with unit capacity and zero cost from source s to u_v.
2. For each vertex $v \in V'$ and each number j, $0 \le j \le m$, there is an arc with unit capacity and cost $\pi(j)$ from u_v to w_j, where π is voter v's price function.
3. For each number j, $0 \le j \le m$, there is an arc from w_j to sink t with capacity q_j and zero cost.

Now, observe that only the vertices w_j are connected to the sink t, with capacity q_j (the third group of arcs). If we require the flow value d to be $\sum_{j=0}^{m} q_j$, then by the third group of arcs, exactly q_j units of flow have to travel from w_j to t. There are polynomial-time algorithms which find a minimum-cost flow of d units from s to t [AMO93]. The constructed flow network has $O(|V'|+m)$ vertices and $O(|V'| \cdot m)$ arcs where the maximum capacity and maximum cost of an arc are $|V'|$ and the maximum cost of shifting p higher by any number of positions, respectively. Thus, our algorithm indeed runs in polynomial time. Furthermore, the flow from vertex u_v to vertex w_j defines the desired shift action in a natural way: the number $f(u_v, w_j)$ of flow units travelling from u_v to w_j corresponds exactly to the number of positions that voter v is going to shift p higher in his preference order. $\square$

For m alternatives, there are at most $m!$ subsets of voters with the same preference order. Thus, we can basically guess a "shifting pattern" for each of these subsets and use the result of Lemma 7.16 to compute a cheapest successful shift action for this pattern.

Theorem 7.17. *Let n denote the number of voters and m the number of alternatives. Let $\mathcal{R}$ be a voting rule whose winner determination procedure is in* XP *when parameterized by m. $\mathcal{R}$* SHIFT BRIBERY *parameterized by m is in* XP *for each price function*

Algorithm 7.1: A brute-force algorithm for SHIFT BRIBERY(O).

Input:
$(A, V, \boldsymbol{\pi}, p)$ — SHIFT BRIBERY(O) instance, m — the number of alternatives

1 $X :=$ the set of all preference orders over A
2 **foreach** *order* $\succ \in X$ **do**
3 $\quad V(\succ) := \{v \in V$ with preference order $\succ\}$
4 **Guess vectors** $\overrightarrow{q(\succ)} := (q_0(\succ), q_1(\succ), \ldots, q_m(\succ)) \in \mathbb{N}^{m+1}$
$\quad$ **with** $\sum_{j=0}^{m} q_j = |V(\succ)|$ **for all** $\succ \in X$
5 $\quad$ **if** *shifting p according to the guessed vectors* $\overrightarrow{q(\succ)}$ *makes p win* **then**
6 $\quad\quad$ Compute a cheapest action SA such that
$\quad\quad\quad$ for each order $\succ$ and each number j, $0 \le j \le m$,
$\quad\quad\quad$ exactly $q_j(\succ) \in V(\succ)$ voters shift p higher by j positions in SA
7 $\quad\quad$ Store SA as a successful shift action
8 **return** the cheapest *stored* successful shift action

family we consider. The algorithm runs in time $O^*((n^m)^{m!})$ times the cost of $\mathscr{R}$'s winner determination.

Proof. Let $I = ((A, V), \boldsymbol{\pi}, p, B)$ be an input instance of SHIFT BRIBERY with $|A| = m$ and $|V| = n$. For each preference order $\succ$ over the alternatives from A, let $V(\succ)$ denote the set of voters that have this preference order $\succ$. We find a cheapest successful shift action for p by using brute-force search described in Algorithm 7.1. Since, in essence, the algorithm tries all interesting shift actions, it is clear that it is correct. Note that if the considered voting rule is anonymous, then shifting p according to the guessed vectors $\overrightarrow{q(\succ)}$ simply means that for each preference order $\succ$ and each number j, p is shifted higher by j positions in the preference orders of $q_j(\succ)$ voters. We show that the algorithm has the desired running time.

In Step 4, the algorithm has $O((n^m)^{m!})$ possible guesses so we can try them all. The condition in Step 5 can be checked in XP time if the given voting rule is anonymous and determining a winner runs in XP time. However, it is not clear how to implement the vectors $\overrightarrow{q(\succ)}$ for each preference order $\succ$ in Step 5. Fortunately, by Lemma 7.16, we can reduce our problem to the MINIMUM-COST FLOW problem and find a cheapest shift action for each preference order $\succ$. This still runs in XP time because each of the constructed network instances has a size polynomial in $|I|$ (Lemma 7.16). In total, we obtain an algorithm with the desired running time for SHIFT BRIBERY(O).

To solve the decision variant, we only need to check whether the cheapest cost found is at most B. □

7.6.2 Number of voters

As for Shift Bribery parameterized by the number of voters, we do not find FPT algorithms for our rules in this setting, but we find a general FPT approximation scheme. The idea of our algorithm is to use a scaling technique combined with a brute-force search through the solution space. The scaling part of the algorithm first reduces the range of prices. Then, the brute-force search finds a near-optimal solution. The ideas underlying the proof are similar to those of Elkind and Faliszewski [EF10, Proposition 2]

Theorem 7.18. *Let m denote the number of alternatives and n the number of voters. Let $\mathcal{R}$ be a voting rule for which winner determination parameterized by n is in* FPT. *There is a factor-$(1+\varepsilon)$ approximation algorithm solving* Shift Bribery(O) *for voting rule $\mathcal{R}$ in time $O^*(\lceil n/\varepsilon+1\rceil^n)$ times the cost of $\mathcal{R}$'s winner determination.*

Proof. Let $I = (A, V, \boldsymbol{\pi}, p)$ be an instance of Shift Bribery(O). Further, let $\varepsilon > 0$ be the desired approximation parameter.

Intuitively, the idea of the algorithm is as follows. First, guess the maximum budget $\pi_{\max}$ to spend on a single voter. Then, rescale and round the price $\boldsymbol{\pi}$ to not exceed a given bound; the reason for this rescaling is to find a successful shift action (with provable good approximation) in FPT time. Finally, find a cheapest shift action according to the new price. Lines 13–16 in Algorithm 7.2 describe these three main steps. After trying each guess for $\pi_{\max}$, we return the cheapest successful shift action that we obtained (Lines 17–19).

We first show that the found shift action is successful for p, and then that this action has cost at most $(1+\varepsilon)\cdot \mathsf{OPT}(I)$ if I admits a successful shift action with cost $\mathsf{OPT}(I)$.

Suppose that SA^* is a successful shift action for p with minimum cost $\mathsf{OPT}(I)$. Let $\pi_{\max}$ be the highest budget spent on a single voter through SA^*. Then, it is easy to see that by brute-forcing (Lines 13–14), we can find this value. Next, we use NewPrice to construct new price functions $\boldsymbol{\pi}'$ for all voters; the values of these functions depend on $\pi_{\max}$ (Lines 1–4).

Finally, we use ScaledCheapestAction to find a cheapest shift action for the newly constructed prices $\boldsymbol{\pi}'$: Observe that if $\pi_{\max}$ is indeed the highest cost among the voters, then for each voter $v \in V$, it must hold that $\boldsymbol{\pi}(v)(\text{SA}^*(v)) \le \pi_{\max}$. This implies by the construction of $\boldsymbol{\pi}'$, that $\boldsymbol{\pi}'(v)(\text{SA}^*(v)) \le \lceil \frac{n}{\varepsilon} \rceil$. We will see later that this bound is

Algorithm 7.2: FPT approximation scheme for SHIFT BRIBERY(O) parameterized by the number of voters.

Input:
$(\mathscr{P} = (A, V), \boldsymbol{\pi}, p)$ — SHIFT BRIBERY(O) instance, ε — approximation ratio,
m — the number of alternatives, n — the number of voters.

1 NEWPRICE$(\pi_{\max})$:
2 **foreach** *voter* $v \in V$ **and** *number* $j \in \{0, 1, \ldots, m\}$ **do**
3 $$\boldsymbol{\pi}'(v)(j) := \begin{cases} \lceil \frac{n \cdot \boldsymbol{\pi}(v)(j)}{\varepsilon \cdot \pi_{\max}} \rceil, & \text{if } \boldsymbol{\pi}(v)(j) \leq \pi_{\max}, \\ \lceil \frac{n \cdot (n+1)}{\varepsilon} \rceil, & \text{otherwise.} \end{cases}$$
4 **return** $\boldsymbol{\pi}'$

5 SCALEDCHEAPESTACTION $(\boldsymbol{\pi}')$:
6 Set $\mathrm{SA}' \colon V \to \mathbb{N}$ with $\mathrm{SA}'(v) = m$ for all $v \in V$
7 **foreach** *budget function* $f \colon V \to \{0, 1, \ldots, \lceil n/\varepsilon \rceil\}$ **do**
8 Compute a shift action SA such that for each voter $v \in V$, $\mathrm{SA}(v)$ is the maximum number j with $\boldsymbol{\pi}'(v)(j) \leq f(v)$
9 **if** *p wins in shift*$(\mathscr{P}, \mathrm{SA})$ **and** $\mathrm{price}(\boldsymbol{\pi}', \mathrm{SA}) \leq \mathrm{price}(\boldsymbol{\pi}', \mathrm{SA}')$ **then**
10 $\mathrm{SA}' := \mathrm{SA}$
11 **return** SA'

12 Set $\mathrm{SA}' \colon V \to \mathbb{N}$ with $\mathrm{SA}'(v) = m$ for all $v \in V$
13 **foreach** *voter* $v^* \in V$ **and** *number* $j^* \in \{0, 1, \ldots, m\}$ **do**
14 Set $\pi_{\max} := \boldsymbol{\pi}(v^*)(j^*)$
15 Set $\boldsymbol{\pi}' :=$ NEWPRICE$(\pi_{\max})$
16 Set SA := SCALEDCHEAPESTACTION$(\boldsymbol{\pi}')$
17 **if** $\mathrm{price}(\boldsymbol{\pi}', \mathrm{SA}) \leq \mathrm{price}(\boldsymbol{\pi}', \mathrm{SA}')$ **then**
18 $\mathrm{SA}' := \mathrm{SA}$
19 **return** SA'

important for the analysis of the FPT running time. Thus, we can brute-force search all possible budget functions f that give each voter at most $\lceil \frac{n}{\varepsilon} \rceil$ units (Line 7) and find one shift action that shifts p forward as much as possible in the preference order of each voter $v \in V$, without exceeding the budget $f(v)$ (Line 8). Finally, we pick a successful action that has the smallest cost (Lines 9–11).

As already mentioned, after trying each value for $\pi_{\max}$, we return the cheapest successful shift action that we obtained.

To determine the cost of our found shift action, consider an iteration where the value of $\pi_{\max}$ is indeed the most expensive budget spent on a voter in $\textsc{Sa}^*$, our "optimal" action. Let $\boldsymbol{\pi}' := \textsc{NewPrice}(\pi_{\max})$ and $\textsc{Sa}' := \textsc{ScaledCheapestAction}(\boldsymbol{\pi}')$ accordingly. Then, for each voter $v \in V$, we have that $\boldsymbol{\pi}(v)(\textsc{Sa}^*(v)) \le \pi_{\max}$. This implies the following upper bound for the cost of our successful shift action $\textsc{Sa}'$ using the rescaled prices $\boldsymbol{\pi}'$:

$$
\begin{aligned}
\mathsf{price}(\textsc{Sa}', \boldsymbol{\pi}') &= \sum_{v \in V} \lceil \frac{n \cdot \boldsymbol{\pi}(v)(\textsc{Sa}^*(v))}{\varepsilon \cdot \pi_{\max}} \rceil \le \sum_{v \in V} (\frac{n \cdot \boldsymbol{\pi}(v)(\textsc{Sa}^*(v))}{\varepsilon \cdot \pi_{\max}} + 1) \\
&= \frac{n}{\varepsilon \cdot \pi_{\max}} \mathsf{OPT}(I) + n. \qquad (7.1)
\end{aligned}
$$

The first equality holds by the definition of $\boldsymbol{\pi}'$. The second inequality holds because of the rounding. The third equality holds because $\textsc{Sa}$ has the minimum cost for prices $\boldsymbol{\pi}$. Since $\textsc{Sa}'$ has the minimum cost for prices $\boldsymbol{\pi}'$, we obtain the following lower bound for the successful shift action $\textsc{Sa}^*$ at the rescaled prices $\boldsymbol{\pi}'$:

$$
\begin{aligned}
\mathsf{price}(\textsc{Sa}, \boldsymbol{\pi}') &\ge \mathsf{price}(\textsc{Sa}', \boldsymbol{\pi}') \ge \sum_{v \in V} \frac{n \cdot \boldsymbol{\pi}(v)(\textsc{Sa}'(v))}{\varepsilon \cdot \pi_{\max}} \\
&= \frac{n}{\varepsilon \cdot \pi_{\max}} \mathsf{price}(\textsc{Sa}', \boldsymbol{\pi}). \qquad (7.2)
\end{aligned}
$$

Combining (7.1) and (7.2), we obtain

$$
\mathsf{price}(\textsc{Sa}', \boldsymbol{\pi}) \le \mathsf{OPT}(I) + \varepsilon \cdot \pi_{\max} \le (1 + \varepsilon) \cdot \mathsf{OPT}(I). \qquad (7.3)
$$

The second inequality holds because $\pi_{\max} \le \mathsf{OPT}(I)$. Thus the algorithm returns a $(1 + \varepsilon)$-approximate solution.

As for the running time, we have $n \cdot m$ possibilities of choosing $\pi_{\max}$. For each selection, we run $\textsc{NewPrice}$ in polynomial time; then, $\textsc{ScaledCheapestAction}$ tries $O(\lceil n/\varepsilon + 1 \rceil^n)$ budget functions and finds a corresponding shift action with the minimum cost. Thus, in total, the algorithm runs in $\lceil n/\varepsilon + 1 \rceil^n \cdot |I|^{O(1)}$ time. □

Is it possible to obtain full-fledged FPT algorithms for the parameterization by the number of voters? For the case of all-or-nothing price functions, we provide a very simple FPT algorithm, but for the other price-function classes the answer is no (under the assumption that $\mathsf{FPT} \neq \mathsf{W[1]}$).

Proposition 7.19. *Let $\mathcal{R}$ be a voting rule for which winner determination parameterized by the number of voters is in* FPT. *For all-or-nothing prices, $\mathcal{R}$* Shift Bribery *parameterized by the number n of voters can be solved in time $O^*(2^n)$ times the cost of $\mathcal{R}$'s winner determination, implying fixed-parameter tractability.*

Proof. Note that with all-or-nothing prices, it suffices to consider shift actions where for each voter's preference order we either shift the preferred alternative to the top or do not shift her at all. Thus, given a preference profile with n voters it suffices to try each of the 2^n possible shift actions of this form. □

In contrast to Proposition 7.19, fixed-parameter tractability for other price functions is not obvious. Indeed, for Copeland we can show W[1]-hardness for unit prices, via a somewhat involved reduction from a variant of the CLIQUE problem. We refer the reader to the journal version [Bre+16a] for the proof.

Theorem 7.20. COPELAND SHIFT BRIBERY *parameterized by the number of voters is* W[1]*-hard for unit prices.*

Since determining the winner under the Copeland rule can be done in polynomial-time, the above W[1]-hardness result shows that unless unlikely complexity class collapses occur, it is impossible to improve Theorem 7.18 to provide an *exact* FPT algorithm. Since the publication of our conference paper [Bre+14b], Bredereck et al. [Bre+16c] complemented our W[1]-hardness result by showing that for the Borda and maximin rules, SHIFT BRIBERY parameterized by the number of voters is W[1]-hard.

7.7 Concluding remarks

We have studied the parameterized complexity of SHIFT BRIBERY under the voting rules Borda, maximin, and Copeland, for several natural parameters that either describe the nature of the solution or the size of the election, and for several families of price functions (arbitrary, convex, unit, sortable, and all-or-nothing). Our results confirm the intuition that the computational complexity depends on all three factors: the voting rule, the parameter, and the type of price function used.

Our work leads to some natural follow-up questions. First, it would be interesting to solve the complexity of SHIFT BRIBERY parameterized by the number of alternatives for arbitrary price functions (see Table 7.1); at the moment we only know that it is W[1]-hard and in XP. For both Borda and maximin, and for the parameter "number t of unit shifts", we obtain a partial problem kernel: we can shrink in polynomial time a given instance to one whose numbers of alternatives and voters are upper-bounded by a function in t (Theorem 7.5 and Corollary 7.7). This function, however, is an exponential function. Thus, it would be interesting to know whether it is possible to design an efficient kernelization algorithm that replaces this exponential function with a polynomial. A concept similar to the kernelization lower bound would be to show a lower bound on the size of *only a part* of the problem kernel; see Section 2.6.1 for more information on kernelization lower bounds.

Furthermore, we have not put any restrictions on the preference profiles in the input instance of SHIFT BRIBERY so far. Our work in Chapter 3 and Chapter 5 naturally leads to the question how the problem behaves when the preference orders are structured or nearly structured.

Going into the direction of the MARGIN OF VICTORY type of problems, it would be interesting to study DESTRUCTIVE SHIFT BRIBERY, where we can push back the despised alternative to prevent it from being a winner. Initial results in this direction are due to Kaczmarczyk and Faliszewski [KF16].

Finally, in Chapter 6, we introduced a model of affecting voting outcomes by adding bundles of voters to the profile (this is an extension of a well-studied problem known as CONTROL BY ADDING VOTERS [BTT92, HHR07]). Bredereck et al. [Bre+15e] considered SHIFT BRIBERY in a similar setting where one can affect multiple voters at the same time (for example, airing an advertisement on TV could affect several voters at the same time). Unfortunately, most of their results are quite negative and it would be interesting to investigate the combinatorial variant of SHIFT BRIBERY from the point of view of heuristic algorithms.

Last but not least, complementing our theoretical work with real-world experiments, starting with tuning the FPT algorithms, would be a promising future research direction.

CHAPTER 8

Parliamentary Voting Rules

Logic is concerned with the truth-value of sentences. Grammar is concerned with the communications-value of sentences. Rhetoric is concerned with the persuasion-value of sentences. And heresthetic is concerned with the strategy-value of sentences. In each case, the art involves the use of language to accomplish some purpose: to arrive at truth, to communicate, to persuade, and to manipulate.

William H. Riker

In the previous two chapters, we have seen two natural manipulative voting attack problems and we have studied the computational complexity of these problems for several common voting rules, including the plurality rule and the Condorcet rule. Notably, all these voting rules proceed in a single stage—be it based on scoring or on pairwise comparisons. In this final chapter of the second part of the thesis, we examine two prominent parliamentary voting rules, the amendment rule and the successive rule. They work in multiple stages where the result of each stage may influence the result of the next stage. Both rules proceed according to a given linear order of the alternatives, an *agenda*. We study computational problems associated with these two rules and obtain the following results: On the one hand, making a specific alternative win by adding the fewest number of manipulators (with arbitrary preferences) or by finding a suitable ordering of the agenda, the agenda control problem, takes polynomial time. On the other hand, our experimental results with real-world data indicate that most preference profiles cannot be manipulated by only few voters and a successful agenda control for some given alternative is typically impossible. If the voters' preferences are incomplete, then deciding whether an alternative can possibly win is NP-complete for both rules. Whilst deciding

This chapter is based on "Parliamentary Voting Procedures: Agenda Control, Manipulation, and Uncertainty" by R. Bredereck, J. Chen, R. Niedermeier, and T. Walsh, *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI '15)* [Bre+15b] and *arXiv:1509.02424v1 [cs.GT]* [Bre+15c] if an alternative necessarily wins is coNP-complete for the amendment rule, it is polynomial-time solvable for the successive rule.

8.1 Introduction

Two prominent voting rules, the successive rule and the amendment rule, are used in many parliamentary chambers to amend and decide upon new legislation [ABM14, Bla58, Far69, Mil77]. Both are *sequential (or multi-stage) voting rules*: the alternatives are ordered (thus forming an *agenda*) and they are considered one by one, making a binary decision based on majority voting in each step. In a nutshell, in each step, the *successive rule* decides whether to accept the current alternative (in which case the procedure stops and the winner is determined) or to reject it and continuing with the remaining alternatives in the given order. The *amendment rule* in each step eliminates one of two current alternatives by means of a majority vote, putting the survivor up against the next alternative on the agenda.

There are many reasons to study the properties of parliamentary voting rules, and especially to consider computational questions: First, parliamentary voting rules are used very frequently in practice. For example, the recent 112th Congress of the US Senate and House of Representatives had 1030 votes to amend and approve bills.[1] This does not take into account the hundreds of committees that also amended and voted on these bills. As a second example, there were 351 divisions within the UK Houses of Lords and Commons in 2013 to amend or approve bills. Both countries use the amendment rule to make parliamentary decisions [Ras00].

Second, parliamentary voting rules are used to make some of the most important decisions in society. We decide to reduce carbon emissions, provide universal health care, or raise taxes based on the outcome of such voting rules. When rallying support for new legislation, it is vital to know what amendments can and cannot be passed. Fortunately, we have excellent historical voting records for parliamentary chambers. We can therefore make high quality predictions about how sincere or “sophisticated/strategic” voters will vote.

Third, Enelow and Koehler [EK80] examined several votes in the US House of Representatives in 1977 and gave evidence that parliamentary voting may be strategic:

> *“Thus, it is shown that sophisticated voting does occur in Congress and in fact is encouraged by the way amendments are used in the legislative process. It should not come as a surprise to congressional scholars that congressmen do not always vote sincerely.”*

[1] Retrieved from www.govtrack.us/ for years 2011-2012.

Fourth, there is both theoretical and empirical evidence that the final outcome critically depends on the order in which amendments are presented. For example, Ordeshook and Schwartz [OS87] remarked that

> *"...legislative decisions are at the mercy of elites who control agendas."*

It is therefore interesting to ask whether, for example, computational complexity is a barrier to the control of the agenda (see Section 8.5 for more information on this question) or to strategic voting (see Section 8.6 for more information on this question) when such parliamentary voting rules are used. It is also interesting to ask if we can efficiently compute whether a particular amendment can possibly (or necessarily) pass despite uncertainty in the votes or the agenda. This refers to the POSSIBLE WINNER and NECESSARY WINNER problems (see Section 8.7). For each voter, we assume that we know his preference order over the alternatives, which is a linear order in the first two types of problems and is a partial order in the last type of problems. As already mentioned in the previous chapters, this assumption seems somewhat unrealistic. However, it is a standard assumption within computational social choice, and we might have a good approximation of this knowledge from pre-election polls.

We provide one of the first computational studies of the two parliamentary voting rules, giving both theoretical and empirical results. We emphasize that, while complementing theoretical research with an empirical study of real-world data is useful, this is the only chapter of the thesis where we perform experiments. Indeed, we only perform experiments for the first two problems studied in this chapter. We do this for three simple reasons. First, POSSIBLE WINNER and NECESSARY WINNER consider partial preference orders, but the real-world data for which we perform experiments only have a very restricted form of partial preference orders. Second, the algorithms solving AGENDA CONTROL and COALITIONAL MANIPULATION run in no more than cubic time (in the size of the input instance), and most remaining problems are NP-complete. Third, in this thesis, we chiefly want to clarify the boundary between fixed-parameter tractable and parameterized intractable cases. We do identify some tractable cases, but the algorithms behind those combine brute-forcing with integer linear programming. To run experiments however would require significant improvements on these algorithms to obtain a reasonable running time, which is outside the scope of this thesis. See Section 9.2 for a more detailed explanation.

Related work. There are many studies in the economic and political science literature on parliamentary voting rules, starting with Black [Bla58] and Farquharson [Far69], concerning "insincere" or "sophisticated" or "strategic" voting [Ban85, EK80, Mil77, MN78, Mou86, OS87, SW82]. Apesteguia, Ballester, and Masatlioglu [ABM14] characterized both the amendment and the successive rules from an axiomatic perspective.

Miller [Mil77] studied which alternatives can become a winner by altering the agenda, the AGENDA CONTROL problem. In particular, he showed through Propositions 1 and 2 in his paper [Mil77] that if each voter votes sincerely, then an alternative can become an amendment winner under some agenda if and only if it belongs to the *Condorcet set* (also known as *top cycle* [Nur05]); a Condorcet set is the smallest set of alternatives that beats every other alternative outside this set. We extend this result by a constructive proof. For the successive rule, however, Miller only showed that every alternative from the Condorcet set can win. Barberà and Gerber [BG16] followed Miller's research of characterizing the set of alternatives that may become an amendment (or a successive) winner by controlling the agenda, assuming that the voters are voting sophisticatedly. Rasch [Ras14] empirically examined the behavior of voters in the Norwegian parliament, where the successive rule is used. He reported that successful insincere voting, where voters may vote differently from their true preferences and the outcome is better for them, is very rare.

Using computational complexity as a barrier against manipulation (that is, against changing the outcome by adding voters) was initiated by Bartholdi III, Tovey, and Trick [BTT89]. They showed that manipulating a special variant of the Copeland voting rule is NP-complete. Bartholdi III and Orlin [BO91] showed that manipulating the Single Transferable Vote (STV) rule is NP-hard even if there is only one manipulator. This voting rule and its many variants have been adopted in the parliamentary voting of many countries. They are sequential voting rules and work similarly to the successive rule except that there is no agenda. Instead, in each step, the alternative that is ranked first by the least number of voters will be deleted from the profile. This means that if an alternative is an STV winner, then there is an agenda for which the same alternative becomes a successive winner. The NP-hardness result for manipulating STV is of particular interest since we provide polynomial-time algorithms for manipulating the closely related successive rule. These two complexity results indicate that it is the agenda that makes an important difference.

Concerning uncertainty in the profiles (that is, each voter's preference order is a partial order), there is some work in the political science literature [Jun89, OP88], but there seems to be significantly more activity on the computational side. Konczak and Lang [KL05] introduced the problems of possible winner and necessary winner determination, and studied them for the Condorcet rule. Since then, these problems have been frequently studied for several other common voting rules [Azi+15, BD10, BHN09, BR12, Haz+12, Wal07].

Moulin [Mou86] discussed the *voting tree* rule which is a generalization of the amendment rule. This general rule employs a binary voting tree where

- the leaves represent the alternatives such that each alternative is represented by at least one leaf, and
- each internal node represents the alternative that wins the pairwise comparison of its children.

The alternative represented by the root defines the winner. If the binary tree is degenerate and if each alternative is represented by exactly one leaf, then this rule is identical to the amendment rule. To tackle the manipulation problem with weighted voters, Conitzer, Sandholm, and Lang [CSL07] provided a cubic-time algorithm for the voting tree rule while our quadratic-time algorithm (Theorem 8.11) is tailored for the amendment rule. Xia and Conitzer [XC11] provided intractability results for the possible (resp. necessary) winner determination problem when the given tree is balanced. Pini et al. [Pin+11] and Lang et al. [Lan+12] showed that the possible (resp. necessary) winner determination problem with weighted voters is NP-complete (resp. coNP-complete) even for a constant number of voters (see Table 8.1 for an overview of known and our new results).

8.2 Results

We investigate computational problems for two prominent parliamentary voting rules: the successive rule and the amendment rule. We study three types of voting problems. First, we study the AGENDA CONTROL problem that asks whether there is an agenda under which a given alternative can win when the voters vote sincerely. Second, we study the COALITIONAL MANIPULATION problem that asks whether a given alternative can win by adding a given number of voters. Third, we study whether a given alternative can possibly (resp. necessarily) win when the voters may have incomplete preferences; we call the corresponding problem POSSIBLE WINNER (resp. NECESSARY WINNER). We also consider the case where each voter has a weight and we add the prefix WEIGHTED to the name of the corresponding problem. See Table 8.1 for an overview of our theoretical results.

Our polynomial-time algorithms for AGENDA CONTROL and COALITIONAL MANIPULATION indicate that the amendment rule is computationally more expensive than the successive rule. From a computational perspective, this implies that the amendment rule may be more resistant to manipulation and agenda control than the successive rule. If the voters' preference orders are incomplete, then deciding whether an alternative possibly wins is NP-complete for both rules. Furthermore,

Problem	Successive	Amendment	References
AGENDA CONTROL	$O(n \cdot m^2)$	$O(n \cdot m^2 + m^3)^\heartsuit$	Thm 8.4 & 8.7
W. COALITIONAL MANIPULATION	$O((n+k) \cdot m)$	$O((n+k) \cdot m^2)$	Thm 8.9 & 8.11, Cor 8.12
POSSIBLE WINNER	NP-c ILP-FPT(m)	NP-c ILP-FPT(m)	Thm 8.14 & 8.15 Cor 8.17
NECESSARY WINNER	$O(n \cdot m^3)$	coNP-c ILP-FPT(m)	Thm 8.21 & 8.22 Cor 8.23
W. POSSIBLE WINNER	NP-c $(m=3)$	NP-c $(m=3)^\spadesuit$	Thm 8.24
W. NECESSARY WINNER	$O(w \cdot n \cdot m^3)$	$O(n)$ for $m \leq 3$ coNP-c $(m=4)^\spadesuit$	Thm 8.25 —

Table 8.1.: Computational complexity results of this chapter and of the literature for our two parliamentary voting rules. The prefix "W." indicates that the relevant problem has weighted voters. The number of voters is denoted by n, the number of alternatives is denoted by m, the number of manipulators is denoted by k, and the sum of weights of all voters in an instance of the COALITIONAL MANIPULATION problem is denoted by w. Recall that "ILP-FPT(m)" stands for "fixed-parameter tractable with respect to m through the integer linear programming technique"; see Section 2.6.1 for more information. The result marked with $^\heartsuit$ also follows from the work of Miller [Mil77]. Results marked with $^\spadesuit$ follow from the work of Pini et al. [Pin+11] and Lang et al. [Lan+12]. Entries containing statements of the form "NP-c $(m = z)$" (resp. "coNP-c $(m = z)$") mean that the respective problem is in NP (resp. coNP) and is NP-hard (resp. coNP-hard) even with only z many alternatives. All hardness results hold even when the agenda is a linear order. Note that the weakly NP-hardness result of Theorem 8.24 implies NP-completeness.

deciding whether an alternative necessarily wins is polynomial-time solvable for the successive rule, but it is coNP-complete for the amendment rule.

Our experimental results with real-world profiles indicate that a successful agenda control is very rare and the size of a coalition for a successful manipulation is at least half of the number of the voters in the original profile. Thus, we can conclude that our parliamentary voting rules are still "safe" against these voting attacks in the real world.

8.3 Chapter outline

In Section 8.4, we provide additional definitions specific to this chapter. In Section 8.5, we focus on AGENDA CONTROL when voters have sincere preference orders and we study the corresponding computational complexity for both parliamentary voting rules. In Section 8.6, we deal with strategic behavior of voters and investigate the computational complexity of COALITIONAL MANIPULATION. In Section 8.7, we are interested in situations with uncertainty where voters' preferences and the agenda are still incomplete. We study the problem of whether an alternative can possibly/will necessarily win in such a situation (the POSSIBLE WINNER problem/the NECESSARY WINNER problem). In Section 8.8, we complement our theoretical study with an experimental evaluation of AGENDA CONTROL and COALITIONAL MANIPULATION for real-world profiles. Section 8.9 concludes our work and presents some challenges for future research.

8.4 Specific notations and definitions

Let $\mathcal{P} = (\mathcal{A}(\mathcal{P}), \mathcal{V}(\mathcal{P}))$ be a preference profile with $\mathcal{A}(\mathcal{P})$ being the set of alternatives and $\mathcal{V}(\mathcal{P})$ being the set of voters. Consider two distinct alternatives $b, c \in \mathcal{A}(\mathcal{P})$ in $\mathcal{P}$. We previously defined that alternative b *beats* alternative c (in a head-to-head contest) when a majority of voters prefers b to c. In this chapter, we call b the *survivor* and c the *loser* of the two alternatives. Let $\mathcal{P}' = (A', V', (\succ'_1, \succ'_2, \ldots, \succ'_n))$ be a second profile for the same set of alternatives and the same set of voters. If for each voter $v_i \in \mathcal{V}(\mathcal{P})$, the preference order $\succ'_i$ from $\mathcal{P}'$ is a superset of the preference order $\succ_i$ from $\mathcal{P}$, then we say that $\mathcal{P}'$ *extends* $\mathcal{P}$. Note that a preference order is in effect a set of ordered pairs of alternatives. If each $\succ'_i$ is even a linear order, then we say that $\mathcal{P}'$ *completes* $\mathcal{P}$.

We can use a directed graph to illustrate the head-to-head contests between every two alternatives. Section 2.2.2 states relevant concepts regarding directed graphs.

Definition 8.1 ((Weighted) majority graph)**.** Given a preference profile $\mathcal{P} = (A, V)$, we construct an arc-weighted directed graph $G := (U, E)$, where U consists of a vertex u_j for each alternative $c_j \in A$ and where there is an arc from vertex u_j to vertex $u_{j'}$ with weight w if w voters prefer c_j to $c_{j'}$. We call the constructed graph G a *weighted majority graph* for profile $\mathcal{P}$. If each voter comes with a weight, then we update the weights of the arcs accordingly.

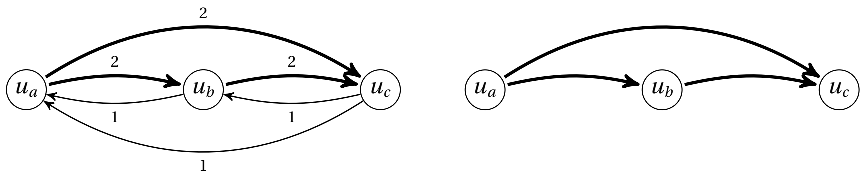
If we ignore the weights and the arcs with weights of at most $|V|/2$, then we obtain a *majority graph* (without weights) for the profile $\mathcal{P}$.

We provide a small example to illustrate the concept of (weighted) majority graphs.

Example 8.1. Let $\mathcal{P}$ be a preference profile with three alternatives a, b, c, and three voters v_1, v_2, v_3 whose preference orders are specified as follows:

$$v_1: a \succ b \succ c, \quad v_2: b \succ a \succ c, \quad v_3: c \succ a \succ b.$$

The weighted majority graph for $\mathcal{P}$ consists of three alternatives and six weighted arcs as depicted in the left figure below. Bold arcs indicate the majority relation.



The corresponding majority graph without weights is depicted in the right figure. The graph shown in the right figure is a tournament. Further, vertex u_a has exactly two out-going arcs, meaning that alternative a beats the other two alternatives. Thus, it is a Condorcet winner.

We consider two of the most common parliamentary voting rules. For both rules, we assume that a linear order over the m alternatives in A is given. We refer to this linear order $\mathcal{L}$ as an *agenda*. If this order is a partial order, then we call it a *partial agenda*, denoted by $\mathcal{B}$. We use the symbol $\rhd$ for the agenda order to distinguish it from the preference order $\succ$: For two distinct alternatives a and b, $a \rhd b$ means that b is considered after (or ordered behind) a in the agenda.

Roughly speaking, the successive rule determines a winner that is the first alternative in the agenda $\mathcal{L}$ such that a majority of voters prefers it to every alternative ordered behind it in $\mathcal{L}$.

Definition 8.2 (Successive rule for a given agenda $\mathcal{L}$)**.** There are at most m rounds with m being the number of alternatives. Starting with round $i := 1$, we repeat the following until we find a successive winner: Let c be the i^{th} alternative in the given agenda $\mathcal{L}$. If there is a majority of voters who prefer alternative c to every alternative that is ordered behind it in $\mathcal{L}$, then c is the decision and we call it a successive winner. Otherwise, we proceed to round $i := i + 1$.

In Europe, the successive rule is used in many parliamentary chambers including those of Austria, Belgium, Denmark, France, Germany, Greece, Iceland, Ireland, Italy, Luxembourg, the Netherlands, Norway, Portugal, and Spain [Ras00].

Example 8.2. Let us look at the profile $\mathcal{P}$ from Example 8.1 again. Consider the following agenda $\mathcal{L}$: $a \rhd b \rhd c$ for the successive rule. We have three alternatives. Thus, the procedure ends after at most three rounds. In the first round, since less than half of the voters prefer a to both b and c (only v_1 does), a is not a successive winner, although it is a Condorcet winner. In the second round, since more than half of the voters prefer b to c (voters v_1 and v_2), b is the successive winner.

The amendment rule determines a winner that is the first alternative in the agenda $\mathscr{L}$ to beat every other alternative that is ordered behind it in $\mathscr{L}$.

Definition 8.3 (Amendment rule for a given agenda $\mathscr{L}$)**.** It proceeds in m rounds with m being the number of alternatives. We define an amendment winner for each round. Let the 1^{st}-round amendment winner be the 1^{st} alternative in $\mathscr{L}$. Then, for each round $2 \le i \le m$, the i^{th}-round amendment winner is either the $(i-1)^{\text{th}}$-round amendment winner b or the i^{th} alternative c in $\mathscr{L}$: if a majority of voters prefers b over c, then it is b. Otherwise it is c. We define the m^{th}-round amendment winner to be the amendment winner.

In Europe, the amendment rule is used in the parliamentary chambers of Finland, Sweden, Switzerland, and the United Kingdom. It is also used in the U.S. Congress and several other countries with Anglo-American ties [Ras00].

We again use Example 8.1 to illustrate how the amendment rule works. It may be helpful to consider the corresponding majority graph (see Definition 8.1).

Example 8.3. Consider the profile and the agenda $\mathscr{L}\colon a \rhd b \rhd c$ given in Example 8.2. Alternative a is the 1^{st}-round winner since it is the first alternative in $\mathscr{L}$. Since a majority of voters prefers a to b and a majority of voters prefers a to c, alternative a is both the 2^{nd}-round and the 3^{rd}-round winner. Hence, it is the amendment winner. Indeed, as shown in the majority graph (see Example 8.1), alternative a is a Condorcet winner.

As already observed by Miller [Mil77], there is a close relation between Condorcet winners and amendment winners:

Observation 8.1 ([Mil77])**.** *A Condorcet winner is an amendment winner, no matter what the agenda looks like.*

We close this section with some remark. For the remainder of this chapter, we assume that the number of voters is *odd* to reduce the impact of tie breaking and we leave the study of different tie breaking rules as future work. We consider both unweighted voters and voters with integer weights. The weighted case is especially interesting in the parliamentary setting: First, there are parliamentary chambers where voters are weighted (for instance, in the Council of Europe, preference orders are weighted by the size of the country). Second, voters will often vote along party lines. This effectively gives us parties casting weighted preference orders. Third, the weighted case can inform the situation where we have uncertainty about the preference orders. For example, Theorem 15 of Conitzer, Sandholm, and Lang

[CSL07] proves that if the manipulation problem for a voting rule is NP-hard for weighted voters with complete preference orders, then deciding who possibly wins in the unweighted case is NP-hard even when there is only a limited form of uncertainty about the preference orders. It would be interesting to prove similar results about uncertainty and weighted preference orders for parliamentary voting rules.

8.5 Agenda Control

The order of the alternatives, that is, the agenda, may depend on the speaker, the Government, logical considerations (for instance, the status quo goes last, the most extreme alternative comes first), the chronological order of submission, or other factors. The agenda used can have a major impact on the final decision. It is also worth noting that there are many possible agendas.

For example, if we use the amendment procedure, then the Condorcet winner is the only amendment winner (Observation 8.1). But the Condorcet winner is only guaranteed to be a successive winner if it is introduced in one of the last two positions in the agenda. To illustrate this fact, consider the profile given in Example 8.1 again. If alternative a is the first to be considered in the agenda, then a will be eliminated because it is not a majority winner (with respect to the rest). Thus, it cannot become a successive winner. However, if a is not the first one to consider in the agenda, then it will become a successive winner because the first one (either b or c) in the agenda will be eliminated and in the next round where two alternatives remain, a beats the other one which is not eliminated. We therefore consider the following computational question for the situation where voters vote sincerely.

AGENDA CONTROL
Input: A preference profile $\mathcal{P} := (A, V)$ with linear preference orders and a preferred alternative $p \in A$.
Question: Is there an agenda for A such that p is the overall winner?

We find that both voting rules are "vulnerable" to agenda control. In particular, we show how to find in polynomial time an appropriate agenda (if it exists) so that the preferred alternative can become a successive (resp. amendment) winner. In the remainder of this section, we assume that a preference profile $\mathcal{P} := (A, V)$ with linear preference orders and a preferred alternative $p \in A$ are given. We assume that the voters vote sincerely according to their linear preference orders. If the agenda is not revealed in advance, risk averse voters are likely to vote sincerely.. Note that this computational question also applies to the situation where voters are strategic, the chair collects there possibly insincere votes, and then decides on an agenda to give a particular outcomes. Recall that n denotes the number of voters and m denotes the number of alternatives in a given profile $\mathscr{P}$.

8.5.1 Successive rule

The basic approach to controlling the successive rule is to build an agenda in reverse order so that each of the alternatives that are currently among the highest positions in the partial agenda may be strong enough to beat p alone but is too weak to be a majority winner against the whole set of alternatives behind it. To formalize this idea, we need the notion of majority winner with respect to a subset of alternatives.

Definition 8.4 (A'-majority winners)**.** Let $A' \subseteq A$ be a subset of alternatives. An alternative $a \in A'$ from the set A' is an A'*-majority winner* if there is a majority of voters who each prefer a to every alternative from $A' \setminus \{a\}$.

Next, we derive necessary conditions for an alternative to be a successive winner.

Observation 8.2. *Let $\mathscr{L}$ be an agenda for the alternatives A, let $A' \subset A$ be a proper subset of alternatives, and let $b \in A \setminus A'$ be an arbitrary alternative. If b is considered before all alternatives in A' by $\mathscr{L}$ and if b is not an $(A' \cup \{b\})$-majority winner, then b cannot be a successive winner.*

Proof. By the definition of the successive procedure, if b would be a successive winner for agenda $\mathscr{L}$, then it would have been an $(A' \cup \{b\})$-majority winner since it is considered before all alternatives in A' by the agenda—a contradiction. □

We can generalize Observation 8.2 to hold for a subset of alternatives.

Lemma 8.3. *Let $A' \subset A$ be a proper subset of alternatives. If every alternative b in $A \setminus A'$ is an $(A' \cup \{b\})$-majority winner, then no alternative from A' can be a successive winner.*

Proof. Suppose for the sake of contradiction that there is an agenda $\mathscr{L}$ for which the successive winner a' would come from A'. By the definition of the successive rule, this would imply that a majority of voters prefers a' over every alternative that is considered behind a' in $\mathscr{L}$. However, by the assumption that every alternative b in $A \setminus A'$ is an $(A' \cup \{b\})$-majority winner and that $a' \in A'$, it must hold that all alternatives in $A \setminus A'$ are considered before a' in $\mathscr{L}$, that is, $\mathscr{L}\colon (A \setminus A') \rhd a'$. Let d be the last alternative in $A \setminus A'$ that is still in front of a', that is,

$$\mathscr{L} \text{ satisfies } ((A \setminus A') \setminus \{d\}) \rhd d \rhd a'.$$

Algorithm 8.1: Algorithm for solving AGENDA CONTROL under the successive rule by computing an appropriate agenda.

Input: $(\mathscr{P} = (A, V), p)$ —- an instance of AGENDA CONTROL
1 Set $A' := \{p\}$; Set $\mathscr{L} := p$
2 **while** $A' \neq A$ **do**
3   **if** $\exists$ *an alternative* $c \in A \setminus A'$ *that is* not *an* $(A' \cup \{c\})$*-majority winner* **then**
4   Set $\mathscr{L} := c \rhd \mathscr{L}$
5   Set $A' := A' \cup \{c\}$
6   **else return** "no appropriate agenda exists"
7 **return** $\mathscr{L}$

But, since a' is a successive winner, there must be a round when d is considered and d would become a successive winner (note that all alternatives behind d come from A')—a contradiction. □

By the above lemma, we can construct an agenda in reverse order by first placing our preferred alternative p in the last position and setting $A' := \{p\}$. We will extend the agenda by putting all alternatives c that are not $(A' \cup \{c\})$-majority winners right in front of A'. Then, we update the set $A' := A' \cup \{c\}$. Using this approach we can solve the AGENDA CONTROL problem for the successive rule in polynomial time.

Theorem 8.4. *For the successive rule, solving* AGENDA CONTROL *and finding an agenda for a yes-instance of* AGENDA CONTROL *can be done in* $O(n \cdot m^2)$ *time, where* n *denotes the number of voters and* m *denotes the number of alternatives.*

Proof. Given a profile $\mathscr{P}$ and an alternative p, we use Algorithm 8.1 to either decide that p cannot be a successive winner or construct an agenda for which p wins.

For the correctness of the algorithm, if the condition in Line 3 does not apply, then by Lemma 8.3, no alternative in $A' \ni p$ can win, implying that p can never win. Thus, we can safely reply with "no".

Now, suppose that the condition in Line 3 always applies. We show that if p is a successive winner of a profile restricted to the alternative set A', then it is also a successive winner of the restricted profile that additionally contains a non-$(A' \cup \{c\})$-majority winner. To this end, let c be an alternative in $A \setminus A'$ that is *not* an $(A' \cup \{c\})$-majority winner. Assume that p is a successive winner under the current agenda $\mathscr{L}$ for the profile $\mathscr{P}'$ restricted to the alternatives in A'. Then, by Observation 8.2, it follows that under every agenda that extends $c \rhd A'$, c is not a successive winner.

This means that the rule would delete c and go on with the alternatives in A'. By assumption, p is a successive winner for profile $\mathscr{P}'$ and agenda $\mathscr{L}$. Therefore, in the profile restricted to the alternatives in $A' \cup \{c\}$, the agenda $c \triangleright \mathscr{L}$ also makes p win. This completes the correctness proof.

Finally, we come to the running time analysis. First, the procedure inside the while loop (Lines 2 and 6) is executed at most m times. Second, inside the while loop, for every alternative $c \in A \setminus A'$, we check whether it is an $A' \cup \{c\}$-majority winner. This check can be done in $O(n)$ time: We maintain a list T of size n that, for each voter v, stores the highest position of alternative from A' ranked by v. We iterate over each voter v and compare the position $v(c)$ of c ranked by v and the position $T(v)$ stored by T for voter v. We count the number of times where $v(c) < T(v)$. If this number is smaller than $n/2$, then c is not an $A' \cup \{c\}$-majority winner; we add c to A' and we update the list T by changing the entry $T(v)$ to $v(c)$ if $v(c) < T(v)$. Altogether, the running time is $O(n \cdot m^2)$. □

8.5.2 Amendment rule

Controlling the amendment rule is closely related to finding a Hamiltonian cycle in a strongly connected tournament (recall that "strongly connected" means that for each two vertices u and v, there is a directed path from u to v). To see this, we first construct a *majority graph* for the given preference profile (see the corresponding definition in Section 8.4). Recall that we assume the number of voters to be odd. The majority graph has m vertices and $\binom{m}{2}$ arcs and is indeed a tournament. From the theory of directed graphs [HM66, Theorem 7], we know that every strongly connected tournament contains a Hamiltonian cycle. Now, the crucial idea is to check whether the vertex that corresponds to p belongs to a strongly connected component that has only out-going arcs. Alternative p can win under an appropriate agenda if and only if this is the case.

Observation 8.5 (Theorem 7, [HM66])**.** *Every strongly connected tournament contains a Hamiltonian cycle.*

By carefully examining the constructive proof for Observation 8.5, we can find a Hamiltonian cycle in $O(m + m^2 + m \cdot (m + m^2)) = O(m^3)$ time.

It is well-known that every directed graph can be partitioned into strongly connected components in linear time, by using depth-first search (see [Cor+09]). Since our majority graph is not only directed but also a tournament, we can easily obtain the following result.

Observation 8.6. *Every tournament can be partitioned into disjoint strongly connected components such that the following holds.*

1. *Each component is a subtournament.*
2. *All arcs from the vertices of a component to another have only one direction.*
3. *The graph resulting from deleting all but one arbitrary vertex from each component is acyclic and a tournament.*

From this observation, we can derive the next theorem. Note that Miller [Mil77] already characterized the set of alternatives that can become an amendment winner under an appropriate agenda. Our theorem strengthens this result by giving a polynomial-time algorithm to construct the desired agenda.

Theorem 8.7. *For the amendment rule, solving* Agenda Control *and finding an agenda for a yes-instance of* Agenda Control *can be done in* $O(n \cdot m^2 + m^3)$ *time, where n denotes the number of voters and m denotes the number of alternatives.*

Proof. By Observation 8.6, every tournament consists of strongly connected subtournaments which can be ordered by topological sorting. Now, observe that only the alternatives corresponding to the vertices from the top-most strongly connected subtournament can become an amendment winner. In other words, if the vertex corresponding to the preferred alternative p does not belong to the top-most subtournament, then p can never win. As already mentioned, by carefully examining the constructive proof for Observation 8.5, we can find a Hamiltonian cycle for a strongly connected tournament with m vertices in $O(m^3)$ time. Now, we construct a sequence L_{ver} of vertices by reversing the orientation of the Hamiltonian cycle, starting with the predecessor of the vertex u_p corresponding to p and ending at the vertex u_p, and let L_{alt} be the order of the alternatives corresponding to L_{ver}. We can verify that p is an amendment winner for every agenda that extends order L_{alt}.

Thus, the problem is reduced to finding strongly connected subtournaments of the majority graph for the given preference profile: If the vertex corresponding to p is in the top-most subtournament, then construct an arbitrary but fixed order that extends L_{alt} and we answer "yes" by returning L_{alt}. Otherwise we answer "no".

Now, we come to the running time. Constructing a majority graph for a profile takes $O(n \cdot m^2)$ time.; note that the majority graph will have $O(m)$ vertices and $O(m^2)$ arcs. Partitioning the majority graph into strongly connected components takes $O(m + m^2)$ time and checking whether the vertex corresponding to p belongs to the top-most component takes $O(m)$ time. Finally, finding a Hamiltonian cycle in a strongly connected tournament takes $O(m^3)$ time. Thus, in total, our for solving AGENDA CONTROL for the amendment rule takes $O(n \cdot m^2 + m^3)$ time. □

We close this section with two remarks. First, the algorithm for the successive rule actually works for both odd and even numbers of voters. Second, our algorithm for the amendment rule can be extended to the case where the number of voters is even. There, alternative p is a winner if and only if no strongly connected component "dominates" (in a topologically ordered way) the strongly connected component that contains vertex u_p; if two alternatives are tied in a head-to-head contest, then we break ties in favor of the alternative that is considered later.

8.6 Manipulation

In this section, we consider the question of how difficult it is for voters to vote strategically to ensure a given outcome, supposing that the other voters vote sincerely. Specifically, we focus on the case where the desired outcome is to make a given alternative win. In social choice theory, however, it is also interesting to know whether a better outcome for the manipulators exists. We will see in this section that both variants of manipulation are polynomial-time solvable for both voting rules (Theorems 8.9 and 8.11 and Observation 8.13).

> COALITIONAL MANIPULATION
> **Input:** A profile $\mathscr{P} := (A, V)$ with linear preference orders, a preferred alternative $p \in A$, a non-negative integer $k \in \mathbb{N}$, and an agenda $\mathscr{L}$ for A.
> **Question:** Is it possible to add a set of k voters (a coalition) such that p wins under agenda $\mathscr{L}$?

We find that deciding whether a manipulation is successful is polynomial-time solvable for both the successive rule and the amendment rule. However, our approach to deciding whether the amendment rule can be successfully manipulated has a running time that is asymptotically higher than than our approach to deciding the same question for the successive rule.

First, we observe that the manipulators can basically vote in the same way.

Observation 8.8. *For both voting rules, if there is a successful (weighted) manipulation, then there is also a successful one where all voters from the coalition rank the alternatives in the same way.*

Proof. For the successive rule, if there is a successful manipulation for the preferred alternative p, then requiring all manipulators to rank p in the first position and to rank the other alternatives in an arbitrary but fixed order also makes p win.

Now, let $\mathscr{P}'$ be a manipulated profile, that is, the original profile plus the manipulators. Let $X := \{x_1, x_2, \ldots, x_s\}$ consist of all alternatives $x_i \in A$ such that there is a round index r_i where x_i is an r_i^{th}-round amendment winner in profile $\mathscr{P}'$. Assume without loss of generality that for every $2 \le i \le s$, alternative x_i beats x_{i-1}. If p is an amendment winner in profile $\mathscr{P}'$, then $p = x_s$. Furthermore, we can verify that each x_i is an r_i^{th}-round amendment winner in the original profile plus the k manipulators who all have preference order $x_s \succ x_{s-1} \succ \ldots \succ x_1 \succ \langle A \setminus X \rangle$. This implies that $x_s = p$ is an amendment winner. □

As in the proof for Observation 8.8, an optimal way of manipulating the successive rule is to let the manipulators rank their most preferred alternative p at the first position. Using this fact, we obtain a simple polynomial algorithm for manipulating the successive rule.

Theorem 8.9. *For the successive rule, solving* COALITIONAL MANIPULATION *and constructing the preference order for the manipulators in a yes-instance can be done in $O((n+k)\cdot m)$ time, where n denotes the number of voters, m denotes the number of alternatives, and k denotes the number of manipulators.*

Proof. In the proof of Observation 8.8, we can observe that if a coalition of k voters can manipulate the successive rule, then ranking alternative p in the first position and the other alternatives in an arbitrary but fixed order can also make p win. This leads to a linear-time $O((k+n)\cdot m)$ algorithm: Let the coalition all vote $p \succ \langle A \setminus \{p\}\rangle$, and check whether p wins under the successive rule. □

For the successive rule, we have seen that how the manipulators should vote does not depend on the sequence of the alternatives in the agenda. For the amendment rule, however, a successful manipulation greatly depends on the agenda. To elaborate this observation, we need the following.

Definition 8.5 (i^{th}-round manipulation winners)**.** Let $(\mathscr{P} := (A, V), p, k, \mathscr{L})$ be an instance of COALITIONAL MANIPULATION under the amendment rule. We call an alternative an *i^{th}-round manipulation winner* if adding some coalition of k additional voters to the original profile $\mathscr{P}$ makes this alternative the i^{th}-round amendment winner under agenda $\mathscr{L}$.

We give an example to illustrate the notion of an i^{th}-round manipulation winner.

Example 8.4. Consider the following profile with four alternatives a, b, c, d and three voters v_1, v_2, v_3. The preference orders of these three voters are given as follows:

$$v_1\colon\ a \succ b \succ c \succ d,\ v_2\colon\ b \succ c \succ a \succ d, \text{ and } v_3\colon\ d \succ b \succ c \succ a.$$

Let the agenda be $\mathcal{L}\colon\ a \rhd b \rhd c \rhd d$. Suppose that there are two manipulators. First of all, alternative a is the only 1^{st}-round manipulation winner. Both alternatives a and b are the 2^{nd}-round manipulation winners: The second alternative in $\mathcal{L}$ is b. If the two added manipulators have the same preference order $a \succ b \succ c \succ d$, then a beats b. If they have the preference order $b \succ a \succ c \succ d$, then b beats a.

By the definition of i^{th}-round manipulation winners, we know that an alternative that is *not* an i^{th}-round manipulation winner can never become an j^{th}-round manipulation winner with $j > i$. Further, one can efficiently check whether an i^{th}-round manipulation winner alternative can survive as a manipulation winner of a later round, as the following shows.

Lemma 8.10. *Let b be the i^{th} alternative in agenda $\mathcal{L}$, $2 \le i \le m$.*

1. *Alternative b is an i^{th}-round manipulation winner if and only if there is an $(i-1)^{th}$-round manipulation winner c such that requiring all manipulators to prefer b to c makes b beat c.*

2. *An $(i-1)^{th}$-round manipulation winner c is also an i^{th}-round manipulation winner if and only if requiring all manipulators to prefer c to b makes c beat b.*

Proof. We only show the first statement as the second one can be shown analogously. For the "only if" part, assume that b is an i^{th}-round manipulation winner and let $\mathcal{P}'$ be the profile with the additional manipulators for which b is an i^{th}-round amendment winner. We rename and enumerate all j^{th}-round amendment winners in $\mathcal{P}'$ with $j \le i-1$ as $x_1, x_2, \ldots, x_s$ so that for each $1 \le \ell \le s-1$, alternative $x_{\ell+1}$ beats x_ℓ in $\mathcal{P}'$. By definition, x_s is the $(i-1)^{\text{th}}$-round amendment winner. We can verify that b is an i^{th}-round amendment winner in the manipulated profile, where every manipulator has preference order $b \succ x_s \succ x_{s-1} \succ \ldots \succ x_1 \succ \langle A \setminus X \rangle$.

For the "if" part, let c be an $(i-1)^{\text{th}}$-round manipulation winner and consider the manipulators' preference orders $\succ'_1, \succ'_2, \ldots, \succ'_k$ that only rank the first $(i-1)$ alternatives in the agenda. If requiring every manipulator to prefer b to c makes b beat c, then at round i, alternative b will survive as the i^{th}-round amendment winner when every manipulator i has a preference order extending $\succ'_i \cup \{b \succ c\}$. This implies that b is an i^{th}-round manipulation winner. □

By Lemma 8.10, we can indeed compute in time quadratic in the input size all alternatives that each can become an amendment winner in some manipulated

Algorithm 8.2: Algorithm for COALITIONAL MANIPULATION under the amendment rule by computing all manipulation winners.

Input: $(\mathscr{P} = (A, V), k, p, \mathscr{L})$ — an instance of COALITIONAL MANIPULATION

1 Set $W_1 :=$ {the first alternative in the agenda $\mathscr{L}$}
2 Set $\succ_1^b := \emptyset$
3 **foreach** *round* $i \in \{2, 3, \ldots, m\}$ **do**
4   Set $W_i := \emptyset$
5   Set $b :=$ the i^{th} alternative in $\mathscr{L}$
6   **foreach** $(i-1)^{th}$*-round manipulation winner* $c \in W_{i-1}$ **do**
7   **if** $b \notin W_i$ **and** b *beats* c *when all manipulators prefer* b *to* c **then**
8   Add b to the set W_i
9   Construct the preference order $\succ_i^b := \{b \succ c\} \cup \succ_{i-1}^c$
10   **if** c *beats* b *when all manipulators prefer* c *to* b **then**
11   Add c to the set W_i
12   Construct the preference order $\succ_i^c := \{c \succ b\} \cup \succ_{i-1}^c$

profile with a coalition of k additional voters, and we can compute the corresponding coalition for each of these alternatives.

Theorem 8.11. *For the amendment rule, solving* COALITIONAL MANIPULATION *and constructing the preference orders for the manipulators in a yes-instance can be done in* $O(n \cdot m^2)$ *time, where* n *denotes the number of voters and* m *denotes the number of alternatives.*

Proof. Based on Lemma 8.10, we build a recursive algorithm which, for each i^{th}-round amendment winner, constructs a linear order over the first i alternatives in $\mathscr{L}$ starting with $i = 1$.

We denote by W_i the set of all i^{th}-round manipulation winners. For each i^{th}-round manipulation winner c, we denote by $\succ_i^c$ the preference orders over the first i alternatives in the agenda $\mathscr{L}$ such that c becomes an i^{th}-round amendment winner by adding k manipulators with a preference order that extends $\succ_i^c$.

The approach to computing all manipulation winners is described in Algorithm 8.2. Obviously, for the first round, the set W_1 and its corresponding preference order are computed correctly. By Lemma 8.10 (1), we know that Steps (7)-(9) are correct, and by Lemma 8.10(2), we know that Steps (10)-(12) are correct. Since each alternative from W_m is a last-round manipulation winner, we answer "yes" and return the corresponding preference order for the input instance $(\mathcal{P} = (A, V), k, p, \mathcal{L})$ if $p \in W_m$ and "no" otherwise.

As for the running time, first, we do some preprocessing: for each two distinct alternatives b, c, we check whether adding k manipulators can make b beat c; let the Boolean variable $T(b,c)$ have value one if this is the case and zero otherwise. Computing all these Boolean values runs in $O((k+n)\cdot m^2)$ time.

To compute W_i, $2 \le i \le m$, we inspect $T(b,c)$ for every alternative c in W_{i-1} and for the i^{th} alternative b in the agenda $\mathcal{L}$. Thus computing all W_i can be done in $O(m^2)$ time. The total running time is $O((n+k)\cdot m^2)$. □

In WEIGHTED MANIPULATION, the voters of the coalition also come with integer weights. However, we have that the weighted and non-weighted cases are equivalent because of Observation 8.8. Observe that if the sum of the weights is greater than the number of the voters in the original profile, then there is always a successful manipulation. Thus, we can conclude the following.

Corollary 8.12. *Solving* WEIGHTED MANIPULATION *and constructing the preference orders for the manipulators in a yes-instance can be solved in $O(n\cdot m)$ time for the successive rule and in $O(n\cdot m^2)$ time for the amendment rule, where n denotes the number of voters and m denotes the number of alternatives.*

We close this section by the remark that deciding whether a coalition of manipulators can achieve a better outcome is also polynomial-time solvable since we just need to solve COALITIONAL MANIPULATION for every possible alternative that is preferred to the original winner by the manipulators. Thus, the following holds.

Observation 8.13. *Deciding whether a coalition of k manipulators can achieve a better outcome by voting insincerely can be done in $O(n\cdot m^2)$ time for the successive rule and in $O(n\cdot m^3)$ time for the amendment rule, where n denotes the number of voters and m denotes the number of alternatives.*

8.7 Possible/Necessary Winner

We typically have partial knowledge about how the voters will vote, and about how the agenda will order the alternatives. Nevertheless, we might be interested in what may or may not be the final outcome. Does our favorite alternative stand any chance of winning? Is it inevitable that the government's alternative will win? Is there an agenda under which our alternative can win? Hence, we consider the question of which alternative possibly or necessarily wins.

POSSIBLE (resp. NECESSARY) WINNER
Input: A preference profile $\mathscr{P} := (A, V)$, a preferred alternative $p \in A$, and a partial agenda $\mathscr{B}$.
Question: Can p win in a (resp. every) completion of the profile $\mathscr{P}$ for an (resp. every) agenda which completes $\mathscr{B}$?

An upper bound for the computational complexity of both problems is easy to see: POSSIBLE WINNER (resp. NECESSARY WINNER) is contained in NP (resp. in coNP) for both the successive rule and the amendment rule, because one can determine a winner for both rules in polynomial time. Thus, in order to show NP-completeness (resp. coNP-completeness), we only need to show NP-hardness (resp. coNP-hardness).

8.7.1 Possible Winner

Our first two results imply that as soon as the voters may have incomplete preference orders, deciding who may be a possible winner is NP-hard even if the agenda is already a linear order. Nevertheless, for the parameter "number m of alternatives", we obtain fixed-parameter tractability. We make use of the integer linear programming techniques (see Section 2.6.1 for more information on this technique).

Theorem 8.14. *For the successive rule,* POSSIBLE WINNER *with the given agenda being linear is* NP*-complete.*

Proof. We show the NP-hardness by reducing from the NP-complete INDEPENDENT SET problem in polynomial time.

INDEPENDENT SET
Input: An undirected graph $G = (U, E)$ and a non-negative integer h.
Question: Is there an *independent set* of size at least h, that is, a subset of at least h vertices such that no two of them are adjacent to each other?

We will give a concrete example for the reduction right after the proof. Let $(G = (U, E), h)$ be an instance of INDEPENDENT SET, where $U = \{u_1, \ldots, u_r\}$ denotes the set of r vertices and $E = \{e_1, \ldots, e_s\}$ denotes the set of s edges. We assume that $r \geq 3$ and $2 \leq h \leq r - 1$. We construct a POSSIBLE WINNER instance $((A, V), p, \mathscr{B})$ as follows. The set A of alternatives consists of the preferred alternative p, one *dummy alternative* d, and of one *edge alternative* c_j for each edge $e_j \in E$:

$$A := \{p, d\} \cup \{c_j \mid e_j \in E\}.$$

We construct three groups of voters, where only the first group of voters has partial orders while the remaining two groups have linear orders. Let $\langle A \setminus \{p, d\} \rangle$ denote the linear order $c_1 \succ c_2 \succ \ldots \succ c_s$.

1. For each vertex $u_i \in U$, we construct a *vertex voter* v_i with a partial preference order specified by

   $$v_i\colon \ \langle I(u_i)\rangle \succ p \succ \langle A \setminus (\{p,d\} \cup I(u_i))\rangle \text{ and } d \succ \langle A \setminus (\{p,d\} \cup I(u_i))\rangle,$$

   where $I(u_i)$ denotes the set of edge alternatives corresponding to edges incident to vertex u_i.

   Briefly put, voter v_i prefers every "incident" edge alternative, p, and d to all "non-incident" edge alternatives. He also prefers the incident edge alternatives to p but thinks that d is incomparable to p and to every "incident" edge alternative.

2. We construct $h-2$ *auxiliary voters* with the same preference order.

   $$h-2 \text{ voters:} \qquad \langle A \setminus \{p,d\}\rangle \succ d \succ p.$$

3. We construct another $r-h-1$ auxiliary voters with the same preference order.

   $$r-h-1 \text{ voters:} \ \langle A \setminus \{p,d\}\rangle \succ p \succ d.$$

We have constructed a total of $2r-3$ voters. Thus, to be a majority winner, an alternative needs to be ranked at the first place by at least $r-1$ voters.

Let the agenda $\mathcal{B}$ be the linear order $c_1 \rhd c_2 \rhd \ldots \rhd c_s \rhd p \rhd d$. This completes the construction, which can clearly be computed in polynomial time.

Before we give a formal correctness proof, let us briefly sketch the idea. Our construction ensures that, in order for p to beat d, in the final round of the procedure, we have to put p (and by the construction of the vertex voters, also $I(u_i)$) in front of d in the preference orders of at least h vertex voters. However, to prevent any edge alternative c_j from being a successive winner in some earlier round of the procedure, we cannot put any edge alternative in front of d more than once. We will see that this is only possible if the vertices corresponding to the voters for which we put p in front of d form an independent set of size h.

We show that G has an independent set of size at least h if and only if p can possibly win under the successive rule in the constructed profile.

For the "only if" part, assume that G admits an independent set $U' \subseteq U$ of size at least h. We complete the partial preference orders of the vertex voters as follows.

1. For each vertex $u_i \in U'$, let voter v_i have the preference order

   $$\langle I(u_i)\rangle \succ p \succ d \succ \langle A \setminus (\{p,d\} \cup I(u_i))\rangle.$$

2. For each vertex $u_i \in U \setminus U'$, let voter v_i have the preference order

$$d \succ \langle I(u_i) \rangle \succ p \succ \langle A \setminus (\{p, d\} \cup I(u_i)) \rangle.$$

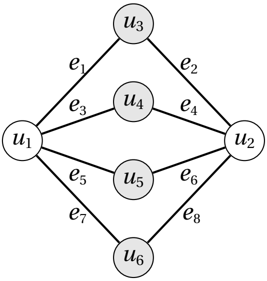
Since U' is an independent set, every edge alternative is preferred to d by at most one vertex voter. Together with the remaining $r-3$ auxiliary voters, every edge alternative is preferred to d by at most $r-2$ voters, causing each edge alternative to be deleted if it is considered prior to d in an agenda (note that we have $2r-3$ voters). Hence, by our agenda, where all edge alternatives are in front of d, all edge alternatives are deleted. In the final round, since the independent set U' has size at least h, at least h vertex voters prefer p to d. Then, p will beat d, because all $r-h-1$ auxiliary voters of the second group prefer p to d, making p a successive winner.

For the "if" part, assume that p can possibly become a successive winner, which means that we can complete the vertex voters' preference orders, ensuring p's victory. Let V' be the set of vertex voters that prefer p to d in such a completion. Since a total of $r-h-1$ auxiliary voters prefer p to d, in order to make p beat d in the last round, V' must have at least h voters (note that the majority quota is $r-1$). We show that the vertex subset U' corresponding to V' is an independent set, that is, no two vertices in U' are adjacent. Suppose for the sake of contradiction that U' contains two adjacent vertices u and u'; denote the edge $\{u, u'\}$ by e_j. In the completed profile, since p is a successive winner, there must be a round where the corresponding edge alternative c_j is considered. Since all edge alternatives with a lower index have been deleted already, by construction of the preference orders, a total of $r-1$ voters (the two vertex voters corresponding to u, u' and all $r-3$ auxiliary voters) rank c_j in the first position in the profile restricted to the alternatives $c_j, c_{j+1}, \ldots, c_s, p, d$; note that all voters prefer $c_\ell \succ c_{\ell+1}$, $1 \le \ell \le s-1$. This, however, will make c_j win—a contradiction. □

We illustrate the NP-hardness reduction for the proof of Theorem 8.14 through an example. Figure 8.1(a) depicts an undirected graph G with 6 vertices and 8 edges. We set $h := 4$. The gray vertices form an independent set of size 4. Figure 8.1(b) depicts the instance for POSSIBLE WINNER with the successive rule constructed by the reduction. The constructed profile has $6+(4-2)+(6-4-1) = 9$ voters and $2+8 = 10$ alternatives. By completing this profile according to our proof, we can verify that p is a successive winner.

Now, we show that POSSIBLE WINNER remains NP-complete for the amendment rule. As already discussed, containment in NP is obvious.

Theorem 8.15. *For the amendment rule,* POSSIBLE WINNER *with the given agenda being linear is* NP*-complete.*



(a)

voter v_1:	$\{c_1 \succ c_3 \succ c_5 \succ c_7 \succ p,\ d\} \succ c_2 \succ c_4 \succ c_6 \succ c_8$
voter v_2:	$\{c_2 \succ c_4 \succ c_6 \succ c_8 \succ p,\ d\} \succ c_1 \succ c_3 \succ c_5 \succ c_7$
voter v_3:	$\{c_1 \succ c_2 \succ p,\ d\} \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8$
voter v_4:	$\{c_3 \succ c_4 \succ p,\ d\} \succ c_1 \succ c_2 \succ c_5 \succ c_6 \succ c_7 \succ c_8$
voter v_5:	$\{c_5 \succ c_6 \succ p,\ d\} \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_7 \succ c_8$
voter v_6:	$\{c_7 \succ c_8 \succ p,\ d\} \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6$
two voters:	$c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ d \succ p$
one voter:	$c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ p \succ d$
agenda $\mathscr{B}$:	$c_1 \rhd c_2 \rhd c_3 \rhd c_4 \rhd c_5 \rhd c_6 \rhd c_7 \rhd c_8 \rhd p \rhd d$

(b)

Figure 8.1.: Illustration for the proof of Theorem 8.14. (a) An undirected graph with 6 vertices and 8 edges. The graph has an independent set of size 4 (the gray vertices) and has a vertex cover of size 2 (the white vertices). (b) The instance $((A, V), p, \mathscr{B})$ of the POSSIBLE WINNER problem for the successive rule obtained from the VERTEX COVER-instance depicted on the left with $h = 4$, where $A = \{c_1, c_2, c_3, c_4, c_5, c_6, c_7, c_8, p, d\}$, the voter set V and the corresponding preference orders and the agenda are depicted in the figure. From the constructed agenda, we notice that in order to let p beat d in the last two rounds, at least $h = 4$ vertex voters must rank $p \succ d$. But, in order not to let an edge alternative obtain too much "support", at most one of its "incident" vertex voters should rank p higher than d.

Proof. We show the NP-hardness by reducing from the NP-complete VERTEX COVER problem in polynomial time; see the beginning of Section 5.6 for the definition. We will present an example for the reduction right after the proof.

Let $(G = (U, E), h)$ be a VERTEX COVER instance, where $U = \{u_1, \ldots, u_r\}$ denotes the set of vertices and $E = \{e_1, \ldots, e_s\}$ denotes the set of edges. We assume that $r \geq 1$ and $1 \leq h \leq r - 1$.

We construct a POSSIBLE WINNER instance $((A, V), p, \mathscr{B})$ as follows. The set A of alternatives consists of the preferred alternative p, one *helper alternative* b, one *dummy alternative* d, and one *edge alternative* c_j for each edge $e_j \in E$:

$$A := \{p, b, d\} \cup \{c_j \mid e_j \in E\}.$$

We construct three groups of voters for the voter set V, where only the first group of voters has partial preference orders while the remaining two groups of voters have linear preference orders. Let $\langle A \setminus \{p, b, d\} \rangle$ denote the order $c_1 \succ c_2 \succ \ldots \succ c_s$.

1. For each vertex u_i, we construct a *vertex voter* v_i with partial order specified by

$$v_i\colon\ \left\langle A \setminus \left(\{p, b, d\} \cup I(u_i)\right) \right\rangle \succ b \succ p \text{ and } \left\langle A \setminus \left(\{p, b, d\} \cup I(u_i)\right) \right\rangle \succ \langle I(u_i) \rangle \succ d,$$

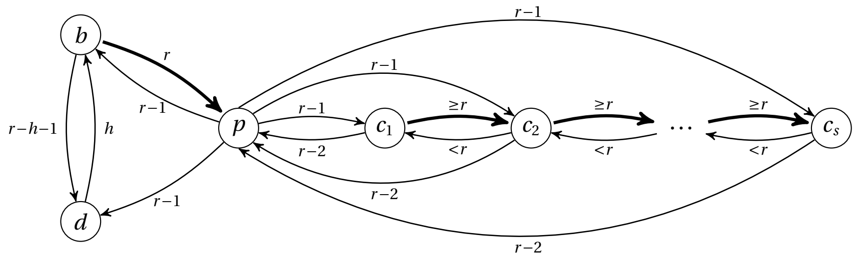


Figure 8.2.: The weighted majority graph (not including all arcs) for the obtained profile with $2r-1$ voters and $s+3$ alternatives in the reduction for POSSIBLE WINNER under the amendment rule (Theorem 8.15). The agenda is $b \rhd d \rhd p \rhd c_s \rhd c_{s-1} \rhd \ldots \rhd c_1$. We use the same symbol for both the alternative and its corresponding vertex. For instance, there is an arc from b to p with weight r because exactly r voters prefer b to p. They all come from the first group of voters. We draw an arc (a, a') as a thick line if there is already a majority of voters preferring a to a'. For the sake of brevity, some (irrelevant) arcs are omitted. By "…" we refer to the remaining edge vertices in increasing sequence.

where $\langle A \setminus (\{p, b, d\} \cup I(u_i)) \rangle$ denotes the order derived from $\langle A \setminus \{p, b, d\} \rangle$ by removing u_i's "incident" edge alternatives.

Briefly put, voter v_i prefers every "non-incident" edge alternative to the remaining ones. He prefers b to p and prefers every "incident" edge alternative to d. But, he thinks b and p are incomparable to all "incident" edge alternatives and to d.

2. We construct $r-h-1$ *auxiliary voters* with the same order

$$r-h-1 \text{ voters: } \; p \succ \langle A \setminus \{p, b, d\} \rangle \succ b \succ d.$$

3. Finally, we construct another h *auxiliary voters* with the same order

$$h \text{ voters: } \qquad p \succ d \succ \langle A \setminus \{p, b, d\} \rangle \succ b.$$

Note that we have constructed a total of $2r-1$ voters. Thus, an alternative a beats another alternative a' if and only if at least r voters prefer a to a'.

Let the agenda $\mathcal{B}$ be the linear order $b \rhd d \rhd p \rhd c_s \rhd c_{s-1} \rhd \ldots \rhd c_1$. This completes the construction, which can clearly be computed in polynomial time.

We illustrate the corresponding majority graph for the constructed profile in Figure 8.2. Now, let us briefly sketch the idea with the help of this majority graph: By the constructed agenda, in order to become an amendment winner, our preferred alternative p has to beat every edge alternative c_j, $1 \le j \le s$. By the preference orders of the auxiliary voters, for each alternative c_j, exactly $r-1$ voters prefer p to c_j (also see the weights of arcs from p to c_j in the graph). Thus, we have to put both p and, by the construction of the vertex voters, b in front of $c_j \succ d$ in the preference order of at least one vertex voter that corresponds to a vertex incident to the edge e_j. This implies that the vertices corresponding to the voters for which we put both p and b in front of d form a vertex cover. Furthermore, since b beats p (see the arc weight in the majority graph), d has to beat b in the first round of the procedure. Since all $r-h-1$ voters from the second group prefer b to d, we are only allowed to put $b \succ p$ in front of d in at most h vertex voters' preference orders. Thus, the vertex cover is of size at most h.

We show that G has a vertex cover of size at most h if and only if p can possibly win under the amendment rule.

For the "only if" part, assume that set U' is a vertex cover of size at most h. We complete the partial orders of the vertex voters as follows.

1. For each vertex $u_i \in U'$, let v_i have the preference order

$$\big\langle A \setminus \big(\{p,b,d\} \cup I(u_i)\big)\big\rangle \succ b \succ p \succ \big\langle I(u_i)\big\rangle \succ d.$$

2. For each vertex $u_i \in U \setminus U'$, let v_i have the preference order

$$\big\langle A \setminus \big(\{p,b,d\} \cup I(u_i)\big)\big\rangle \succ \big\langle I(u_i)\big\rangle \succ d \succ b \succ p.$$

Since $|U \setminus U'| \ge r-h$, at least $r-h$ vertex voters prefer d to b. Furthermore, all h voters from the third group prefer d to b, implying that a strict majority of voters prefer d to b. Thus, alternative d beats b and survives as the second round winner. Since we assume that $h \ge 1$, at least one additional vertex voter prefers p to d, and since all $r-1$ voters from the second and the third group prefer p to d, we obtain that alternative p beats d and survives as the third round winner. Since U' is a vertex cover, for each edge alternative c_j, $1 \le j \le s$, there is at least one vertex voter preferring p to c_j. This implies that p beats c_j (note that all $r-1$ auxiliary voters prefer p to c_j), making p an amendment winner.

For the "if" part, assume that there is a completion of the constructed profile, completing the preference orders of the vertex voters, so that p is an amendment winner. Let $\mathscr{P}$ be such a completion and let V' be the set of all vertex voters who prefer p to some edge alternative c_j. As we already noticed, since p is in front of

voter v_1:	$c_2 \succ c_4 \succ c_6 \succ c_8 \succ \{c_1 \succ c_3 \succ c_5 \succ c_7 \succ d,\ b \succ p\}$
voter v_2:	$c_1 \succ c_3 \succ c_5 \succ c_7 \succ \{c_2 \succ c_4 \succ c_6 \succ c_8 \succ d,\ b \succ p\}$
voter v_3:	$c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ \{c_1 \succ c_2 \succ d,\ b \succ p\}$
voter v_4:	$c_1 \succ c_2 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ \{c_3 \succ c_4 \succ d,\ b \succ p\}$
voter v_5:	$c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_7 \succ c_8 \succ \{c_5 \succ c_6 \succ d,\ b \succ p\}$
voter v_6:	$c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ \{c_7 \succ c_8 \succ d,\ b \succ p\}$
three voters:	$p \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ b \succ d$
two voters:	$p \succ d \succ c_1 \succ c_2 \succ c_3 \succ c_4 \succ c_5 \succ c_6 \succ c_7 \succ c_8 \succ b$
agenda $\mathcal{B}$:	$b \rhd d \rhd p \rhd c_8 \rhd c_7 \rhd c_6 \rhd c_5 \rhd c_4 \rhd c_3 \rhd c_2 \rhd c_1$

Table 8.2.: The instance $((A,V),p,\mathcal{B})$ of POSSIBLE WINNER for the amendment rule where $V = \{v_1, v_2, \ldots, v_6\}$ and $A = \{c_1, c_2, \ldots, c_8, p, d\}$, obtained from the graph in Figure 8.1(a) and $h = 2$, the voter set V and the corresponding preference orders and the agenda are depicted above. By the construction of the agenda, we notice that in order to let p become an amendment winner, at most $h = 2$ vertex voters can rank $b \succ p \succ c_j \succ d$ for some c_j. In order to beat every edge alternative c_j, at least one "incident" voter must rank $b \succ p \succ c_j$.

every edge alternative in the agenda, p must beat every edge alternative. Thus, for each edge alternative c_j, there is at least one vertex voter $v_i \in V'$ who prefers p to c_j. By the construction of the preference orders, this means that V' corresponds to a vertex cover.

Every vertex voter in V' also prefers b to d (because he originally ranks $b \succ p$ and $c_j \succ d$). Since all $r - h - 1$ auxiliary voters from the second group prefer b to d and since p can only possibly win if b does not survive the second round (because b beats p), there are at most h voters in V'. This further implies that the vertex set corresponding to V' is a vertex cover of size at most h. □

We illustrate the NP-hardness reduction through an example. Let us consider the undirected graph depicted in Figure 8.1(a) again. We set $h := 2$, and the vertex cover consists of the white vertices. Then, the constructed instance of POSSIBLE WINNER for the amendment rule has $6 + (6 - 2 - 1) + 2 = 11$ voters and $3 + 8 = 11$ alternatives. This instance can be found in Table 8.2.

We have just shown that it is NP-hard to decide whether a given alternative can *possibly* be a successive (or amendment) winner, even when the given agenda is already a linear order (see Theorems 8.14 and 8.15). In both NP-hardness reductions, the number m of alternatives and the number n of voters are unbounded. In most parliamentary elections, however, there is only a limited number of alternatives. For this case we obtain tractability results. Specifically, we show that with respect to the parameter m, POSSIBLE WINNER is *fixed-parameter tractable* (FPT, see Section 2.6 for more information on this complexity class).

Our approach is based on the integer linear programming technique. We reduce POSSIBLE WINNER to INTEGER LINEAR PROGRAMMING FEASIBILITY (Definition 2.8) where the input matrix has $2^{O(m^2)}$ columns (that is, variables) and $2^{O(m^2)}$ rows (that is, constraints), and the absolute value of each coefficient and of each entry in the goal vector is at most $2n$ (recall that m denotes the number of alternatives and n the number of voters). By Theorem 2.2, this immediately implies fixed-parameter tractability for the parameter m.

Theorem 8.16. *Let m denote the number of alternatives and n the number of voters of a given* POSSIBLE WINNER *instance. Let* $\mathsf{ilp}(\rho_1, \rho_2, \rho_3)$ *denote the running time of* INTEGER LINEAR PROGRAMMING FEASIBILITY *which has ρ_1 variables and ρ_2 constraints, and the maximum of the absolute values of the coefficients and the constant terms is ρ_3 with $\rho_1, \rho_2, \rho_3 \in \mathbb{N}$. Then,* POSSIBLE WINNER *can be solved in* $O(m! \cdot \mathsf{ilp}(m! \cdot 2^{m^2}, 2n + m, n))$ *time for the successive rule and in* $O(m! \cdot 2^m \cdot \mathsf{ilp}(m! \cdot 2^{m^2}, 2n + 3m, 2n))$ *time for the amendment rule.*

Proof. The general idea for both rules is to "guess" the possible completion of the agenda and the sequential outcome of the voting regarding this agenda for which our preferred p may win and use integer linear programs (ILPs) to check whether such a guess is valid.

We introduce some notation for the description of our ILPs for both rules. Let Π denote the set of partial preference orders of the voters in the given profile. For each partial order $\succ \in \Pi$, we write $N(\succ)$ to denote the number of voters with partial order $\succ$ in the original profile, and we write $C(\succ)$ to denote the set of all possible linear orders completing $\succ$. For instance, if we have three alternatives a, b, c, then $C(\{a, b\} \succ c) = \{a \succ b \succ c, b \succ a \succ c\}$ and $C(a \succ b \succ c) = \{a \succ b \succ c\}$. Accordingly, for each linear order $\succ^*$, we write $C^{-1}(\succ^*)$ to denote the set of all partial orders that $\succ^*$ can complete.

For each partial order $\succ$ and each possible completion $\succ^* \in C(\succ)$ of $\succ$, we introduce an integer variable $x(\succ, \succ^*)$ denoting the number of voters with partial order $\succ$ in the input profile that will have linear order $\succ^*$ in a possible completing profile. Since, for a set of m elements, there are at most 2^{m^2} possible partial orders and at most $m!$ possible linear orders, we have introduced at most $m! \cdot 2^{m^2}$ variables. We use these variables for both the successive and the amendment rules.

Successive rule. Suppose we guess that our preferred alternative p will possibly win under agenda $\mathcal{L}$: $a_1 \rhd a_2 \rhd \ldots \rhd a_y \rhd \ldots \rhd a_m$ with $a_y = p$. We need one more notation: For each alternative a_i, $1 \le i \le m-1$, let $F(a_i)$ denote the set of linear orders $\succ$ with $a_i \succ \{a_{i+1}, a_{i+2}, \ldots, a_m\}$. Now, the crucial point is that our preferred alternative p is a successive winner under agenda $\mathcal{L}$ if and only if the following two conditions hold.

1. The profile can be completed so that *no* alternative a_i, $1 \le i \le y-1$, is an $\{a_i, a_{i+1}, \ldots, a_m\}$-majority winner.
2. Alternative p is an $\{a_y, a_{y+1}, \ldots, a_m\}$-majority winner.

We can describe these two conditions via a formulation of the ILP feasibility problem.

$$\sum_{\substack{\succ^* \in F(p) \\ \succ \in C^{-1}(\succ^*)}} x(\succ, \succ^*) > \frac{n}{2}, \tag{8.1}$$

$$\sum_{\succ^* \in C(\succ)} x(\succ, \succ^*) = N(\succ), \qquad \forall \text{ partial orders } \succ \in \Pi, \tag{8.2}$$

$$\sum_{\substack{\succ^* \in F(a_i) \\ \succ \in C^{-1}(\succ^*)}} x(\succ, \succ^*) \le \frac{n}{2}, \qquad \forall 1 \le i \le y-1, \tag{8.3}$$

$$x(\succ, \succ^*) \ge 0, \qquad \forall \text{ partial orders } \succ \in \Pi. \tag{8.4}$$

We explain the meaning of these three groups of constraints. First, constraint set (8.1) ensures that our preferred alternative p is an $\{a_y, a_{y+1}, \ldots, a_m\}$-majority winner (condition 2) and constraint set (8.3) ensures that *no* alternative a_j, $1 \le j \le y-1$, is an $\{a_j, a_{j+1}, \ldots, a_m\}$-majority winner (condition 1). Second, constraint set (8.4) ensures that our variables are non-negative. Finally, constraint set (8.2) ensures that each voter's partial order is completed to exactly one linear order. The correctness of our formulation thus follows.

Before we come to the running time, let us mention again that we need to change the equations (8.2) and the strict inequality (8.1) to obtain an instance of INTEGER LINEAR PROGRAMMING FEASIBILITY. This is quite straightforward and the numbers of additional variables and additional constraints are upper-bounded by the number of original constraints, respectively.

We derive our desired running time from the time of running an ILP. First of all, we brute-force search into all $m!$ possible completions of the input agenda. Then, for each of these completions, we run an ILP with $O(m! \cdot 2^{m^2})$ variables $x(\succ, \succ^*)$ and $O(2n+m)$ constraints where the absolute value of each coefficient and the constant term is at most n. For a remark on a transformation from our ILP formulation to an ILP instance, we refer the reader to the end of the case of the successive rule in this proof or to Section 2.6.1.

Amendment rule. Just as for the successive rule, we guess a completion of the agenda. Let $\mathcal{L} := a_1 \triangleright a_2 \triangleright \ldots \triangleright a_y \triangleright \ldots \triangleright a_m$ be the guessed agenda with a_y being our preferred alternative p. Apart from this, we also guess the amendment winner b_i of each round i. We do this by guessing whether a_i will be the i^{th}-round amendment winner: Let $b_1 := a_1$ because by definition, the first round amendment winner is a_1. For each $2 \le i \le y-1$, we guess a Boolean value $c_i \in \{0,1\}$ and let $b_i := a_i$ if $c_i = 1$ and let $b_i := b_{i-1}$ otherwise. Note that guessing the amendment winner of a round numbered higher than or equal to y is not necessary since the goal includes making p beat every alternative a_z with $z > y$.

Now, we can use an ILP formulation to check whether a valid guess can be realized by completing our partial profile. To this end, for each two distinct alternatives a and b, let $G(a,b)$ denote the set of all linear orders with $a \succ b$. Our ILP formulation consists of four groups of constraints:

$$\sum_{\succ^* \in C(\succ)} x(\succ,\succ^*) = N(\succ), \quad \forall \text{ partial orders } \succ \in \Pi, \tag{8.5}$$

$$\sum_{\substack{\succ^* \in G(b_{i-1},a_i)\\ \succ \in C^{-1}(\succ^*)}} x(\succ,\succ^*) - \sum_{\substack{\succ^* \in G(a_i,b_{i-1})\\ \succ \in C^{-1}(\succ^*)}} x(\succ,\succ^*) < (1-c_i)\cdot(n+1), \quad \forall 2 \le i \le y-1, \tag{8.6}$$

$$\sum_{\substack{\succ^* \in G(b_{i-1},a_i)\\ \succ \in C^{-1}(\succ^*)}} x(\succ,\succ^*) - \sum_{\substack{\succ^* \in G(a_i,b_{i-1})\\ \succ \in C^{-1}(\succ^*)}} x(\succ,\succ^*) > -c_i\cdot(n+1), \quad \forall 2 \le i \le y-1, \tag{8.7}$$

$$\sum_{\substack{\succ^* \in G(p,a_i)\\ \succ \in C^{-1}(\succ^*)}} x(\succ,\succ^*) - \sum_{\substack{\succ^* \in G(a_i,p)\\ \succ \in C^{-1}(\succ^*)}} x(\succ,\succ^*) > 0, \quad \forall y+1 \le i \le m, \tag{8.8}$$

$$x(\succ,\succ^*) \ge 0, \quad \forall \text{ partial orders } \succ \in \Pi. \tag{8.9}$$

The meaning of these four groups of constraints is as follows: The first group (constraint (8.5)) ensures that each voter's partial order is completed to exactly one linear order. The second and the third groups of constraints make the guessed alternative b_i an i^{th}-round amendment winner: If $c_i = 1$, then the right-hand side of constraint (8.6) is zero. Hence, satisfying this constraint makes a_i beat b_{i-1} (because the number of voters preferring a_i to b_{i-1} is greater than the number of voters preferring b_{i-1} to a_i). Thus, $b_i = a_i$. Otherwise, c_i equals 0, which implies that the right-hand side of constraint (8.7) is zero. Hence, satisfying this constraint makes b_{i-1} beat a_i (because the number of voters preferring b_{i-1} to a_i is greater than the number of voters preferring a_i to b_{i-1}). Consequently, the i^{th}-round amendment winner b_i is b_{i-1}. The fourth group (constraint (8.8)) ensures that for each $i \in \{y+1, y+2, \ldots, m\}$, the number of voters preferring p to a_i is greater than the number of voters preferring a_i to p, ensuring that p beats a_i. The last group (constraint (8.9)) ensures that our variables are non-negative. The correctness of our ILP formulation thus follows.

To analyze the running time, first of all, for each valid completion of the agenda and valid sequence of the amendment winners (there are $m! \cdot 2^m$ many), we run our ILP which has $O(m!\cdot 2^{m^2})$ variables $x(\succ, \succ^*)$ and $O(2n+3m)$ constraints, and where the absolute value of each coefficient and each constant term is at most $2n$. For a remark on a transformation from our ILP formulation to an ILP instance, we refer the reader to the end of the case of the successive rule in this proof or to Section 2.6.1. □

By Theorem 2.2, we can derive the following tractability result because an integer linear program with ρ_1 variables and ρ_2 constraints, and whose coefficients and constant terms are between $-\rho_3$ and ρ_3, can be encoded in $O(\rho_1 \cdot \rho_2 \cdot \log(\rho_3 + 2))$ bits (see Footnote 2 and Theorem 2.2).

Corollary 8.17. *Let m denote the number of alternatives and n denote the number of voters of a given* Possible Winner *instance. Then, for both voting rules,* Possible Winner *can be solved in $O(\rho^{2.5\rho+o(\rho)+2} \cdot \log(n+2))$ time, where $\rho = m! \cdot 2^{m^2}$.*

We close this section by remarking that the complexity result in Corollary 8.17 is of classification nature only. It would be interesting to know whether our fixed-parameter tractability results achieved through integer linear programming can also be achieved by a direct combinatorial (fixed-parameter) algorithm (also see the more general discussion of Bredereck et al. [Bre+14a, Key question 1]).

8.7.2 Necessary Winner

In notable contrast to Possible Winner, the Necessary Winner problem for the successive rule tends to be computationally easier than for the amendment rule. Observe that a given alternative p is *not* a necessary winner if and only if the given profile and partial agenda can be completed so that there is another alternative $c \neq p$ such that one the following holds:

1. c beats p and it is considered after p in the completed agenda or

2. c is already a majority winner when considered and it is considered before p.

The reason for the difference in complexity between the two voting rules seems to be that for the successive rule, checking whether there is a completion of the profile and the agenda satisfying one of the above two conditions can be done in polynomial time, but it is not clear how to check this in polynomial time for the amendment rule. Throughout the rest of this section, we assume that a NECESSARY WINNER instance $(\mathscr{P} = (A, V), p, \mathscr{B})$ is given.

Successive rule. We start with the successive rule and show that deciding whether our preferred alternative is not necessarily a successive winner can be solved in polynomial time. We introduce some notion regarding the alternatives in a (possibly not linear) agenda.

Definition 8.6 ($\mathscr{B}_c^{\Leftarrow}$, $\mathscr{B}_c^{\sim}$, and $\mathscr{B}_c^{\Rightarrow}$)**.** Given a partial agenda $\mathscr{B}$ (which is possibly not linear), for each alternative $c \in A$, let $\mathscr{B}_c^{\Leftarrow}$ be the set of all alternatives c' that are ordered in front of c by $\mathscr{B}$ (that is, $\mathscr{B}\colon c' \rhd c$). Let $\mathscr{B}_c^{\sim}$ be the set of all alternatives c' whose relative positions to c are not specified by $\mathscr{B}$ (that is, $\mathscr{B}\colon c \sim_{\rhd} c'$), and let $\mathscr{B}_c^{\Rightarrow}$ be the set of all alternatives c' that are ordered behind c by $\mathscr{B}$ (that is, $\mathscr{B}\colon c \rhd c'$).

Note that for each alternative $c \in A$, the three sets $\mathscr{B}_c^{\Leftarrow}$, $\mathscr{B}_c^{\sim}$, and $\mathscr{B}_c^{\Rightarrow}$ are pairwise disjoint and that $\mathscr{B}_c^{\Leftarrow} \dot\cup \mathscr{B}_c^{\sim} \dot\cup \mathscr{B}_c^{\Rightarrow} = A \setminus \{c\}$.

We derive the main idea behind our polynomial-time algorithm from the following simple observation.

Observation 8.18. *Alternative p is not a necessary successive winner if and only if there is a completion $(\mathscr{P}^*, \mathscr{L})$ of profile $\mathscr{P}$ and agenda $\mathscr{B}$ such that some other alternative may win, that is, such that*

1. *p is not an $(\mathscr{L}_p^{\Rightarrow} \cup \{p\})$-majority winner, or*
2. *$\mathscr{L}_p^{\Leftarrow}$ contains an alternative c that is an $(\mathscr{L}_c^{\Rightarrow} \cup \{c\})$-majority winner.*

We will show that checking whether there is a completion satisfying one of the above conditions can be done in polynomial time. First, we need two concepts regarding the completions of a profile and we need to rephrase the above conditions.

Definition 8.7 (c-discriminating and c-privileging)**.** Let c be an arbitrary alternative. Consider a specific linear preference order $\succ^*$ that completes a partial preference order $\succ$ such that for each incomparable pair X of alternatives in $\succ$ with $c \notin X$, the relative order of this pair X in $\succ^*$ is determined by an arbitrary but fixed linear order. We say that this specific preference order $\succ^*$ is *c-discriminating* if for each alternative c' that is incomparable to c in $\succ$, it holds that $c' \succ^* c$. Similarly, $\succ^*$ is *c-privileging* if for each alternative c' that is incomparable to c in $\succ$, it holds that $c \succ^* c'$.

Now, consider a profile $\mathcal{P}^*$ that completes $\mathcal{P}$. We say that $\mathcal{P}^*$ is *c-discriminating* if the preference order of each voter in $\mathcal{P}^*$ is c-discriminating and that $\mathcal{P}^*$ is *c-privileging* if the preference order of each voter in $\mathcal{P}^*$ is c-privileging.

Note that a c-discriminating (resp. c-privileging) profile is unique. We give an example to illustrate these two concepts.

Example 8.5. Let $\mathcal{P}$ be a preference profile with four alternatives a, b, c, d, and two voters v_1, v_2 whose partial preference orders are specified as

$$v_1\colon\ b \succ c \succ d \ \text{ and } \ v_2\colon d \succ b.$$

Consider the fixed order $a \succ b \succ c \succ d$ over the alternatives.

The profile with the following preference orders

$$v_1\colon\ a \succ b \succ c \succ d \ \text{ and } \ v_2\colon a \succ d \succ b \succ c\ ,$$

is c-discriminating; note that it is unique with respect to the fixed order.

The profile with the following preference orders

$$v_1\colon\ b \succ c \succ a \succ d \ \text{ and } \ v_2\colon c \succ a \succ d \succ b,$$

is c-privileging; note that it is unique with respect to the fixed order.

Now, we can rephrase conditions 1 and 2 from Observation 8.18 so that we can verify them in polynomial time.

Lemma 8.19. *There is a completion $(\mathcal{P}^*, \mathcal{L})$ of $(\mathcal{P}, \mathcal{B})$ satisfying condition 1 from Observation 8.18 if and only if p is not an $(A \setminus \mathcal{B}_p^{\Leftarrow})$-majority winner in the p-discriminating completion of $\mathcal{P}$.*

Proof. For the "only if" case, suppose that $(\mathcal{P}^*, \mathcal{L})$ completes $(\mathcal{P}, \mathcal{B})$ and satisfies condition 1. This means that at least half of the voters in $\mathcal{P}^*$ prefer some alternative $c \in \mathcal{L}_p^{\Rightarrow}$ to p. Since $\mathcal{L}$ completes $\mathcal{B}$, it follows that $\mathcal{L}_p^{\Rightarrow} \subseteq A \setminus (\mathcal{B}_p^{\Leftarrow} \cup \{p\})$. Let $\mathcal{P}^{**}$ be the p-discriminating profile. Since each voter in $\mathcal{P}^{**}$ who prefers some alternative $c \in \mathcal{L}_p^{\Rightarrow}$ to p will certainly still prefer c to p in $\mathcal{P}^{**}$ (because this profile "discriminates" p), it must hold that at least half of the voters in $\mathcal{P}^{**}$ prefer some alternative $c \in \mathcal{L}_p^{\Rightarrow} \subseteq A \setminus (\mathcal{B}_p^{\Leftarrow} \cup \{p\})$ to p. Thus, p is not an $(A \setminus \mathcal{B}_p^{\Rightarrow})$-majority winner in $\mathcal{P}^{**}$.

For the "if" case, suppose that p is not an $(A \setminus \mathcal{B}_p^{\Leftarrow})$-majority winner in the p-discriminating profile $\mathcal{P}^{**}$. Consider an arbitrary agenda $\mathcal{L}$ that satisfies

$$\mathcal{B}_p^{\Leftarrow} \rhd p \rhd (\mathcal{B}_p^{\sim} \cup \mathcal{B}_p^{\Rightarrow}).$$

One can verify that $(\mathcal{P}^{**}, \mathcal{L})$ satisfies condition 1 because p is not a $(\mathcal{B}_p^{\sim} \cup \mathcal{B}_p^{\Rightarrow} \cup \{p\})$-majority winner. $\square$

Lemma 8.20. *Assume that no completion of $(\mathcal{P}, \mathcal{B})$ satisfies condition 1. Then, there is a completion $(\mathcal{P}^*, \mathcal{L})$ of $(\mathcal{P}, \mathcal{B})$ satisfying condition 2 from Observation 8.18 if and only if there is some alternative $b \in \mathcal{B}_p^{\Leftarrow}$ being a $(\mathcal{B}_b^{\Rightarrow} \cup \{b\})$-majority winner in the b-privileging completion of $\mathcal{P}$.*

Proof. For the "only if" case, suppose that $(\mathcal{P}^*, \mathcal{L})$ is a completion of $(\mathcal{P}, \mathcal{B})$ which satisfies condition 2 and let $c \in \mathcal{L}_p^{\Leftarrow}$ be such an $(\mathcal{L}_c^{\Rightarrow} \cup \{c\})$-majority winner. This implies that c beats p in $\mathcal{P}^*$ because $p \in \mathcal{L}_c^{\Rightarrow}$. Observe that $c \in \mathcal{B}_p^{\Leftarrow} \cup \mathcal{B}_p^{\sim}$ holds because $\mathcal{L}$ completes $\mathcal{B}$. If $c \in \mathcal{B}_p^{\sim}$ holds, that is, if $\mathcal{B}$ does not specify the relative order of c and p, then the same completing profile $\mathcal{P}^*$ and a completing agenda of $\mathcal{B}$ with $p \rhd c$ satisfy condition 1 which is not possible by our assumption. Thus, we have that $c \in \mathcal{B}_p^{\Leftarrow}$.

Now, we show that c is a $(\mathcal{B}_c^{\Rightarrow} \cup \{c\})$-majority winner in the c-privileging completion $\mathcal{P}^{**}$ of $\mathcal{P}$. Observe that each voter v in $\mathcal{P}^*$ who prefers c to $\mathcal{L}_c^{\Rightarrow}$ must also prefer c to $\mathcal{B}_c^{\Rightarrow}$ because $\mathcal{L}$ completes $\mathcal{B}$. Together with the assumption that c is an $(\mathcal{L}_c^{\Rightarrow} \cup \{c\})$-majority winner in $\mathcal{P}^*$, this implies that more than half of the voters in $\mathcal{P}^*$ prefer c to $\mathcal{B}_c^{\Rightarrow}$. Since each voter in $\mathcal{P}^*$ who prefers c to $\mathcal{B}_c^{\Rightarrow}$ will still prefer c to $\mathcal{B}_c^{\Rightarrow}$ in the c-privileging profile $\mathcal{P}^{**}$, more than half of the voters in $\mathcal{P}^{**}$ prefer c to $\mathcal{B}_c^{\Rightarrow}$.

For the "if" case, suppose that $\mathcal{B}_p^{\Leftarrow}$ contains an alternative b that is a $\mathcal{B}_b^{\Rightarrow} \cup \{b\}$-majority winner in the b-privileging completion $\mathcal{P}^{**}$ of $\mathcal{P}$. Now, consider a completing agenda $\mathcal{L}$ that satisfies

$$\mathcal{B}_b^{\Leftarrow} \rhd \mathcal{B}_b^{\sim} \rhd c \rhd \mathcal{B}_b^{\Rightarrow}.$$

We can easily verify that $(\mathcal{P}^{**}, \mathcal{L})$ satisfies condition 2 (with respect to b). $\square$

Now, we have all ingredients to show that deciding on a necessary successive winner is polynomial-time solvable.

Theorem 8.21. *For the successive rule,* NECESSARY WINNER *can be solved in $O(n \cdot m^3)$ time.*

Algorithm 8.3: Algorithm for solving Necessary Winner by checking whether p is a necessary winner.

```
Input: (𝒫 = (A, V), p, ℬ) — an instance of Necessary Winner
1 Compute the p-discriminating completion 𝒫* of 𝒫
2 if p is not an (A \ ℬ_p^⇐)-majority winner in 𝒫* then
3     return "no"
4 foreach alternative c ∈ ℬ_p^⇐ do
5     Compute the c-privileging completion 𝒫** of 𝒫
6     if c is a (ℬ_c^⇒ ∪ {c})-majority winner in 𝒫** then
7         return "no"
8 return "yes"
```

Proof. By Observation 8.18 and by Lemmas 8.19 and 8.20, we can conclude that p is not a necessary winner if and only if

1. p is not an $(A \setminus \mathscr{B}_p^{\Leftarrow})$-majority winner in the p-discriminating completion of $\mathscr{P}$ or

2. $\mathscr{B}_p^{\Leftarrow}$ contains an alternative c which is a $(\mathscr{B}_c^{\Rightarrow} \cup \{c\})$-majority winner in the c-privileging completion of $\mathscr{P}$.

With regard to the second requirement, we remark that a $(\mathscr{B}_c^{\Rightarrow} \cup \{c\})$-majority winner is not guaranteed to win under the successive procedure because some other alternative c' from $\mathscr{B}_c^{\Leftarrow}$ is already a $(\mathscr{B}_{c'}^{\Rightarrow} \cup \{c'\})$-majority winner. Nevertheless, p will not be a successive winner in this case.

Algorithm 8.3 checks whether one of the two requirements is fulfilled. Fortunately, this can be done in polynomial time: Computing the p-discriminating or the c-privileging completion for some alternative $c \in A \setminus \{p\}$ takes $O(n \cdot m^2)$ time and finding the majority winner also takes $O(n \cdot m^2)$ time. The algorithm iterates at most m times through the loop in Steps (4)–(7). Altogether it takes $O(n \cdot m^3)$ time. □

Amendment rule. Adapting the Vertex Cover reduction from the proof of Theorem 8.15, we can show that Necessary Winner for the amendment rule is coNP-hard.

Theorem 8.22. *For the amendment rule,* Necessary Winner *with the given agenda being linear is* coNP*-complete.*

Proof. Recall that in the proof of Theorem 8.15 we constructed a profile $\mathcal{P} = (A, V)$ with $2r - 1$ voters and a linear agenda $\mathcal{B}$ for a given instance $(G = (U, E), h)$ of VERTEX COVER, and we showed that G admits no vertex cover of size at most h if and only if our preferred alternative p is not a possible amendment winner. Note that p is not a possible winner if and only if in some completion of the profile $\mathcal{P}$,

1. the helper alternative b beats the dummy alternative d or
2. at least one edge alternative c_ℓ beats p.

Since all $(r-1)$ auxiliary voters and at least one vertex voter prefer each edge alternative to alternative b, each edge alternative beats b. This implies that if b beats d in the second round, then c_s beats b, and b will be deleted in the fourth round (note that b beats p in all cases). Thus, p is not a possible winner if and only if there is a completion of the profile, where some edge alternative c_ℓ, $1 \le \ell \le s$, becomes an $(s - \ell + 4)^{\text{th}}$-round amendment winner. Since every voter has the same preference order $c_1 \succ c_2 \succ \ldots \succ c_s$ over all edge alternatives, edge alternative c_1 beats every remaining edge alternative c_j, $j > 1$, and becomes the amendment winner. This implies that c_1 necessarily wins if and only if p does not possibly win. Hence, the construction in the proof of Theorem 8.15 provides a polynomial-time reduction from the coNP-complete CO-VERTEX COVER problem to our NECESSARY WINNER problem for the amendment rule. □

Using the ILP formulation for POSSIBLE WINNER under the amendment rule (Theorem 8.16), we can check whether there is a possible winner different from p. Since p is a necessary winner if and only if there is no other possible winner, using the results of Lenstra [Len83], Kannan [Kan87], and Frank and Tardos [FT87] we can conclude the following.

Corollary 8.23. *Let m denote the number of alternatives and n denote the number of voters of a given* NECESSARY WINNER *instance. Then, for the amendment rule,* NECESSARY WINNER *can be solved in $O(\rho^{2.5\rho+o(\rho)+2} \cdot \log(n+2))$ time, where $\rho = m! \cdot 2^{m^2}$.*

8.7.3 The case of weighted voters

If each voter comes with a weight, then for the amendment rule, POSSIBLE WINNER and NECESSARY WINNER are weakly NP-hard when the number of alternatives is three and four, respectively [Lan+12, Pin+11]. For the successive rule, we also obtain NP-hardness for WEIGHTED POSSIBLE WINNER.

Theorem 8.24. *For the successive rule,* WEIGHTED POSSIBLE WINNER *is weakly* NP-*hard even for three alternatives and when the agenda $\mathcal{B}$ is linear.*

Proof. We show NP-hardness by providing a polynomial-time reduction from the weakly NP-complete PARTITION problem [GJ79].

> PARTITION
> **Input:** A multi-set $X = (x_1, x_2, \ldots, x_r)$ of positive integers.
> **Question:** Is there a *perfect partition* $X_1 \dot\cup X_2 = X$ of the integers such that both parts sum up to the same value, that is, $\sum_{x \in X_1} x = \sum_{x \in X_2} x$?

Let $X = (x_1, x_2, \ldots, x_r)$ be a PARTITION instance with $\sum_{x \in X} x = 2K$. We construct a WEIGHTED POSSIBLE WINNER instance $((A, V), p, \mathcal{B}, \omega\colon V \to \mathbb{N})$ with ω being the weight function. The set A of alternatives consists of the preferred alternative p and two further alternatives a and b:

$$A := \{p, a, b\}.$$

The set V of voters consists of r *number voters* and one *dummy voter*:

1. For each integer $x_i \in X$ (which is positive), we construct one *number voter* v_i with a partial order specified by $a \succ p$ and with weight $\omega(v_i) = x_i$.

2. We construct one additional *dummy voter* d with linear order $p \succ b \succ a$ and with weight $\omega(d) = 1$.

Finally, the partial agenda $\mathcal{B}$ is linear and set to $a \rhd b \rhd p$. This completes the construction, which clearly can be computed in polynomial time.

Before going into the details of the proof, observe that the total weight of all voters is $2K + 1$ and, hence, the majority quota is $K + 1$.

For the "only if" part, assume that there is a perfect partition $X_1 \dot\cup X_2 = X$ of the integers such that $\sum_{x \in X_1} x = \sum_{x \in X_2} x = K$. Then, we complete the profile as follows. For each number voter v_i, if $x_i \in X_1$, then the preference order of voter v_i is completed to $a \succ p \succ b$; otherwise, the preference order of voter v_i is completed to $b \succ a \succ p$. By our agenda $\mathcal{B}$, alternative a will be deleted in the second round because it is not a majority winner (only the voters corresponding to the integers in X_1 prefer a to $\{p, b\}$). In the profile restricted to b and p, the dummy voter plus the voters that correspond to the integers in X_1 prefer p to b; the sum of their weights is $K + 1$. Thus, p beats b and becomes a winner.

For the "if" part, assume that there is a completion $\mathcal{P}^*$ of the profile where p is a successive winner. This implies that a is not a majority winner. Thus, by the preference order of the dummy voter which has weight one, the sum of the weights of the number voters that have preference order $b \succ a \succ p$ is at least K. In the third round, p must beat b, which implies that the sum of the weights of the number voters that have preference order $b \succ a \succ p$ is at most K. Summarizing, the number voters that have preference order $b \succ a \succ p$ have a total weight equal to K. The corresponding integers sum up to K. □

As already mentioned in the beginning of Section 8.7.3, Pini et al. [Pin+11] and Lang et al. [Lan+12] showed that WEIGHTED NECESSARY WINNER is weakly NP-hard for the amendment rule, even for four alternatives. We complement this result by showing that it is polynomial-time solvable for the successive rule, and it is linear-time solvable for the amendment rule when the number of alternatives is at most three.

Theorem 8.25. *Let m denote the number of alternatives, n denote the number of voters, and w denote the sum of the weights of all voters of a given* NECESSARY WINNER *instance. The following holds. 1. For the successive rule,* WEIGHTED NECESSARY WINNER *can be solved in $O(w \cdot n \cdot m^3)$ time. 2. For the amendment rule,* WEIGHTED NECESSARY WINNER *can be solved in linear time for up to three alternatives.*

Proof. First, we observe that the algorithm (Theorem 8.21: Algorithm 8.3) that we provide to check whether an alternative is a necessary successive winner of a given profile without weights can be easily adapted to also solve the case of weighted voters. Thus, for the successive rule, WEIGHTED NECESSARY WINNER is also solvable in $O(w \cdot n \cdot m^3)$ time.

We show that for up to three alternatives, WEIGHTED NECESSARY WINNER is linear-time solvable. This is closely related to computing the possible and necessary Condorcet winners. To show our result, for each two alternatives a and b, let $\omega(a, b)$ be the sum of the weights of all voters who prefer a to b.

We check all (up to) six completions $\mathcal{L}$ of a given partial agenda $\mathcal{B}$ and answer that our preferred alternative p is a necessary amendment winner only if all checks return "yes". First, if p is in one of the first two positions in agenda $\mathcal{L}$, then the problem is equivalent to asking whether p is a necessary Condorcet winner, that is, whether all completions of the preference orders make p beat the other(s), one by one. The answer to the latter is yes if and only if for each remaining alternative c, the sum of the weights of all voters who prefer p to c is more than half of the total sum of weights. This can be checked in linear time.

We are left with the very last case where the profile has three alternatives, denoted by a, b, and p, and where the preferred alternative p is in the last position in the agenda $\mathcal{L}$. Let K be the total sum of weights. In this case, p is a necessary winner if and only if either p is a Condorcet winner or the sum of the weights of all voters who prefer p to the *necessary Condorcet winner* among a and b is greater than $K/2$:

(1) If $\omega(p,a) > K/2$ and if $\omega(p,b) > K/2$, then answer "yes";

(2) Otherwise, if $\omega(p,a) > K/2$ and if $\omega(a,b) > K/2$, then answer "yes";

(3) Otherwise, if $\omega(p,b) > K/2$ and if $\omega(b,a) > K/2$, then answer "yes";

(4) Otherwise, answer "no".

The condition stated in step (1) implies that p is a Condorcet winner. As we know, if a Condorcet winner exists, then it is the only unique amendment winner (Observation 8.1). Thus, step (1) gives a correct answer. If the conditions in step (1) does not hold, then the first inequality in step (2) implies $\omega(p,b) \leq K/2$. The second inequality in step (2) implies that in every completion of the profile, a will beat b. In this case, p will beat a and becomes an amendment winner. Thus, step (2) gives a correct answer. Analogously, we can show that step (3) also gives a correct answer.

If neither of the first three steps applies, then we obtain that one of the conditions between $\omega(p,a) \leq K/2$ and $\omega(p,b) \leq K/2$ and one of the conditions between $\omega(a,b) \leq K/2$ and $\omega(b,a) \leq K/2$ must hold. This results in four possible combinations of the conditions, each of which ensures that p is not a necessary winner. Thus, the last step also gives a correct answer. All checks can be done in linear time. □

8.8 Empirical study of Agenda Control and Coalitional Manipulation

Our polynomial-time algorithms from Sections 8.5 and 8.6 do not compute how many alternatives can win through control (or manipulation). We therefore investigate this empirically. To this end, we use real-world data from the PrefLib collection of preference profiles [MW13] to examine empirically the ratio of profiles that admit successful agenda control or manipulation operations. Since only one case of the possible and the necessary winner problems is polynomial-time solvable and since PrefLib offers only a very restricted variant of incomplete preferences, we do not run experiments for Possible Winner or Necessary Winner. Our algorithms for Agenda Control and Coalitional Manipulation are written in C++ and are available through http://www.akt.tu-berlin.de/menue/software/. Our results are shown in Tables 8.3 and 8.4.

8.8.1 Data background

PrefLib is a library for real-world preferences. The data is provided by various research groups. As of August, 2015, PrefLib contained 314 profiles with complete preference orders: 100 of them have three alternatives, 108 of them have four alternatives, one of them has 7 alternatives, and the remaining 105 profiles have between 9 and 242 alternatives. Among all profiles with complete preference orders, 135 ones have an odd number of voters, where 56 of these have three alternatives, 52 of these have four alternatives, one of these has 7 alternatives, and the remaining 26 profiles have between 14 and 242 alternatives. The number of voters ranges from 5 to 14081.

8.8.2 Agenda Control

For each of the 135 profiles with an odd number of voters (note that for reasons of simplicity, we only implemented our algorithms for odd numbers of voters), using the algorithms from Theorems 8.4 and 8.7, we compute the number m_s (resp. m_a) of alternatives for which a successive (resp. amendment) agenda control is possible. Then, we calculate the *control vulnerability ratio* as

$$\frac{m_s - 1}{m - 1} \quad \text{and} \quad \frac{m_a - 1}{m - 1},$$

where m denotes the number of alternatives. Note that we use $m - 1$ because we factor out the alternative that wins originally. For instance, a control vulnerability ratio of 0.5 means that there are $\frac{m-1}{2}$ alternatives such that each of them can become the winner under an appropriately designed agenda. We use both the arithmetic mean and the geometric mean to compute the average values among all profiles (see Table 8.3). While the arithmetic mean is the sum of the control vulnerability ratios of all profiles divided by t, the geometric mean is the t^{th} root of the product of the control vulnerability ratios of all profiles, where t is the total number of profiles studied.

Summary. Our results show that the successive rule tends to be more vulnerable to agenda control than the amendment rule: For profiles with up to four alternatives, 0.157 of the alternatives have a chance to win under the successive rule, while it is 0 under the amendment rule. For the remaining profiles, the statistics are 0.81 versus 0.035. We observe that no alternative other than the original winner can win under the amendment rule by altering the agenda when the number of alternatives is at most four; there are one third of such profiles. This is remarkable since this means that the original winner of each of the one-third profiles is already a Con-

control vulnerability ratio	$m \leq 4$		$m \geq 5$	
	Successive	Amendment	Successive	Amendment
Arithmetic mean	0.157	0.000	0.081	0.035
Geometric mean	0.000	0.000	0.000	0.000

Table 8.3.: Experiments on agenda control with real-world data. We evaluate all 135 profiles from PrefLib that have linear preference orders and an odd number of voters. We distinguish between profiles with $m \leq 4$ alternatives, and those with $m \geq 5$ alternatives. The reason for this separation is twofold. First, while a large number of profiles has either three or four alternatives (one third each), for $m \geq 5$, in most cases, less than *five* profiles have m alternatives. Second, the results for profiles with up to four alternatives are pretty different from the other profiles.

dorcet winner. The values of the geometric are zero since there is at least one profile for which there is a Condorcet winner.

8.8.3 Manipulation

Since PrefLib does not offer any agenda, we have to generate a set $\mathcal{Z}$ of agendas for manipulation to obtain a good representation. The size of $\mathcal{Z}$ depends on the number m of alternatives:

1. If $m \leq 8$, then let $\mathcal{Z}$ be the set of all possible agendas, that is, $|\mathcal{Z}| := m!$.

2. Otherwise, we generate a set $\mathcal{Z}$ of agendas by randomly choosing $|\mathcal{Z}| := \min(n^2, 8!)$ agendas from the set of all possible agendas, where n denotes the number of voters in the input. All possible agendas are equally likely of being chosen for $\mathcal{Z}$.

For each alternative c and for each agenda $\mathcal{L} \in \mathcal{Z}$, using the algorithms behind Theorems 8.9 and 8.11, we compute the minimum coalition size, that is, the minimum number of voters needed to make c a successive (resp. an amendment) winner. Let this number for the successive (resp. amendment) rule be $\kappa_s(\mathcal{P}, c, \mathcal{L})$ (resp. $\kappa_a(\mathcal{P}, c, \mathcal{L})$). This number each time is upper-bounded by $n+1$, where n denotes the number of voters in the given profile.

We calculate the *manipulation resistance ratio* as

$$\frac{\sum_{\mathcal{L} \in \mathcal{Z}} \sum_{c \in C} \kappa_s(\mathcal{P}, c, \mathcal{L})}{|\mathcal{Z}| \cdot (m-1) \cdot (n+1)} \quad \text{and} \quad \frac{\sum_{\mathcal{L} \in \mathcal{Z}} \sum_{c \in C} \kappa_a(\mathcal{P}, c, \mathcal{L})}{|\mathcal{Z}| \cdot (m-1) \cdot (n+1)}.$$

Measurements:	$m \le 8$		$m \ge 9$	
the ratio of…	Successive	Amendment	Successive	Amendment
…manipulation resistance	0.455	0.402	0.945	0.924
…2nd winner coalition	0.288	0.222	0.523	0.468
…smallest coalition	0.263	0.221	0.386	0.385

Table 8.4.: Experiments on manipulation with real-world data. We evaluate all 314 profiles from PrefLib that have linear preference orders. We separately consider profiles with $m \le 8$ and $m \ge 9$ alternatives. The reason for this separation is that we only consider all $m!$ possible agendas when $m \le 8$. As all three measures are ratios between $[0,1]$, we use the geometric mean to compute the average and omit showing the values using the arithmetic mean since they are similar to the ones obtained with the geometric mean.

For instance, a manipulation resistance ratio of 0.4 means that each alternative in a profile needs a coalition of $0.4\cdot(n+1)$ additional manipulators, on average, to become a winner.

From our experimental results, we observe that on average, a successful manipulation needs at least $n/2$ manipulators (see the first row in Table 8.4), where n denotes the number of voters in the given profile. This is because in order to become a winner, most alternatives need a coalition of at least $n+1$ additional voters. To get a more thorough understanding of the manipulation issue, however, we consider two related concepts:

- The ratio of the 2*nd* *winner coalition size*, that is, the coalition size for the alternative that becomes a winner after the original winner is removed. Formally, the 2nd winner coalition is defined as
$$\frac{\sum_{\mathscr{L}\in\mathcal{Z}} \kappa_s(\mathscr{P}, c^*, \mathscr{L})}{|\mathcal{Z}|\cdot(n+1)} \quad \text{and} \quad \frac{\sum_{\mathscr{L}\in\mathcal{Z}} \kappa_a(\mathscr{P}, c^*, \mathscr{L})}{|\mathcal{Z}|\cdot(n+1)},$$
where c^* is a successive (resp. an amendment) winner after the original winner is removed.

- The ratio of the *smallest coalition size*, that is, the size $\sigma_s(\mathscr{P},\mathscr{L})$ (resp. $\sigma_a(\mathscr{P},\mathscr{L})$) of the smallest coalition that makes some alternative other than the original winner win. Formally, the smallest coalition is defined as
$$\frac{\sum_{\mathscr{L}\in\mathcal{Z}} \sigma_s(\mathscr{P},\mathscr{L})}{|\mathcal{Z}|\cdot(n+1)} \quad \text{and} \quad \frac{\sum_{\mathscr{L}\in\mathcal{Z}} \sigma_a(\mathscr{P},\mathscr{L})}{|\mathcal{Z}|\cdot(n+1)}.$$

Summary. Our results show that successful manipulations with few voters are rare; let n denote the number of voters in the respective profile: For profiles with up to eight alternatives the average coalition size is at least $0.4 \cdot (n+1)$ (even the 2nd winner coalition size is $0.2 \cdot (n+1)$; the smallest coalition size is only slightly lower), while for profiles with at least nine alternatives the average coalition size is almost $n+1$ (even the 2nd winner coalition size is roughly $0.5 \cdot (n+1)$). Comparing the two rules, it seems that making an alternative win under the successive rule requires more manipulators than making it win under the amendment rule. The difference, however, is not significant.

8.9 Concluding remarks

Our work indicates that, from a computational perspective, the amendment rule seems to have superior resistance against agenda control and strategic voting, compared to the successive rule. This supports the suggestion made by Apesteguia, Ballester, and Masatlioglu [ABM14] that most European and Latin American parliaments should rather go the Anglo-American way, that is, they should use the amendment rule instead of the successive rule.

In practice there are many other procedural complications regarding how and when alternatives can be placed, the order of such alternatives, as well as modifications to these simple voting rules. For example, in some parliaments, the parliament can vote on reordering the alternatives. Studying such complications is one interesting direction for future work.

We see this chapter as an initial step towards computational complexity studies of the two parliamentary voting rules. After identifying the computational complexity of several voting problems for both studied rules, it would be natural and interesting to further explore and exploit the structures to identify tractable special cases, as we have done for the combinatorial voter control problem and the shift bribery problem in the previous two chapters. In particular, following the spirit of Betzler, Hemmann, and Niedermeier [BHN09], it would be of interest to complement our computational hardness results for possible and necessary winner problems with a refined complexity analysis concerning tractable special cases. For instance, our NP-hardness reductions for POSSIBLE WINNER assume that voters may have arbitrary partial preferences. It would be interesting to know whether this still holds if voter preferences have a certain structure, such as, for instance, being single-peaked or semi-single-peaked [Ras87] (also see Section 5.5.2 for the corresponding definition of single-peakedness).

For both rules, we obtain that their NP-complete POSSIBLE WINNER are fixed-parameter tractable when parameterized "number m of alternatives". This automatically yields a problem kernel for m (see Section 2.6.1 for more information on problem kernels). However, we remark that similarly to the case of combinatorial voter control studied in Chapter 6, where we note the absence of polynomial kernels, we can adapt our polynomial reductions for POSSIBLE WINNER to provide a polynomial-parameter transformation from SET COVER (see the definition at the beginning of the proof of Theorem 6.6) parameterized by the universe size to POSSIBLE WINNER parameterized by m. By Proposition 2.1 and by the fact that the unparameterized versions of both problems are NP-complete, we obtain that a polynomial kernel does not exist unless a complexity-theoretic assumption ($\mathsf{NP} \not\subseteq \mathsf{coNP/poly}$) fails. For the parameter "number n of voters", however, we could not settle the parameterized complexity of either problem yet.

It would be natural to also adopt a more game-theoretic view on the strategic voting scenarios where all voters act strategically, as opposed to COALITIONAL MANIPULATION, where only the voters of a coalition vote strategically (with the same goal) and all other voters vote sincerely. Note that we can see both the successive and amendment rules as repeated games, so a natural concept to consider is the subgame-perfect Nash equilibrium [Osb04].

Finally, it would be also interesting to study further manipulation scenarios for parliamentary voting rules including, for example, the sophisticated voting setting, which has been widely studied in the political science literature (see [Ban85, EK80, Mil77, MN78, Mou86, OS87, SW82]), or control by changing the set of alternatives as discussed by Rasch [Ras14].

CHAPTER 9

Conclusion

The Great Way is not difficult for those who have no preferences.
Zen proverb

In the first half of this thesis, Chapters 3 to 5, we have studied various nicely structured preference profiles, such as single-crossing or one-dimensional Euclidean profiles. We also studied nearly nicely structured preference profiles and investigated questions concerning the computational cost of achieving nicely structured profiles by deleting the fewest number of voters or alternatives, respectively. Our results draw a clear line between polynomial-time solvable and NP-hard problems as they classify all considered problem variants with respect to their classical computational complexity.

In the second half, Chapters 6 to 8, we have investigated computationally hard problems arising in the context of voting. Almost all of these problems are about manipulating the voting outcome, but in different ways. For instance, the combinatorial variant of the CONSTRUCTIVE CONTROL BY ADDING VOTERS (CC-AV) problem (Chapter 6) is about making a specific alternative win by persuading the fewest number of unregistered voters to vote, where each persuaded voter also adds a bundle of other voters for free. The SHIFT BRIBERY problem (Chapter 7) is about making a specific alternative win through campaign management, that is, shifting her higher in the preference orders of some voters; each such shift has a cost and the goal is not to exceed a given budget. For each of these problems, we have identified structures such that the problem restricted to one of these structures remains intractable, as well as some structures such that the problem restricted to one of these structures is of a presumably lower complexity. For instance, the combinatorial variant of CC-AV for the case where the bundling function is anonymous remains intractable for single-peaked preferences, while it is polynomial-time solvable for single-crossing preferences.

In this chapter, we summarize our main contributions (Section 9.1) and highlights, and discuss some future research directions and challenges that we find most interesting (Section 9.2).

9.1 Main contributions

Motivated by Ballester and Haeringer's work [BH11] on exploring sufficient and necessary conditions for the existence of single-peaked preferences by forbidden substructures, we studied the single-crossing preferences and we showed that a profile is single-crossing if and only if it contains neither γ-configurations nor δ-configurations (see Examples 3.4 and 3.5 and Section 3.5). Since both γ-configurations and δ-configurations consist of at most six alternatives and at most four voters, the characterization allows us to detect the single-crossing property in $O(n \cdot m^2)$ time, without having to probe every possible order of the voters.

In Chapter 4, we moved on to a more restricted, but arguably more natural preference structure, the one-dimensional Euclidean preferences [Hot29]. This structure is a restriction of both single-peakedness and single-crossingness but, in contrast, it cannot be characterized by finitely many forbidden substructures. We constructed an infinite sequence of preference profiles such that (a) none of these profiles is one-dimensional Euclidean, and (b) the deletion of an arbitrary voter in each of the constructed profiles immediately makes the profile one-dimensional Euclidean (see Theorem 4.3).

The notions of single-peakedness and single-crossingness are popular assumptions among economists and politicians. Yet, real-world preference profiles may contain some noise which destroys these nice structures. Keeping this in mind, in Chapter 5 we studied the distance of a given profile from having a nice structure, such as single-peakedness, single-crossingness, and group-separability. We showed that in almost all cases computing this distance is computationally hard.

Summarizing the results of Chapters 3 to 5, we find that nothing interesting is ever one-sided. We have both a positive case where finitely many forbidden substructures are *sufficient* to characterize existing nice properties, and a negative case where finitely many forbidden substructures are *insufficient*. From the computational perspective, we presented an efficient, polynomial-time algorithm to make a given preference profile single-crossing by deleting the fewest number of voters, while we showed that for most other cases, including achieving single-peakedness by deleting the fewest number of voters, an efficient algorithm is unlikely unless the well-accepted assumption of $\mathsf{P} \neq \mathsf{NP}$ fails.

In Chapters 6 and 7, we turned to new aspects of two conceptually different voting problems, the CONSTRUCTIVE CONTROL BY ADDING VOTERS problem [BTT92] and the SHIFT BRIBERY problem [EFS09]. In Chapter 6, we introduced a combinatorial variant of CONSTRUCTIVE CONTROL BY ADDING VOTERS, where adding a voter also means adding a bundle of other voters for free. We found that the combinatorial aspect makes the problem computationally hard even for two alternatives. For the plurality rule, we exploited structures in this extremely hard problem in two ways:

First, we identified specific natural parameters and applied parameterized complexity techniques. We focused on the parameters "number k of the bundles allowed to add", maximum bundle size, and the similarity measure for each bundle. We obtained an almost complete picture of the complexity with respect to these three parameters. Second, we considered the case with nicely structured preference profiles and obtained two surprisingly different results: the problem with the voter bundles displaying an anonymous structure remains computationally hard for single-peaked profiles, while it is polynomial-time solvable for single-crossing profiles.

In Chapter 7, we studied how different natural prices of voters and different parameters affect the complexity of the SHIFT BRIBERY problem for three popular voting rules. We also presented an approximation algorithm which approximates the budget within a factor of $(1+\varepsilon)$ and has a running time whose super-polynomial part depends only on the approximation parameter ε and the parameter "number n of voters".

The previous two manipulative attacks can be applied to every voting rule. Thus, we study them for some common voting rules such as the plurality rule and the Borda rule. These rules basically proceed in a single stage. Yet, there are other voting rules which work in stages. For these rules, there may be manipulative attacks specific to them. Thus, in Chapter 8, we focused on two sequential voting rules, the successive rule and the amendment rule, that are used in the parliaments of many countries [Ras00]; both rules use a linear order of the alternatives, which we call an *agenda*. We discussed computational questions for both rules pertaining to agenda control, manipulation, and uncertainty in voting. Our theoretical results indicate that manipulation and agenda control are computationally easier than asking for possible or necessary winners with uncertainty in the voting. This is quite interesting since both manipulation and agenda control can be considered as special cases of the possible winner problem, where the former assumes either complete or empty preferences of each voter but a fixed agenda, while the latter assumes complete preferences of each voter, but an empty agenda. With this in mind, we concluded that it is the fact that the preferences and the agenda are partial orders which makes the problem computationally hard.

9.2 Outlook and open questions

In this section, we discuss some assumptions we make when we study preferences and voting problems (except for the case when asking for possible/necessary winners), and we list some specific open questions which we find most exciting.

Linear orders as preferences. Throughout this thesis, we almost always assumed that each voter ranks the alternatives, from best to worst, according to a linear order over the alternatives; the only exception is Chapter 8, where we allowed the preferences to be partial orders. This is of course a very strict requirement. In political elections, however, we usually approve and disapprove of alternatives instead of ranking them. Human perception is so limited that it is almost impossible to give a linear order over more than a dozen alternatives (also see the work of Balinski and Laraki [BL11] for more discussion on the problems of rank-ordering alternatives). Linear orders ignore the case where alternatives are incomparable, while partial orders are too general to model real-world societal voting behavior. Furthermore, voter preferences may vary from time to time as Nathan Myhrvold indicates:

> *"Researchers who examined the voting records of wine judges found that 90 percent of the time they give inconsistent ratings to a particular wine when they judge it on multiple occasions."*[1]

Keeping this in mind, we can, for example, assume that each voter has a certain probability for each possible preference order. It would be interesting to reexamine the notion of classical preference structure and voting problems for this probabilistic model.

Manipulative attacks. To model the effect of large-scale campaign management, we introduced a combinatorial variant of the already known problem of CONSTRUCTIVE CONTROL BY ADDING VOTERS, where adding a voter means also adding a subset of other voters. This addition only has a one-time effect in the sense that the subset of voters who are added this way do not, in turn, have any influence on other voters. A cascading effect may be worth studying as well, where the voters entering the voting due to some other voter may in turn cause further voters to be added.

Following the convention in the literature [EFS09], we assumed that we know the cost of each shift for a voter when we study the SHIFT BRIBERY problem. This may be a bit unrealistic as, first, the cost of each shift can also depend on the

[1] http://www.bloomberg.com/bw/magazine/how-to-decant-wine-with-a-blender-09222011.html

alternatives involved instead of only on the number of positions shifted, and, second, determining the exact cost of each shift of each voter seems tedious if not impossible. Thus, a more fine-grained model of this voting problem is desirable.

Empirical studies. Most of the problems studied in this thesis are computationally hard, and our goal in studying such hard voting problems is to clarify the boundary between fixed-parameter tractable and parameterized intractable cases. Any problem that can be solved in $k^{O(k)} \cdot |I|^4$ time for input I, is already fixed-parameter tractable for the parameter k, although it may take a long time even for moderately-sized input. Obviously, when running experiments for such a problem, a fixed-parameter algorithm with a running time upper bound lower than $k^{O(k)} \cdot |I|^4$ would be desirable.

While this would be an interesting research direction, it is *not* the main concern of this thesis, which is to chart the border between tractable and intractable cases. Thus, we did not perform any experiments except for two manipulative attack problems studied in Chapter 8: AGENDA CONTROL and COALITIONAL MANIPULATION. Both problems are polynomial-time solvable for both voting rules studied in Chapter 8. While polynomial-time solvability for manipulative attacks seems bad, it does not immediately imply that we can always successfully manipulate the outcome. To gain deeper insight into the likelihood of successful manipulative attacks, we carried out an empirical study on real-world preference profiles and our results greatly eased our worries because we found that, in a nutshell, a successful manipulative attack is very rare. With this in mind, we should bridge the gap between theory and real life, and explore the remaining problems once more from an experimental point of view.

Challenges. There are a number of research challenges and open questions arising from this thesis. We have already discussed some of them in the concluding sections in Chapters 3 to 8. We close with a few more which are not restricted to the focus of a specific chapter and which we deem particularly worthwhile of investigation.

1. It would be of great interest to find polynomial kernels for the parameterized variants of our problems that are fixed-parameter tractable, or to show that they are unlikely to exist under some complexity-theoretic assumption (see Section 2.6.1 for more information on the concept of polynomial kernels).

2. As already discussed, complementing our theoretical findings with empirical studies is, in itself, an interesting future research direction. It is also quite challenging because running experiments on large-scale real-world data needs

algorithms that have a reasonable running time; our parameterized algorithms are so far of theoretical interest only. This would be even more interesting if the relevant problem admits a polynomial kernel because this allows us to have *efficient* preprocessing algorithms to shrink the input instances to obtain a problem kernel that is polynomial in the parameter, and solve the kernel exactly.

3. In Chapters 3 and 4, we have studied nicely structured profiles such as the single-peaked, single-crossing, and one-dimensional Euclidean profiles. In Chapter 5, we expanded this by investigating profiles that are close to being single-peaked or single-crossing. Investigating profiles that are close to being one-dimensional Euclidean however, remains unexplored. Thus, it would be interesting to know the computational complexity of deleting the fewest number of alternatives (resp. voters) to make a profile one-dimensional Euclidean.

4. In this thesis, we have studied several types of voting problems. Most of them are already known to be, or have been shown in this thesis to be intractable, and we paid major attention to finding special structures which help to make the respective problem tractable. These structures could be about properties of the voters' preferences, such as single-peakedness, or about quantifiable measures (or *parameters*) of the input. Yet, we mostly focused on the parameter structure rather than on the preference structure. We obtained only two results on single-peakedness and on single-crossingness in Chapter 6, where we studied the COMBINATORIAL CONSTRUCTIVE CONTROL BY ADDING VOTERS problem. We deem it worthwhile to further study other voting problems for structured preferences. For instance, it would be interesting to know whether the NP-hardness results of POSSIBLE WINNER for the two studied parliamentary voting rules still hold if the voters preferences are single-peaked or single-crossing.

Many reductions, showing intractability results, produce instances that are far from being close to any known real-world instances. Thus, in order to identify special tractable cases, with the intuition that "no structure no efficacy", we explored and exploited structures for several computationally hard voting problems. We hope that a more in-depth structure exploitation can be made in the computational complexity study of problems even outside of the voting context.

Bibliography

[AB09] S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009 (cited on p. 16).

[Abe91] J. Abello. “The weak Bruhat order of S_Σ, consistent sets, and Catalan numbers”. In: *SIAM Journal on Discrete Mathematics* 4(1) (1991), pp. 1–16 (cited on p. 29).

[ABM14] J. Apesteguia, M. A. Ballester, and Y. Masatlioglu. “A foundation for strategic agenda voting”. In: *Games and Economic Behavior* 87 (2014), pp. 91–99 (cited on pp. 186, 188, 226).

[AJ84] J. M. Abello and C. R. Johnson. “How large are transitive simple majority domains?” In: *SIAM Journal on Algebraic and Discrete Methods* 5(4) (1984), pp. 603–618 (cited on p. 79).

[Alo+13] N. Alon, R. Bredereck, J. Chen, S. Kratsch, R. Niedermeier, and G. J. Woeginger. “How to put through your agenda in collective binary decisions”. In: *Proceedings of the 3rd International Conference on Algorithmic Decision Theory*. Vol. 8176. Lecture Notes in Computer Science. 2013, pp. 30–44 (cited on p. xiii).

[Alo+15] N. Alon, R. Bredereck, J. Chen, S. Kratsch, R. Niedermeier, and G. J. Woeginger. “How to put through your agenda in collective binary decisions”. In: *ACM Transactions on Economics and Computation* 4(1) (2015), pp. 1–28 (cited on p. xiii).

[AMO93] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993 (cited on p. 177).

[Arr50] K. J. Arrow. “A difficulty in the concept of social welfare”. In: *Journal of Political Economy* 58(4) (1950), pp. 328–346 (cited on pp. 3, 16, 71).

[ASS02] K. J. Arrow, A. K. Sen, and K. Suzumura, eds. *Handbook of Social Choice and Welfare, Volume 1*. North-Holland, 2002 (cited on pp. i, v, ix, 15, 72).

[ASS10] K. J. Arrow, A. K. Sen, and K. Suzumura, eds. *Handbook of Social Choice and Welfare, Volume 2*. North-Holland, 2010 (cited on pp. i, v, ix).

[Azi+15] H. Aziz, M. Brill, F. A. Fischer, P. Harrenstein, J. Lang, and H. G. Seedig. “Possible and necessary winners of partial tournaments”. In: *Journal of Artificial Intelligence Research* 54 (2015), pp. 493–534 (cited on p. 188).

[Bai87] N. Baigent. “Metric rationalisation of social choice functions according to principles of social choice”. In: *Mathematical Social Sciences* 13(1) (1987), pp. 59–65 (cited on p. 76).

[Ban85] J. S. Banks. “Sophisticated voting outcomes and agenda control”. In: *Social Choice and Welfare* 1(4) (1985), pp. 295–306 (cited on pp. 187, 227).

[Bau+12] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. "Campaigns for lazy voters: truncated ballots". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '12)*. IFAAMAS, 2012, pp. 577–584 (cited on p. 154).

[BC] L. Bulteau and J. Chen. *Embedding preferences into euclidean space of small dimensions*. Working paper (cited on pp. 44, 70).

[BCW13a] R. Bredereck, J. Chen, and G. J. Woeginger. "A characterization of the single-crossing domain". In: *Social Choice and Welfare* 41(4) (2013), pp. 989–998 (cited on pp. x, 7, 25, 41).

[BCW13b] R. Bredereck, J. Chen, and G. J. Woeginger. "Are there any nicely structured preference profiles nearby?" In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 2013, pp. 62–68 (cited on pp. x, 7).

[BCW16] R. Bredereck, J. Chen, and G. J. Woeginger. "Are there any nicely structured preference profiles nearby?" In: *Mathematical Social Sciences* 79 (2016), pp. 61–73 (cited on pp. x, 7, 71).

[BD10] N. Betzler and B. Dorn. "Towards a dichotomy of finding possible winners in elections based on scoring rules". In: *Journal of Computer and System Sciences* 76(8) (2010), pp. 812–836 (cited on pp. 153, 188).

[Bet+11] N. Betzler, J. Guo, C. Komusiewicz, and R. Niedermeier. "Average parameterization and partial kernelization for computing medians". In: *Journal of Computer and System Sciences* 77(4) (2011), pp. 774–789 (cited on pp. 155, 163, 165).

[Bet+12] 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*. Vol. 7370. Lecture Notes in Computer Science. Springer, 2012, pp. 318–363 (cited on pp. 21, 116).

[Bev+14a] R. van Bevern, R. Bredereck, L. Bulteau, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. "Star partitions of perfect graphs". In: *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming (ICALP '14)*. Vol. 8572. Lecture Notes in Computer Science. Springer, 2014, pp. 174–185 (cited on p. xiii).

[Bev+14b] R. van Bevern, R. Bredereck, J. Chen, V. Froese, R. Niedermeier, and G. J. J. Woeginger. "Network-based dissolution". In: *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14)*. Vol. 8635. LNCS. Springer, 2014, pp. 69–80 (cited on p. xiii).

[Bev+15a] R. van Bevern, R. Bredereck, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. "Network-based vertex dissolution". In: *SIAM Journal on Discrete Mathematics* 29(2) (2015), pp. 888–914 (cited on p. xiii).

[Bev+15b] R. van Bevern, J. Chen, F. Hüffner, S. Kratsch, N. Talmon, and G. J. Woeginger. "Approximability and parameterized complexity of multicover by c-intervals". In: *Information Processing Letters* 115(10) (2015), pp. 744–749 (cited on p. xiii).

[Bev+16] R. van Bevern, R. Bredereck, L. Bulteau, J. Chen, V. Froese, R. Niedermeier, and G. J. Woeginger. "Star partitions of perfect graphs". In: *Journal of Graph Theory* (2016). Accepted (cited on p. xiii).

[BG16] S. Barberà and A. Gerber. *Sequential Voting and Agenda Manipulation*. Tech. rep. Universitat Autònoma de Barcelona and Barcelona GSE, 2016 (cited on p. 188).

[BG93] J. F. Buss and J. Goldsmith. "Nondeterminism within P". In: *SIAM Journal on Computing* 22(3) (1993), pp. 560–572 (cited on p. 18).

[BGE93] S. Barberà, F. Gul, and S. Ennio. "Generalized median voter schemes and committees". In: *Journal of Economic Theory* 61(2) (1993), pp. 262–289 (cited on p. 75).

[BGN10] N. Betzler, J. Guo, and R. Niedermeier. "Parameterized computational complexity of Dodgson and Young elections". In: *Information and Computation* 208(2) (2010), pp. 165–177 (cited on p. 175).

[BH11] M. Ballester and G. Haeringer. "A characterization of the single-peaked domain". In: *Social Choice and Welfare* 36(2) (2011), pp. 305–322 (cited on pp. ix, 26, 37, 39, 41, 77, 78, 80, 81, 230).

[BHN09] N. Betzler, S. Hemmann, and R. Niedermeier. "A multivariate complexity analysis of determining possible winners given incomplete votes". In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI '09)*. AAAI Press, 2009, pp. 53–58 (cited on pp. 175, 188, 226).

[BJ04] S. Barberà and M. O. Jackson. "Choosing how to choose: self-stable majority rules and constitutions". In: *Quarterly Journal of Economics* 119(3) (2004), pp. 1011–1048 (cited on p. 26).

[BJK02] S. J. Brams, M. A. Jones, and D. M. Kilgour. "Single-peakedness and disconnected coalitions". In: *Journal of Theoretical Politics* 14(3) (2002), pp. 359–383 (cited on p. 40).

[BJK14] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. "Kernelization lower bounds by cross-composition". In: *SIAM Journal on Discrete Mathematics* 28(1) (2014), pp. 277–305 (cited on p. 18).

[BL07] A. Bogomolnaia and J.-F. Laslier. "Euclidean preferences". In: *Journal of Mathematical Economics* 43(2) (2007), pp. 87–98 (cited on pp. 3, 70).

[BL11] M. Balinski and R. Laraki. *Majority Judgment. Measuring, Ranking, and Electing*. MIT Press, 2011 (cited on p. 232).

[BL76] K. S. Booth and G. S. Lueker. “Testing for the consecutive ones property, interval graphs, and graph planarity using PQ-tree algorithms”. In: *Journal of Computer and System Sciences* 13(3) (1976), pp. 335–379 (cited on pp. i, v, 35, 36).

[Bla48] D. Black. “On the rationale of group decision making”. In: *Journal of Political Economy* 56(1) (1948), pp. 23–34 (cited on pp. ix, 2, 25, 39, 72, 108, 147).

[Bla58] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958 (cited on pp. 186, 187).

[BM11] S. Barberà and B. Moreno. “Top monotonicity: a common root for single peakedness, single crossing and the median voter result”. In: *Games and Economic Behavior* 73(2) (2011), pp. 345–359 (cited on p. 26).

[BO91] J. J. Bartholdi III and J. B. Orlin. “Single transferable vote resists strategic voting”. In: *Social Choice and Welfare* 8(4) (1991), pp. 341–354 (cited on pp. 3, 188).

[Bod+09] H. L. Bodlaender, F. V. Fomin, D. Lokshtanov, E. Penninkx, S. Saurabh, and D. M. Thilikos. “(meta) kernelization”. In: *Proc. 50th FOCS*. IEEE Computer Society, 2009, pp. 629–638 (cited on p. 18).

[Bon04] M. Bona. *Combinatorics of Permutations*. Chapman and Hall/CRC, 2004 (cited on p. 29).

[Bou+04] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. “CP-nets: a tool for representing and reasoning with conditional *ceteris paribus* preference statements”. In: *Journal of Artificial Intelligence Research* 21 (2004), pp. 135–191 (cited on p. 109).

[BR12] D. Baumeister and J. Rothe. “Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules”. In: *Information Processing Letters* 112(5) (2012), pp. 186–190 (cited on pp. 153, 188).

[Bra+13] F. Brandt, P. Harrenstein, K. Kardel, and H. G. Seedig. “It only takes a few: On the hardness of voting with a constant number of agents”. In: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS ’13)*. IFAAMAS, 2013, pp. 375–382 (cited on pp. 108, 175).

[Bra+15] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. “Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates”. In: *Journal of Artificial Intelligence Research* 53 (2015), pp. 439–496 (cited on pp. 72, 76, 106–108, 142).

[Bra+16] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, eds. *Handbook of Computational Social Choice*. Cambridge University Press, 2016 (cited on p. 21).

[Bre+08] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor. “Approximability of manipulating elections”. In: *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI ’08)*. AAAI Press, 2008, pp. 44–49 (cited on p. 153).

[Bre+12] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, and O. Suchý. "A multivariate complexity analysis of lobbying in multiple referenda". In: *Proceedings of the 26th Conference on Artificial Intelligence*. AAAI Press, 2012, pp. 1292–1298 (cited on p. xiii).

[Bre+13] R. Bredereck, J. Chen, S. Hartung, C. Komusiewicz, R. Niedermeier, and O. Suchý. "On explaining integer vectors by few homogenous segments". In: *Proceedings of the 13th International Workshop on Algorithms and Data Structures (WADS '13)*. Vol. 8037. Lecture Notes in Computer Science. Springer, 2013, pp. 207–218 (cited on p. xiii).

[Bre+14a] R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. "Parameterized algorithmics for computational social choice: Nine research challenges". In: *Tsinghua Science and Technology* 19(4) (2014), pp. 358–373 (cited on pp. 21, 102, 116, 214).

[Bre+14b] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. "Prices matter for the parameterized complexity of shift bribery". In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '14)*. AAAI Press, 2014, pp. 1398–1404 (cited on pp. xii, 7, 154, 176, 182).

[Bre+14c] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, O. Suchý, and G. J. Woeginger. "A multivariate complexity analysis of lobbying in multiple referenda". In: *Journal of Artificial Intelligence Research* 50 (2014), pp. 409–446 (cited on pp. xiii, 175).

[Bre+15a] R. Bredereck, J. Chen, S. Hartung, C. Komusiewicz, R. Niedermeier, and O. Suchý. "On explaining integer vectors by few homogeneous segments". In: *Journal of Computer and System Sciences* 81(4) (2015), pp. 766–782 (cited on p. xiii).

[Bre+15b] R. Bredereck, J. Chen, R. Niedermeier, and T. Walsh. "Parliamentary voting procedures: Agenda control, manipulation, and uncertainty". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI '15)*. AAAI Press, 2015, pp. 164–170 (cited on pp. xiii, 8, 185).

[Bre+15c] R. Bredereck, J. Chen, R. Niedermeier, and T. Walsh. *Parliamentary Voting Procedures: Agenda Control, Manipulation, and Uncertainty*. Tech. rep. arXiv:1509.02424v1 [cs.GT], 2015 (cited on p. 185).

[Bre+15d] R. Bredereck, P. Faliszewski, R. Niedermeier, P. Skowron, and N. Talmon. "Elections with few candidates: Prices, weights, and covering problems". In: *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT '15)*. Vol. 9346. Lecture Notes in Computer Science. Springer, 2015, pp. 414–431 (cited on pp. 154, 175, 176).

[Bre+15e] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. "Large-scale election campaigns: Combinatorial shift bribery". In: *Proceedings of the 14th International*

Conference on Autonomous Agents and Multiagent Systems (AAMAS '15). IFAAMAS, 2015, pp. 67–75 (cited on pp. 150, 155, 156, 183).

[Bre+16a] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. "Prices matter for the parameterized complexity of shift bribery". In: *Information and Computation* (2016). To appear (cited on pp. xii, 7, 151, 157, 168, 171, 174, 176, 182).

[Bre+16b] R. Bredereck, J. Chen, F. Hüffner, and S. Kratsch. "Parameterized complexity of team formation in social networks". In: *Proceedings of the 11th International Conference on Algorithmic Aspects in Information and Management (AAIM '16)*. Vol. 9778. Lecture Notes in Computer Science. Springer, 2016, pp. 137–149 (cited on p. xiii).

[Bre+16c] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. "Complexity of shift bribery in committee elections". In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI '16)*. AAAI Press, 2016, pp. 2452–2458 (cited on pp. 154, 156, 182).

[Bre+16d] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. "Large-scale election campaigns: Combinatorial shift bribery". In: *Journal of Artificial Intelligence Research* 55 (2016), pp. 603–652 (cited on p. 154).

[BSU13] N. Betzler, A. Slinko, and J. Uhlmann. "On the computation of fully proportional representation". In: *Journal of Artificial Intelligence Research* 47(1) (2013), pp. 475–519 (cited on p. 76).

[BT86] J. J. Bartholdi III and M. Trick. "Stable matching with preferences derived from a psychological model". In: *Operations Research Letters* 5(4) (1986), pp. 165–169 (cited on pp. 4, 37, 41).

[BTT89] 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". In: *Social Choice and Welfare* 6(2) (1989), pp. 157–165 (cited on pp. 3, 16, 71, 175, 188).

[BTT92] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. "How hard is it to control an election". In: *Mathematical and Computer Modelling* 16(8–9) (1992), pp. 27–40 (cited on pp. ii, vi, 5, 105–107, 113, 114, 124, 137, 183, 230).

[BTY11] H. L. Bodlaender, S. Thomassé, and A. Yeo. "Kernel bounds for disjoint cycles and disjoint paths". In: *Theoretical Computer Science* 412(35) (2011), pp. 4570–4578 (cited on pp. 18, 19).

[BU09] N. Betzler and J. Uhlmann. "Parameterized complexity of candidate control in elections and related digraph problems". In: *Theoretical Computer Science* 410(52) (2009), pp. 5425–5442 (cited on p. 108).

[Bul+15] L. Bulteau, J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. "Combinatorial voter control in elections". In: *Theoretical Computer Science* 589 (2015), pp. 99–120 (cited on pp. xii, 7, 105).

[CGS12] D. Cornaz, L. Galand, and O. Spanjaard. "Bounded single-peaked width and proportional representation". In: *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI '12)*. IOS Press, 2012, pp. 270–275 (cited on p. 74).

[CGS13] D. Cornaz, L. Galand, and O. Spanjaard. "Kemeny elections with bounded single-peaked or single-crossing width". In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 2013, pp. 76–82 (cited on p. 74).

[Che+13] J. Chen, C. Komusiewicz, R. Niedermeier, M. Sorge, O. Suchý, and M. Weller. "Effective and efficient data reduction for the subset interconnection design problem". In: *Proceedings of the 24th International Symposium on Algorithms and Computation (ISAAC '07)*. Vol. 8283. Lecture Notes in Computer Science. 2013, pp. 361–371 (cited on p. xiii).

[Che+14] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. "Combinatorial voter control in elections". In: *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14)*. Vol. 8635. Lecture Notes in Computer Science. Springer, 2014, pp. 153–164 (cited on pp. xii, 7).

[Che+15a] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. "Elections with few voters: Candidate control can be easy". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*. AAAI Press, 2015, pp. 2045–2051 (cited on pp. xiii, 108, 150, 175).

[Che+15b] J. Chen, C. Komusiewicz, R. Niedermeier, M. Sorge, O. Suchý, and M. Weller. "Polynomial-time data reduction for the subset interconnection design problem". In: *SIAM Journal on Discrete Mathematics* 29(1) (2015), pp. 1–25 (cited on p. xiii).

[CLX09] V. Conitzer, J. Lang, and L. Xia. "How hard is it to control sequential elections via the agenda?" In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI '09)*. AAAI Press, 2009, pp. 103–108 (cited on p. 109).

[Con09] V. Conitzer. "Eliciting single-peaked preferences using comparison queries". In: *Journal of Artificial Intelligence Research* 35 (2009), pp. 161–191 (cited on pp. 107, 108).

[Con85] M. J. A. N. C. de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: L'Imprimerie Royale, 1785 (cited on pp. 1, 2, 15).

[Coo64] C. H. Coombs. *A Theory of Data*. John Wiley and Sons, 1964 (cited on pp. 25, 40, 42, 44).

[Cor+09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2009 (cited on pp. 99, 197).

[CPW15] J. Chen, K. Pruhs, and G. J. Woeginger. *The one-dimensional Euclidean domain: Finitely many obstructions are not enough*. Tech. rep. arXiv:1506.03838v1 [cs.GT], 2015 (cited on p. 7).

[CPW16] J. Chen, K. Pruhs, and G. J. Woeginger. "The one-dimensional Euclidean domain: Finitely many obstructions are not enough". In: *Social Choice and Welfare* (2016). To appear (cited on pp. xi, 7, 39).

[CSL07] V. Conitzer, T. Sandholm, and J. Lang. "When are elections with few candidates hard to manipulate?" In: *Journal of the ACM* 54(3) (2007), pp. 1–33 (cited on pp. 189, 193).

[Cyg+15] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015 (cited on pp. i, v, 19, 21).

[Dem94] G. Demange. "Intermediate preferences and stable coalition structures". In: *Journal of Mathematical Economics* 23(1) (1994), pp. 45–58 (cited on p. 26).

[DF13] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013 (cited on pp. i, v, 21, 117, 118).

[DF94] J. Doignon and J. Falmagne. "A polynomial time algorithm for unidimensional unfolding representations". In: *Journal of Algorithms* 16(2) (1994), pp. 218–233 (cited on pp. 37, 40, 41, 47, 73, 100, 102).

[DLS14] M. Dom, D. Lokshtanov, and S. Saurabh. "Kernelization lower bounds through colors and IDs". In: *ACM Transactions on Algorithms* 11(13) (2 2014), pp. 1–13 (cited on p. 124).

[DS12] B. Dorn and I. Schlotter. "Multivariate complexity analysis of swap bribery". In: *Algorithmica* 64(1) (2012), pp. 126–151 (cited on pp. 153, 155, 156, 161, 175).

[Dwo+01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. "Rank aggregation methods for the web". In: *Proceedings of the 10th International Conference on World Wide Web*. ACM Press, 2001, pp. 613–622 (cited on p. 175).

[EF10] E. Elkind and P. Faliszewski. "Approximation algorithms for campaign management". In: *Proceedings of the 6th International Workshop On Internet And Network Economics (WINE '10)*. Springer, 2010, pp. 473–482 (cited on pp. 153–155, 158, 179).

[EF14] E. Elkind and P. Faliszewski. "Recognizing 1-Euclidean preferences: an alternative approach". In: *Proceedings of the 7th International Symposium on Algorithmic Game Theory (SAGT '14)*. Vol. 8768. Lecture Notes in Computer Science. Springer, 2014, pp. 146–157 (cited on pp. 47, 102).

[EFS09] E. Elkind, P. Faliszewski, and A. Slinko. "Swap bribery". In: *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT '09)*. Vol. 5814. Lecture Notes in Computer Science. Springer, 2009, pp. 299–310 (cited on pp. ii, vi, xi, 6, 152–155, 160, 171, 231, 232).

[EFS12] E. Elkind, P. Faliszewski, and A. Slinko. "Clone structures in voters' preferences". In: *Proceedings of the 13th ACM Conference on Electronic Commerce (EC '12)*. ACM Press, 2012, pp. 496–513 (cited on pp. ix, 37, 41, 73, 76, 96, 148).

[EH08] J. M. Enelow and M. J. Hinich. *Advances in the Spatial Theory of Voting*. Cambridge University Press, 2008 (cited on p. 3).

[EHH15] G. Erdélyi, E. Hemaspaandra, and L. A. Hemaspaandra. "More natural models of electoral control by partition". In: *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT '15)*. Vol. 9346. Lecture Notes in Computer Science. Springer, 2015, pp. 396–413 (cited on pp. 107, 108, 111).

[EK80] J. M. Enelow and D. H. Koehler. "The amendment in legislative strategy: Sophisticated voting in the U.S. Congress". In: *The Journal of Politics* 42(2) (1980), pp. 396–413 (cited on pp. 186, 187, 227).

[EL14] E. Elkind and M. Lackner. "On detecting nearly structured preference profiles". In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '14)*. AAAI Press, 2014, pp. 661–667 (cited on pp. 41, 75, 102).

[EL15] E. Elkind and M. Lackner. "Structure in dichotomous preferences". In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI '15)*. AAAI Press, 2015, pp. 2019–2025 (cited on p. 102).

[Elk+15] E. Elkind, P. Faliszewski, M. Lackner, and S. Obraztsova. "The complexity of recognizing incomplete single-crossing preferences". In: *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*. AAAI Press, 2015, pp. 865–871 (cited on p. 102).

[ELÖ08] B. Escoffier, J. Lang, and M. Öztürk. "Single-peaked consistency and its complexity". In: *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI '08)*. IOS Press, 2008, pp. 366–370 (cited on p. 41).

[ELP13] G. Erdélyi, M. Lackner, and A. Pfandler. "Computational aspects of nearly single-peaked electorates". In: *Proceedings of the 27th AAAI Conference on Artificial Intelligence (AAAI '13)*. AAAI Press, 2013, pp. 283–289 (cited on pp. 74, 75, 88, 100).

[ELP15] G. Erdély, M. Lackner, and A. Pfandler. "Manipulation of k-approval in nearly single-peaked electorates". In: *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT '15)*. Vol. 9246. Lecture Notes in Computer Science. Springer, 2015, pp. 71–85 (cited on p. 76).

[EP98] D. Epple and G. J. Platt. "Equilibrium and local redistribution in an urban economy when households differ in both preferences and incomes". In: *Journal of Urban Economics* 43(1) (1998), pp. 23–51 (cited on p. 26).

[Erd+15] G. Erdélyi, M. R. Fellows, J. Rothe, and L. Schend. "Control complexity in Bucklin and fallback voting: A theoretical analysis". In: *Journal of Computer and System Sciences* 81(4) (2015), pp. 632–660 (cited on pp. 107, 108).

[Fal+09a] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. "Llull and Copeland Voting Computationally Resist Bribery and Constructive Control". In: *Journal of Artificial Intelligence Research* 35(1) (2009), pp. 275–341 (cited on p. 107).

[Fal+09b] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. "Llull and Copeland voting computationally resist bribery and constructive control". In: *Journal of Artificial Intelligence Research* 35 (2009), pp. 275–341 (cited on pp. 153, 158, 175).

[Fal+11] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. "The shield that never was: Societies with single-peaked preferences are more open to manipulation and control". In: *Information and Computation* 209(2) (2011), pp. 89–107 (cited on pp. 72, 76, 107, 108, 142).

[Fal+15] P. Faliszewski, Y. Reisch, J. Rothe, and L. Schend. "Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting". In: *Autonomous Agents and Multi-Agent Systems* 29(6) (2015), pp. 1091–1124 (cited on p. 107).

[Far69] R. Farquharson. *Theory of Voting*. Yale University Press, 1969 (cited on pp. 186, 187).

[Fel+10] 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: *Proceedings of the 30th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '10)*. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, 2010, pp. 459–468 (cited on p. 175).

[FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006 (cited on pp. i, v, 18, 21).

[FH77] S. Földes and P. L. Hammer. "Split graphs". In: 1977, pp. 311–315 (cited on p. 26).

[FHH09] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. "How hard is bribery in elections?" In: *Journal of Artificial Intelligence Research* 35 (2009), pp. 485–532 (cited on pp. 152, 153).

[FHH11a] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. "Multimode control attacks on elections". In: *Journal of Artificial Intelligence Research* 40 (2011), pp. 305–351 (cited on p. 107).

[FHH11b] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. "Multimode control attacks on elections". In: *Journal of Artificial Intelligence Research* 40 (2011), pp. 305–351 (cited on p. 153).

[FHH14] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. "The complexity of manipulative attacks in nearly single-peaked electorates". In: *Artificial Intelligence* 207 (2014), pp. 69–99 (cited on pp. 73, 76, 102, 108).

[FHH15] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. “Weighted electoral control”. In: *Journal of Artificial Intelligence Research* 52 (2015), pp. 507–542 (cited on pp. 107, 108, 114, 141).

[FS11] L. Fortnow and R. Santhanam. “Infeasibility of instance compression and succinct pcps for NP”. In: *Journal of Computer and System Sciences* 77(1) (2011), pp. 91–106 (cited on p. 18).

[FT87] A. Frank and É. Tardos. “An application of simultaneous Diophantine approximation in combinatorial optimization”. In: *Combinatorica* 7(1) (1987), pp. 49–65 (cited on pp. 20, 121, 123, 219).

[Gae09] W. Gaertner. *A Primer in Social Choice Theory: Revised Edition*. Oxford University Press, 2009 (cited on pp. ix, 15).

[Ger+15] E. Gertler, E. Mackin, M. Magdon-Ismail, L. Xia, and Y. Yi. “Computing manipulations of ranking systems”. In: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS ’15)*. IFAAMAS, 2015, pp. 685–693 (cited on p. 153).

[Gib73] A. Gibbard. “Manipulation of voting schemes: a general result”. In: *Econometrica* 41(4) (1973), pp. 587–601 (cited on pp. 3, 15).

[GJ79] M. R. Garey and D. S. Johnson. *Computers and Intractability—A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979 (cited on pp. 16, 19, 82, 92, 171, 220).

[GJS76] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer. “Some simplified NP-complete graph problems”. In: *Theoretical Computer Science* 1(3) (1976), pp. 237–267 (cited on p. 141).

[GNW07] J. Guo, R. Niedermeier, and S. Wernicke. “Parameterized complexity of Vertex Cover variants”. In: *Theory of Computing Systems* 41(3) (2007), pp. 501–520 (cited on p. 142).

[Gol+09] L. Golab, H. Karloff, F. Korn, A. Saha, and D. Srivastava. “Sequential dependencies”. In: *35th International Conference on Very Large Data Bases (PVLDB ’09)*. Vol. 2(1). VLDB Endowment, 2009, pp. 574–585 (cited on pp. 148, 149).

[GR08a] Á. Galambos and V. Reiner. “Acyclic sets of linear orders via the Bruhat orders”. In: *Social Choice and Welfare* 30(2) (2008), pp. 245–264 (cited on p. 29).

[GR08b] Á. Galambos and V. Reiner. “Acyclic sets of linear orders via the Bruhat orders”. In: *Social Choice and Welfare* 30(2) (2008), pp. 245–264 (cited on p. 79).

[Gra78] J.-M. Grandmont. “Intermediate preferences and the majority rule”. In: *Econometrica* 46(2) (1978), pp. 317–330 (cited on p. 26).

[GS96] J. S. Gans and M. Smart. “Majority voting with single-crossing preferences”. In: *Journal of Public Economics* 59(2) (1996), pp. 219–237 (cited on p. 26).

[Haz+12] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. “On the evaluation of election outcomes under uncertainty”. In: *Artificial Intelligence* 189 (2012), pp. 1–18 (cited on p. 188).

[HHR07] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. “Anyone but him: the complexity of precluding an alternative”. In: *Artificial Intelligence* 171(5–6) (2007), pp. 255–285 (cited on pp. 105, 107, 183).

[HKS85] A. J. Hoffman, A. W. J. Kolen, and M. Sakarovitch. “Totally-balanced and greedy matrices”. In: *SIAM Journal on Algebraic and Discrete Methods* 6(4) (1985), pp. 721–730 (cited on p. 26).

[HLM13] L. A. Hemaspaandra, R. Lavaee, and C. Menton. “Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control”. In: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS ’13)*. IFAAMAS, 2013, pp. 1345–1346 (cited on p. 107).

[HM66] F. Harary and L. Moser. “The theory of round robin tournaments”. In: *The American Mathematical Monthly* 73(3) (1966), pp. 231–246 (cited on p. 197).

[Hot29] H. Hotelling. “Stability in competition”. In: *Economic Journal* 39(153) (1929), pp. 41–57 (cited on pp. 39, 40, 230).

[Ina64] K.-i. Inada. “A note on the simple majority decision rule”. In: *Econometrica* 32(32) (1964), pp. 525–531 (cited on pp. 72, 77).

[Ina69] K.-i. Inada. “The simple majority decision rule”. In: *Econometrica* 37(3) (1969), pp. 490–506 (cited on pp. 25, 72, 77).

[IPZ01] R. Impagliazzo, R. Paturi, and F. Zane. “Which problems have strongly exponential complexity?” In: *Journal of Computer and System Sciences* 63(4) (2001), pp. 512–530 (cited on p. 74).

[Jun89] J. P. Jung. “Condorcet consistent binary agendas under incomplete information”. In: *Models of Strategic Choice in Politics*. University of Michigan Press, 1989 (cited on p. 188).

[Kan87] R. Kannan. “Minkowski’s convex body theorem and integer programming”. In: *Mathematics of Operations Research* 12(3) (1987), pp. 415–440 (cited on pp. 20, 121, 123, 219).

[Kar68] S. Karlin. *Total Positivity*. Stanford University Press, 1968 (cited on pp. 26, 72).

[KF16] A. Kaczmarczyk and P. Faliszewski. “Destructive shift bribery”. In: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS ’16)*. IFAAMAS, May 2016, pp. 305–313 (cited on p. 183).

[KL05] K. Konczak and J. Lang. “Voting procedures with incomplete preferences”. In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence (IJCAI ’07)*. AAAI Press, 2005, pp. 124–129 (cited on pp. 153, 188).

[KN12] C. Komusiewicz and R. Niedermeier. “New races in parameterized algorithmics”. In: *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science (MFCS ’12)*. Vol. 7464. Lecture Notes in Computer Science. Springer, 2012, pp. 19–30 (cited on p. 162).

[Kno10] V. Knoblauch. “Recognizing one-dimensional Euclidean preference profiles”. In: *Journal of Mathematical Economics* 46(1) (2010), pp. 1–5 (cited on pp. 47, 102).

[Kom15] C. Komusiewicz. “Kernelization, partially polynomial kernels”. In: *Encyclopedia of Algorithms*. Ed. by M. Kao. Springer, 2015 (cited on p. 165).

[KPS97] B. Klaus, H. Peters, and T. Storcken. “Strategy-proof division of a private good when preferences are single-dipped”. In: *Economics Letters* 55(3) (1997), pp. 339–346 (cited on p. 72).

[Kun06] F.-C. Kung. “An algorithm for stable and equitable coalition structures with public goods”. In: *Journal of Public Economic Theory* 8(3) (2006), pp. 345–355 (cited on p. 26).

[Kur30] K. Kuratowski. “Sur le problème des courbes gauches en topologie”. In: *Fundamenta Mathematicae* 15 (1930), pp. 271–283 (cited on p. 26).

[Lac14] M. Lackner. “Incomplete preferences in single-peaked electorates”. In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI ’14)*. AAAI Press, 2014, pp. 742–748 (cited on p. 102).

[Lan+12] J. Lang, M. S. Pini, F. Rossi, D. Salvagnin, K. B. Venable, and T. Walsh. “Winner determination in voting trees with incomplete preferences and weighted votes”. In: *Autonomous Agents and Multi-Agent Systems* 25(1) (2012), pp. 130–157 (cited on pp. 189, 190, 219, 221).

[LB62] C. G. Lekkerkerker and J. C. Boland. “Representation of finite graphs by a set of intervals on the real line”. In: *Fundamenta Mathematicae* 51 (1962), pp. 45–64 (cited on pp. 4, 26, 69).

[Len83] H. W. Lenstra. “Integer programming with a fixed number of variables”. In: *Mathematics of Operations Research* 8(4) (1983), pp. 538–548 (cited on pp. 20, 115, 116, 121, 123, 176, 219).

[Liu+09] H. Liu, H. Feng, D. Zhu, and J. Luan. “Parameterized computational complexity of control problems in voting systems”. In: *Theoretical Computer Science* 410(27-29) (2009), pp. 2746–2753 (cited on pp. 108, 113).

[LZ10] H. Liu and D. Zhu. “Parameterized complexity of control problems in Maximin election”. In: *Information Processing Letters* 110(10) (2010), pp. 383–388 (cited on p. 108).

[Mar08] D. Marx. “Parameterized complexity and approximation algorithms”. In: *The Computer Journal* 51(1) (2008), pp. 60–78 (cited on p. 158).

[Mat+12] N. Mattei, M. S. Pini, F. Rossi, and K. B. Venable. "Bribery in voting over combinatorial domains is easy". In: *Proceedings of the 12th International Symposium on Artificial Intelligence and Mathematics (ISAIM '12)*. 2012 (cited on p. 109).

[McG53] D. C. McGarvey. "A theorem on the construction of voting paradoxes". In: *Econometrica* 21(4) (1953), pp. 608–610 (cited on pp. 169, 170).

[Mei+08] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. "The complexity of strategic behavior in multi-winner elections". In: *Journal of Artificial Intelligence Research* 33 (2008), pp. 149–178 (cited on p. 107).

[MF14] K. Magiera and P. Faliszewski. "How hard is control in single-crossing elections?" In: *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI '14)*. IOS Press, 2014, pp. 579–584 (cited on pp. 106, 108, 142).

[Mil77] N. R. Miller. "Graph-theoretical approaches to the theory of voting". In: *American Journal of Political Science* 21(4) (1977), pp. 769–803 (cited on pp. 186, 188, 190, 193, 198, 227).

[Mil80] N. R. Miller. "A new solution set for tournaments and majority voting: Further graph-theoretical approaches to the theory of voting". In: *American Journal of Political Science* 24(1) (1980), pp. 68–96 (cited on p. xii).

[Mir71] J. A. Mirrlees. "An exploration in the theory of optimal income taxation". In: *Review of Economic Studies* 38 (1971), pp. 175–208 (cited on pp. 2, 25, 26, 72, 108).

[ML16] V. Menon and K. Larson. "Reinstating combinatorial protections for manipulation and bribery in single-peaked and nearly single-peaked electorates". In: *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI '16)*. AAAI Press, 2016, pp. 565–571 (cited on p. 102).

[MN08] T. Meskanen and H. Nurmi. "Closeness counts in social choice". In: *Power, Freedom, and Voting*. Springer, 2008, pp. 289–306 (cited on p. 76).

[MN78] R. D. McKelvey and R. G. Niemi. "A multistage game representation of sophisticated voting for binary procedures". In: *Journal of Economic Theory* 18(1) (1978), pp. 1–22 (cited on pp. 188, 227).

[Mon09] B. Monjardet. "Acyclic domains of linear orders: A survey". In: *The Mathematics of Preference, Choice and Order*. Springer, 2009, pp. 139–160 (cited on p. 79).

[Mou80] H. Moulin. "On strategy-proofness and single peakedness". In: *Public Choice* 35(4) (1980), pp. 437–455 (cited on pp. 25, 72).

[Mou86] H. Moulin. "Choosing from a tournament". English. In: *Social Choice and Welfare* 3(4) (1986), pp. 271–291 (cited on pp. 188, 189, 227).

[MR81] A. H. Meltzer and S. F. Richard. "A rational theory of the size of government". In: *Journal of Political Economy* 89(5) (1981), pp. 914–927 (cited on p. 26).

[MRS11] T. R. Magrino, R. L. Rivest, and E. Shen. "Computing the margin of victory in IRV elections". In: *2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, (EVT/WOTE '11)*. USENIX Association, 2011 (cited on p. 153).

[MV80] S. Micali and V. V. Vazirani. "An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs". In: *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science (FOCS '80)*. Vol. 21. IEEE Computer Society Press, 1980, pp. 17–27 (cited on p. 137).

[MW13] N. Mattei and T. Walsh. "Preflib: A library for preferences". In: *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT '13)*. Vol. 8176. Lecture Notes in Computer Science. http://preflib.org. Springer, 2013, pp. 259–270 (cited on pp. 101, 222).

[Nie06] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006 (cited on pp. i, v, 18, 21).

[Nur05] H. Nurmi. *Voting Procedures under Uncertainty*. Springer, 2005 (cited on p. 188).

[OP88] P. C. Ordeshook and T. R. Palfrey. "Agendas, strategic voting, and signaling with incomplete information". In: *American Journal of Political Science* 32(2) (1988), pp. 441–466 (cited on p. 188).

[OS87] P. C. Ordeshook and T. Schwartz. "Agendas and the control of political outcomes". In: *The American Political Science Review* 81(1) (1987), pp. 179–200 (cited on pp. 187, 188, 227).

[Osb04] M. J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2004 (cited on p. 227).

[Pap94] C. H. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994 (cited on p. 16).

[Pat71] P. K. Pattanaik. *Voting and Collective Choice*. Cambridge University Press, 1971 (cited on p. 2).

[Pet16] D. Peters. "Recognising multidimensional Euclidean preferences". In: *Proceedings of the 6th International Workshop on Computational Social Choice (COMSOC '16)*. 2016 (cited on p. 70).

[Pin+11] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. "Incompleteness and incomparability in preference aggregation: complexity results". In: *Artificial Intelligence* 175(7-8) (2011), pp. 1272–1289 (cited on pp. 189, 190, 219, 221).

[Poo89] K. T. Poole. *Spatial Models of Parliamentary Voting*. Cambridge University Press, 1989 (cited on p. 3).

[Prz16] T. Przedmojski. "Algorithms and Experiments for (Nearly) Restricted Domains in Elections". Master thesis. MasterThesis. TU Berlin, Feb. 2016 (cited on pp. 75, 101).

[PS15] C. Puppe and A. M. Slinko. *Condorcet domains, median graphs and the single crossing property*. `arXiv:1505.06982v1 [cs.GT]`. 2015 (cited on p. 79).

[PX12] D. Parkes and L. Xia. "A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs". In: *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI '12)*. AAAI Press, 2012, pp. 1429–1435 (cited on p. 107).

[Ras00] B. E. Rasch. "Parliamentary floor voting procedures and agenda setting in Europe". In: *Legislative Studies Quarterly* 25(1) (2000), pp. 3–23 (cited on pp. 186, 192, 193, 231).

[Ras14] B. E. Rasch. "Insincere voting under the successive procedure". In: *Public Choice* 158(3–4) (2014), pp. 499–511 (cited on pp. 188, 227).

[Ras87] B. E. Rasch. "Manipulation and strategic voting in the Norwegian parliament". In: *Public Choice* 52(1) (1987), pp. 57–73 (cited on p. 226).

[Rob77] K. Roberts. "Voting over income tax schedules". In: *Journal of Public Economics* 8(3) (1977), pp. 329–340 (cited on pp. 2, 25, 26, 39, 72, 108).

[Rot15] J. Rothe, ed. *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer, 2015 (cited on p. 21).

[Rot90] P. Rothstein. "Order restricted preferences and majority rule". In: *Social Choice and Welfare* 7(4) (1990), pp. 331–342 (cited on p. 26).

[RPH98] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. "Computationally manageable combinational auctions". In: *Management Science* 44(8) (1998), pp. 1131–1147 (cited on pp. xii, 109).

[San06] T. Sandholm. "Optimal winner determination algorithms". In: *Combinatorial Auctions*. Ed. by P. Cramton, Y. Shoham, and R. Steinberg. MIT Press, 2006. Chap. 14 (cited on p. 109).

[Sap09] A. Saporiti. "Strategy-proofness and single-crossing". In: *Theoretical Economics* 4(2) (2009), pp. 127–163 (cited on p. 26).

[Sat75] M. A. Satterthwaite. "Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions". In: *Journal of Economic Theory* 10(2) (1975), pp. 187–217 (cited on pp. 3, 15).

[Sen66] A. K. Sen. "A possibility theorem on majority decisions". In: *Econometrica* 34(2) (1966), pp. 491–499 (cited on pp. 72, 77, 78).

[Sen70] A. K. Sen. *Collective Choice and Social Welfare*. North-Holland, 1970 (cited on p. 2).

[SFB13] X. Sui, A. Francois-Nienaber, and C. Boutilier. "Multi-dimensional single-peaked consistency and its approximations". In: *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*. AAAI Press, 2013, pp. 375–382 (cited on p. 75).

[SFE15] I. Schlotter, P. Faliszewski, and E. Elkind. “Campaign management under approval-driven voting rules”. In: *Algorithmica* (2015), pp. 1–32 (cited on pp. 154, 161).

[Sko+15] P. Skowron, L. Yu, P. Faliszewski, and E. Elkind. “The complexity of fully proportional representation for single-crossing electorates”. In: *Theoretical Computer Science* 569 (2015), pp. 43–57 (cited on p. 76).

[SP69] A. Sen and P. K. Pattanaik. “Necessary and sufficient conditions for rational choice under majority decision”. In: *Journal of Economic Theory* 1(2) (1969), pp. 178–202 (cited on pp. 72, 77).

[ST06] A. Saporiti and F. Tohmé. “Single-crossing, strategic voting and the median choice rule”. In: *Social Choice and Welfare* 26(2) (2006), pp. 363–383 (cited on pp. 26, 29).

[Sto63] D. E. Stokes. “Spatial models of party competition”. In: *The American Political Science Review* 57(2) (1963), pp. 368–377 (cited on p. 3).

[SW14] A. Slinko and S. White. “Is it ever safe to vote strategically?” In: *Social Choice and Welfare* 43(2) (2014), pp. 403–427 (cited on p. 109).

[SW82] K. A. Shepsle and B. R. Weingast. “Uncovered sets and sophisticated voting outcomes with implications for agenda institutions”. In: *American Journal of Political Science* 28(1) (1982), pp. 49–74 (cited on pp. 188, 227).

[Tid87] T. N. Tideman. “Independence of clones as a criterion for voting rules”. In: *Social Choice and Welfare* 4(3) (1987), pp. 185–206 (cited on p. 74).

[Tov84] C. A. Tovey. “A simplified NP-complete satisfiability problem”. In: *Discrete Applied Mathematics* 8(1) (1984), pp. 85–89 (cited on p. 125).

[Wal07] T. Walsh. “Uncertainty in preference elicitation and aggregation”. In: *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI ’07)*. AAAI Press, 2007, pp. 3–8 (cited on pp. 72, 76, 108, 188).

[Wes77] F. Westhoff. “Existence of equilibria in economies with a local public good”. In: *Journal of Economic Theory* 14(1) (1977), pp. 84–112 (cited on p. 26).

[XC11] L. Xia and V. Conitzer. “Determining possible and necessary winners under common voting rules given partial orders”. In: *Journal of Artificial Intelligence Research* 41 (2011), pp. 25–67 (cited on pp. 153, 189).

[Xia12] L. Xia. “Computing the margin of victory for various voting rules”. In: *Proceedings of the 13th ACM Conference on Electronic Commerce (EC ’12)*. ACM Press, 2012, pp. 982–999 (cited on p. 153).

[Yap83] C.-K. Yap. “Some consequences of non-uniform conditions on uniform classes”. In: *Theoretical Computer Science* 26(3) (1983), pp. 287–300 (cited on p. 18).

[YG14] Y. Yang and J. Guo. “The control complexity of r-approval: from the single-peaked case to the general case”. In: *Proceedings of the 13th International Conference on*

Autonomous Agents and Multiagent Systems (AAMAS '14). IFAAMAS, 2014, pp. 621–628 (cited on p. 74).

[You77] H. P. Young. "Extending Condorcet's rule". In: *Journal of Economic Theory* 16(2) (1977), pp. 335–353 (cited on p. 76).

APPENDIX A

Problem Compendium

For easy reference, we list below the problem definitions of the classical, that is, unparameterized problems that are under consideration in this thesis.

1. AGENDA CONTROL
   Input: A preference profile $\mathscr{P} := (A, V)$ with linear preference orders and a preferred alternative $p \in A$.
   Question: Is there an agenda for A such that p is the overall winner?

2. CLIQUE
   Input: An undirected graph $G = (V(G), E(G))$ and a non-negative integer $h \in \mathbb{N}$.
   Question: Does G admit a size-h *clique*, that is, a size-h vertex subset $U \subseteq V(G)$ such that $G[U]$ is complete?

3. For the two parliamentary voting rules studied in Chapter 8, we have the following manipulation problem.

   COALITIONAL MANIPULATION
   Input: A profile $\mathscr{P} := (A, V)$ with linear preference orders, a preferred alternative $p \in A$, a non-negative integer $k \in \mathbb{N}$, and an agenda $\mathscr{L}$ for A.
   Question: Is it possible to add a set of k voters (a coalition) such that p wins under agenda $\mathscr{L}$?

4. COMBINATORIAL CONSTRUCTIVE CONTROL BY ADDING VOTERS (C-CC-AV)
   Input: A preference profile (A, V) with linear preference orders, a set W of unregistered voters with $V \cap W = \emptyset$ such that each unregistered voter has a linear preference order over A, a bundling function $\kappa\colon W \to 2^W$, a preferred alternative $p \in A$, and a non-negative integer $k \in \mathbb{N}$.
   Question: Is there a subset of voters $W' \subseteq W$ of size at most k such that p is a winner in the profile with voter set $V \cup \kappa(W')$?

5. Dominating Set
   Input: An undirected graph $G = (V(G), E(G))$ and a non-negative integer $h \in \mathbb{N}$.
   Question: Does G admit a *dominating set* of size at most h, that is, a vertex subset $U \subseteq V(G)$ with $|U| \leq h$ such that each vertex from $V(G) \setminus U$ is adjacent to at least one vertex from U?

6. Independent Set
   Input: An undirected graph $G = (U, E)$ and a non-negative integer h.
   Question: Is there an *independent set* of size at least h, that is, a subset of at least h vertices such that no two of them are adjacent to each other?

7. Maximum 2-Satisfiability (Max2Sat)
   Input: A set U of Boolean variables, a collection C of size-two clauses over U and a positive integer h.
   Question: Is there a truth assignment for U which satisfy at least h clauses in C?

8. Partial Vertex Cover (PVC)
   Input: An undirected graph $G = (V(G), E(G))$ and two non-negative integers $h, \ell \in \mathbb{N}$.
   Question: Does G admit a size-h vertex subset $U \subseteq V(G)$ which intersects at least ℓ edges in G?

9. Partition
   Input: A multi-set $X = (x_1, x_2, \ldots, x_r)$ of positive integers.
   Question: Is there a *perfect partition* $X_1 \dot\cup X_2 = X$ of the integers such that both parts sum up to the same value, that is, $\sum_{x \in X_1} x = \sum_{x \in X_2} x$?

10. For $\Pi \in \{$worst-restricted, medium-restricted, best-restricted, value-restricted, single-peaked, single-caved, single-crossing, group-separable, β-restricted$\}$:

    a) Π Alternative Deletion
       Input: A profile with m alternatives and a non-negative integer $k \leq m$.
       Question: Can we delete at most k alternatives so that the resulting profile satisfies the Π-property?

    b) Π Maverick Deletion
       Input: A profile with n voters and a non-negative integer $k \leq n$.
       Question: Can we delete at most k voters so that the resulting profile satisfies the Π-property?

11. POSSIBLE (resp. NECESSARY) WINNER
Input: A preference profile $\mathscr{P} := (A, V)$, a preferred alternative $p \in A$, and a partial agenda $\mathscr{B}$.
Question: Can p win in a (resp. every) completion of the profile $\mathscr{P}$ for an (resp. every) agenda which completes $\mathscr{B}$?

12. $\mathscr{R}$ SHIFT BRIBERY
Input: A profile $\mathscr{P}$ with voter set V, a mapping $\boldsymbol{\pi}$ of SHIFT BRIBERY price functions for V, a preferred alternative $p \in \mathscr{A}(\mathscr{P})$, and a budget bound $B \in \mathbb{N}$.
Question: Is there a shift action $\textsc{Sa}$ with cost $\mathsf{price}(\boldsymbol{\pi}, \textsc{Sa}) \leq B$ and p is an $\mathscr{R}$-winner in $\mathrm{shift}(\mathscr{P}, \textsc{Sa})$?

13. SET COVER
Input: A family $\mathscr{F} = (S_1, \ldots, S_r)$ of sets over a universe $\mathscr{U} = \{u_1, \ldots, u_s\}$ of elements and a non-negative integer $h \geq 0$.
Question: Is there a size-at-most-h *set cover*, that is, a collection $\mathscr{F}'$ of h sets in $\mathscr{F}$ whose union is $\mathscr{U}$?

14. (2-2)-3SAT
Input: A collection $\mathscr{C}$ of clauses over the variable set $\mathscr{X} = \{x_1, \ldots, x_n\}$, where each clause has either two or three literals, and each variable appears exactly four times, twice as a positive literal and twice as a negative literal.
Question: Is there a truth assignment that satisfies all clauses in $\mathscr{C}$?

15. VERTEX COVER
Input: An undirected graph $G = (U, E)$ and an integer $k \leq |U|$.
Question: Is there a *vertex cover* $U' \subseteq U$ of at most k vertices, that is, $|U'| \leq k$ and $\forall e \in E \colon e \cap U' \neq \emptyset$?

Acknowledgment

千言万语
Thousands and thousands of words.
Chinese proverb

First of all, I would like to thank my supervisor Rolf Niedermeier (TU Berlin) for giving me the opportunity to pursue my Ph.D. in his group, for guiding me to shape my scientific life, and for broadening my world view.

I would like to express my sincere gratitude to Edith Elkind (Oxford University) and Jérôme Lang (LAMSADE, Université Paris-Dauphine) for reviewing my thesis and for giving me many helpful tips on writing.

I would also like to thank my former and current colleagues for many fruitful discussions: René van Bevern, Robert Bredereck, Laurent Bulteau, Stefan Fafianie, Vincent Froese, Sven Grottke, Sepp Hartung, Falk Hüffner, Andreas Köpke, Christian Komusiewicz, Stefan Kratsch, André Nichterlein, Rolf Niedermeier, Jan Sablatnig, Ruedi Seiler, Manuel Sorge, Ondřej Suchý, Nimrod Talmon, Anh Quyen Vuong, Mathias Weller, Gerhard J. Woeginger, and Adam Wolisz. I also want to thank my co-authors Noga Alon, Nadja Betzler, Piotr Faliszewski, Jiong Guo, Kirk Pruhs, and Toby Walsh.

Finally, I am deeply in debt to my friends and my family. Without their endless love and tolerance, I would not have been able to finish this thesis.

Schriftenreihe **Foundations of Computing**
Hrsg.: Prof. Dr. Stephan Kreutzer, Prof. Dr. Uwe Nestmann, Prof. Dr. Rolf Niedermeier

ISSN 2199-5249 (print)
ISSN 2199-5257 (online)

01: Bevern, René van: Fixed-Parameter Linear-Time Algorithms for NP-hard Graph and Hypergraph Problems Arising in Industrial Applications. - 2014. - 225 S.
ISBN **978-3-7983-2705-4** (print) EUR **12,00**
ISBN **978-3-7983-2706-1** (online)

02: Nichterlein, André: Degree-Constrained Editing of Small-Degree Graphs. - 2015. - xiv, 225 S.
ISBN **978-3-7983-2705-4** (print) EUR **12,00**
ISBN **978-3-7983-2706-1** (online)

03: Bredereck, Robert: Multivariate Complexity Analysis of Team Management Problems. - 2015. - xix, 228 S.
ISBN **978-3-7983-2764-1** (print) EUR **12,00**
ISBN **978-3-7983-2765-8** (online)

04: Talmon, Nimrod: Algorithmic Aspects of Manipulation and Anonymization in Social Choice and Social Networks. - 2016. - xiv, 275 S.
ISBN **978-3-7983-2804-4** (print) EUR **13,00**
ISBN **978-3-7983-2805-1** (online)

05: Siebertz, Sebastian: Nowhere Dense Classes of Graphs. Characterisations and Algorithmic Meta-Theorems. - 2016. - xxii, 149 S.
ISBN **978-3-7983-2818-1** (print) EUR **11,00**
ISBN **978-3-7983-2819-8** (online)

Universitätsverlag der TU Berlin

Exploiting Structure in Computationally Hard Voting Problems

This thesis explores and exploits structures inherent in voting problems. Some of these structures are found in the preferences of the voters, such as the domain restrictions, which have been widely studied in social choice theory. Others can be expressed as quantifiable measures (or parameters) of the input, which makes them accessible to a parameterized complexity analysis.
The thesis deals with two major topics. The first topic revolves around preference structures, e.g. single-crossing or one-dimensional Euclidean structures. In particular, it shows that single-crossingness can be characterized by finitely many forbidden substructures, and that the same is not possible for one-dimensional Euclideanness. It also studies the computational complexity of making a given preference profile nicely structured by deleting the fewest number of either voters or alternatives. The second topic comprises the parameterized complexity analysis of two computationally hard voting problems: Combinatorial Voter Control and Shift Bribery, making use of some of the structural properties studied in the first part of the thesis. It also investigates questions on the computational complexity, both classical and parameterized, of several voting problems for two widely used parliamentary voting rules.

ISBN 978-3-7983-2825-9 (print)
ISBN 978-3-7983-2826-6 (online)

ISBN 978-3-7983-2825-9

http://verlag.tu-berlin.de