

Computational Complexity in Three Areas of Computational Social Choice: Possible Winners, Unidirectional Covering Sets, and Judgment Aggregation

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der
Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Dorothea Maria Baumeister

aus Duisburg

Düsseldorf, im Juli 2012

Aus dem Institut für Informatik der
Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Referent: Prof. Dr. Jörg Rothe

Koreferent: Prof. Dr. Egon Wanke

Koreferent: Prof. Dr. Ulle Endriss

Tag der mündlichen Prüfung: 19. September 2012

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation eigenständig und ohne unerlaubte Hilfe angefertigt und diese in der vorliegenden oder in ähnlicher Form noch bei keiner anderen Institution eingereicht habe.

Teile dieser Arbeit wurden bereits in den folgenden Schriften veröffentlicht bzw. zur Publikation angenommen: [BBF⁺10, BR10, BER11, BRR11, BFLR12, BR12, BRR⁺12, BEER12b, BEER12a]. Die Arbeit [BBF⁺12] ist vorbehaltlich einer geringfügigen Überarbeitung zur Veröffentlichung angenommen.

Düsseldorf, 16. Juli 2012

Dorothea Baumeister

Acknowledgments

First, I would like to thank my advisor Jörg Rothe. He was always willing to listen and provided answers to all my questions. To all problems he was able to find a solution. He introduced me to scientific research and gave me the opportunity to write this thesis.

I thank Egon Wanke for agreeing to act as a referee for this thesis. His fascinating lectures about graph theory especially attracted my interest that is reflected in my work. Equally I thank Ulle Endriss for being a referee and a member of my dissertation defense committee.

For a good collaboration and many inspiring discussions I would like to thank all my coauthors. Thank you, Felix Brandt, Gábor Erdélyi, Olivia Erdélyi, Piotr Faliszewski, Felix Fischer, Edith Hemaspaandra, Lane Hemaspaandra, Jan Hoffmann, Jérôme Lang, Claudia Lindner, Magnus Roos, Irene Rothe, Jörg Rothe, Lena Schend and Lirong Xia. In addition, many comments from anonymous referees of various conferences and journals improved my work. Furthermore, I would like to thank my colleagues in Düsseldorf, Gábor Erdélyi, Claudia Forstinger, Guido Königstein, Claudia Lindner, Anja Rey, Magnus Roos, and Lena Schend for an intensive exchange of ideas and excellent administrative and technical support. I also want to thank Anja and Lena for proofreading parts of this thesis.

Needles to say that my family and friends always supported me, but I would like to especially thank my father Thomas Baumeister and my sister Barbara Baumeister for proofreading parts of this thesis, and my mother for occasionally taking care of my daughter while I was writing this thesis. I also thank Ariane Dehghan for drawing pictures that I used in several talks and posters.

I am exceptionally grateful to my husband Florian for supporting me in manifold ways, and to my daughter Johanna for her beaming smile.

My research was supported in part by DFG grants RO 1202/11-1, RO 1202/12-1, RO 1202/15-1, the ESF EUROCORES program LogICCC, the SFF grant “Cooperative Normsetting” of Heinrich-Heine-Universität Düsseldorf, and a DAAD grant for a PPP-PROCOPE project.

Zusammenfassung

Diese Arbeit beschäftigt sich mit der Berechnungskomplexität von verschiedenen Problemen aus drei Bereichen der *Computational Social Choice*. Der erste Bereich beschäftigt sich mit Wahlen und speziell dem Problem, zu bestimmen, ob ein ausgewählter Kandidat in einer Wahl mit unvollständiger Information ein Gewinner sein kann. Im zweiten Bereich, der im weiteren Sinne mit dem Problem der Gewinnerbestimmung verwandt ist, wird die Berechnungskomplexität von Problemen bezüglich *minimal upward* und *minimal downward covering sets* untersucht. Das letzte Gebiet ist *Judgment Aggregation*. Hier wird nicht die Komplexität einer Art von „Gewinnerproblem“ untersucht, sondern die dreier Formen von Beeinflussung, nämlich Manipulation, Bestechung und Kontrolle.

Alle untersuchten Probleme kommen aus der *Computational Social Choice*, einem Bereich an der Schnittstelle zwischen Social-Choice-Theorie und Informatik, mit einem bidirektionalen Transfer zwischen beiden Disziplinen. Hier liegt der Schwerpunkt auf der Untersuchung der Berechnungskomplexität von Problemen, die aus der Social-Choice-Theorie kommen. Das zentrale Problem der Gewinnerbestimmung in Wahlen ist in der Praxis wünschenswerterweise in Polynomialzeit zu lösen. Bei dem damit in engem Zusammenhang stehenden *Possible Winner*-Problem stellt sich die Frage, ob eine Wahl, die in einer bestimmten Weise unvollständig angegeben ist, so vervollständigt werden kann, dass ein gewünschter Kandidat gewinnt. Im Gegensatz zur Gewinnerbestimmung ist es meist jedoch nicht erwünscht, dass mögliche Gewinner in Polynomialzeit berechnet werden können, da dies einen Anreiz zur Manipulation des Wahlprozesses geben würde. Der erste Teil dieser Arbeit beschäftigt sich mit der Komplexität von verschiedenen *Possible Winner*-Problemen und liefert Resultate für die Klassen P und NP.

Ebenso verwandt mit dem Gewinnerproblem sind Lösungskonzepte für Dominanzgraphen, wie sie aus einer paarweisen Mehrheitsrelation resultieren können. Ein Lösungskonzept ist eine Möglichkeit, die „beliebtesten“ Elemente eines solchen Dominanzgraphen zu bestimmen. Im zweiten Teil dieser Arbeit wird die Komplexität von verschiedenen Problemen bezüglich sogenannter *upward* und *downward covering sets* untersucht. Hierbei wird Härte und Vollständigkeit nicht nur für NP, sondern auch für die Komplexitätsklassen coNP und Θ_2^P , und Zugehörigkeit zu Σ_2^P gezeigt.

Der letzte Teil dieser Arbeit beschäftigt sich mit *Judgment Aggregation*. Hier werden individuelle Urteile über möglicherweise miteinander verbundene logische Propositionen aggregiert. In dieser Arbeit werden Manipulation, Bestechung und Kontrolle in solchen Prozessen untersucht. Das Manipulationsproblem fragt, ob ein Richter einen Anreiz hat, ein unaufrichtiges Urteil abzugeben; im Bestechungsproblem versucht eine externe Person das Ergebnis durch Bestechung der Richter abzuändern, und in den Kontrollproblemen wird die Menge der teilnehmenden Richter verändert. Hier kann NP-Härte eine Art Schutz gegen diese unerwünschten Arten von Einflussnahme bieten. Zusätzlich zur klassischen Komplexitätstheorie untersuchen wir hier auch die parametrisierte Komplexität und zeigen W[2]-Härte für verschiedene Probleme.

Abstract

This thesis studies the computational complexity of different problems from three areas of computational social choice. The first one is voting, and especially the problem of determining whether a distinguished candidate can be a winner in an election with some kind of incomplete information. The second setting is in the broader sense related to the problem of determining winners. Here the computational complexity of problems related to minimal upward and downward covering sets is investigated. The last area is judgment aggregation. In contrast to the problems mentioned above we do not study the complexity of some kind of “winner” problem, but the complexity of three forms of influencing the outcome, namely manipulation, bribery, and control.

All studied problems come from computational social choice, which is a field at the interface between social choice theory and computer science, with a bidirectional transfer between these two disciplines. We focus on the study of the computational complexity of problems coming from social choice theory. One central problem in social choice is that of winner determination in elections. From a computational point of view it is desirable that the winner can be determined in polynomial time. Associated with this problem is the possible winner problem. Here, the question is whether an election, which is in some sense incompletely specified, can be completed such that a distinguished candidate wins the election. In contrast to the winner problem, it is not always desirable that possible winners can be computed in polynomial time, since this may give incentive to some kind of manipulation in the voting process. The first part of the thesis deals with the complexity of different possible winner problems, and establishes results for the classes P and NP.

Also related to the winner problem in voting are solution concepts for dominance graphs as they may result from a pairwise majority relation. A solution concept is a way of identifying the “most desirable” elements of such dominance graphs. In the second part of this thesis, we study the complexity of various problems related to so-called upward and downward covering sets. We show hardness and completeness not only for NP, but also for the complexity classes coNP and Θ_2^p , and we show membership in Σ_2^p .

The last part of this thesis is concerned with judgment aggregation. Here the task is not to determine a winner, but to aggregate the individual judgment sets over possibly interconnected logical propositions. We study manipulation, bribery, and control in such processes. The manipulation problem asks whether a judge has an incentive to report an untruthful judgment set, in the bribery problem an external actor seeks to change the outcome by bribing some of the judges, and in the control problems the set of participating judges may be changed. Again, this may be undesirable, hence showing NP-hardness can be seen as providing some kind of protection against manipulation, bribery, and control. In addition to classical complexity results, we also study the parameterized complexity and establish W[2]-hardness for various problems.

Contents

Acknowledgments	v
Abstract	ix
1 Introduction	1
1.1 Overview of Computational Social Choice	1
1.2 Outline of the Thesis	2
2 Basics	5
2.1 Elections	5
2.2 Solution Concepts	11
2.3 Judgment Aggregation	13
2.4 Computational Complexity	15
3 Possible Winner	19
3.1 Framework and Basic Definitions	20
3.2 The Possible Winner Problem for Scoring Rules	23
3.2.1 Final Step to a Full Dichotomy	23
3.3 The Possible Winner Problem with Respect to the Addition of New Alternatives for Scoring Rules	27
3.3.1 State of the Art	28
3.3.2 New Results for Scoring Rules	29
3.4 The Possible Winner Problem with Truncated Ballots	34
3.4.1 Results for k -approval	37
3.5 The Possible Winner Problem with Uncertain Weights	40
3.5.1 Results	43
3.6 The Possible Winner Problem under Uncertain Voting System	46
3.6.1 Results	47
3.7 Related Problems	53
3.8 Summary	55
4 Upward and Downward Covering	59
4.1 Definitions and Notation	59
4.2 Results and Discussion	64

4.3	Minimal and Minimum-Size Upward Covering Sets	67
4.3.1	Constructions for Minimal and Minimum-Size Upward Covering Sets	67
4.3.2	Proofs for Minimal and Minimum-Size Upward Covering Sets	77
4.4	Minimal and Minimum-Size Downward Covering Sets	81
4.4.1	Constructions for Minimal and Minimum-Size Downward Covering Sets	81
4.4.2	Proofs for Minimal and Minimum-Size Downward Covering Sets	89
4.5	Summary	92
5	Interference in Judgment Aggregation	93
5.1	Preliminaries	93
5.2	Problem Definitions	99
5.3	Results	103
5.3.1	Manipulation in Judgment Aggregation	103
5.3.2	Bribery in Judgment Aggregation	107
5.3.3	Control in Judgment Aggregation	110
5.4	Conclusions and Future Work	116
	Bibliography	119

1 Introduction

“If we exclude the possibility of interpersonal comparisons of utility, then the only methods of passing from individual tastes to social preferences which will be satisfactory and which will be defined for a wide range of sets of individual orderings are either imposed or dictatorial.”

Nobel prize winner Kenneth J. Arrow [Arr63]

1.1 Overview of Computational Social Choice

The field *Computational Social Choice*, COMSOC for short, is emanating from the combination of social choice theory and computer science.

Social choice theory provides a theoretical framework for collective decision making, such as the aggregation of individual preferences in voting. Computational social choice on the one hand applies techniques from computer science to analyze social choice mechanisms, and on the other hand concepts from social choice are employed in computer science technologies.

The methods studied in social choice originate mostly from political elections, but nowadays elections or, more general, decision making processes are employed in more diverse manners, for example in the aggregation of results from multiple search engines, in the interaction between autonomous software agents or in the allocation of internet bandwidth. To realize such tasks mechanisms originally developed in social choice theory are employed. In such large scale environments a formal mathematical specification and analysis is particularly important and faces us with various problems. To name just a few, for example cheating gets a new dimension, and computational complexity might be a way to protect decision making processes from undesirable interference. Furthermore a huge number of alternatives requires a compact representation of the preferences to reduce communication complexity.

Besides various problems associated with elections computational social choice also studies problems from fair division, judgment aggregation, multiagent resource allocation, auctions, game theory, and many more. The main focus of this thesis is the analysis of the computational complexity of problems from voting, solution concepts, and judgment aggregation.

More information about the evolving field of computational social choice can be found for example in the German book from Rothe et al. [RBLR11], the book chapter from Brandt et al. [BCE12], or the surveys from Chevaleyre et al. [CELM07] and Endriss [End11].

1.2 Outline of the Thesis

This thesis studies the computational complexity of problems from three different areas of computational social choice. Chapter 2 provides the basics of all three fields and briefly introduces to complexity theory. More precisely, Section 2.1 deals with the fundamentals of voting theory needed in Chapter 3, which concerns possible winner problems, including the description of some common voting rules. Solution concepts are introduced in Section 2.2, as a basis for the study of unidirectional covering sets in Chapter 4. Section 2.3 provides some insights in the theory of judgment aggregation, which is studied from a computational point of view in Chapter 5. To conclude Chapter 2, Section 2.4 gives a brief overview of computational complexity.

The first main part of this thesis is Chapter 3, which deals with several possible winner problems. After an introduction the original POSSIBLE WINNER problem (see [KL05]) is defined in Section 3.1 along with basic definitions needed for the study of various possible winner problems. The original possible winner problem for pure scoring rules is then studied in Section 3.2, followed by the study of the possible winner problem with respect to the addition of new alternatives in Section 3.3. Section 3.4 deals with the possible winner problem with truncated ballots. The possible winner problem with uncertain weights is investigated in Section 3.5. Finally, Section 3.6 deals with the possible winner problem under uncertain voting system. The relation of the different possible winner problems to other types of influence on elections, such as bribery and control, is discussed in Section 3.7. This chapter concludes with a short summary of the obtained results and future research directions in Section 3.8.

The second part of this thesis is presented in Chapter 4 and deals with the computational complexity of various upward and downward covering set problems (see [BF08]). In Section 4.1, definitions and notation for unidirectional covering sets are provided. The results are presented in Section 4.2 along with a discussion. The constructions and the proofs used to obtain the results are given in Section 4.3 for minimal and minimum-size upward covering sets and in Section 4.4 for minimal and minimum-size downward covering sets. All results are summarized in Section 4.5.

Chapter 5 forms the third part of this thesis and studies the computational complexity of problems related to judgment aggregation (see [LP09]). After a short introduction to the theoretical framework in Section 5.1, the formal problem

definitions are given in Section 5.2. The obtained results for manipulation, bribery, and control problems in judgment aggregation are presented in Section 5.3. This chapter concludes with a summary and a prospective for future work in Section 5.4.

2 Basics

In this chapter we will provide the basics in the three different fields considered in this thesis: elections, solution concepts, and judgment aggregation. Furthermore a short overview of computational complexity is given. Throughout this thesis, \mathbb{Z} will denote the set of integers, \mathbb{N} the set of positive integers, \mathbb{N}_0 the set of non negative integers, and \mathbb{Q} the set of rationals.

2.1 Elections

We are faced with elections in our everyday life in a variety of ways. Nowadays elections are not only used in the classical sense, like political elections, but also in computational settings, for example in the aggregation of web page rankings and to avoid spam results from web searches [DKNS01, FKS03], or to solve planning problems in multiagent systems [ER93, ER91]. Thus a central topic in social choice are elections, where the task is to aggregate individual preferences in order to obtain a collective outcome.

Formally, an election is a pair (C, V) , where C represents a set of candidates or alternatives and V is a list of voters, represented by their votes. Note that the votes are stored as a list, since different voters may have the same vote. Occasionally, we will denote the votes by $v_1, \dots, v_{|V|}$. To facilitate readability, we will always refer to candidates and voters in the masculine form. Such a pair (C, V) will be referred to as a *preference profile*. The winners of an election are determined by voting rules, which can formally be represented by so-called *social-choice correspondences*. This is a function that assigns an element of the power set of all candidates to every preference profile. For a social-choice correspondence f and a preference profile $P = (C, V)$, $f(P) \subseteq C$ denotes the set of winners. Note that it is also possible that there may be no winner at all. This may be the case in Condorcet elections, see [BTT92] for a definition, for example. Hemaspaandra et al. [HHM12] also give an example from the Baseball hall of fame elections for an election with an empty set of winners. Furthermore it is not requested that there is a single winner. The case where only a single candidate can win the election is modeled by so-called *social-choice functions*, where the domain of the function is the set of all candidates.

The representation of the votes depends on the voting system at hand. In the most common model the voters report their votes as (strict) linear orders over

the set of candidates, where the most liked candidate is on the first position and the most despised candidate on the last position. For example, the preference $a > b > c$ means that the voter strictly prefers a to b , and b to c . The underlying preference relation will be denoted by $>$, and is

- *total* (i.e., $\forall a, b \in C$ it holds that either $a > b$ or $b > a$),
- *transitive* (i.e., $\forall a, b, c \in C$ it follows from $a > b$ and $b > c$ that $a > c$), and
- *asymmetric* (i.e., $\forall a, b \in C$ if $a > b$ it does not hold that $b > a$).

As the voters are represented via their respective vote, we will use the terms *vote*, *voter*, and *ballot* synonymously.

When studying the possible winner problems in Chapter 3, we will mainly focus on elections held under scoring rules and especially k -approval, Copeland, and preference-based approval voting, which will be defined in the following.

The important class of (*positional*) *scoring rules* is for an m -candidate election defined by a scoring vector $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$, where $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m$ and $\alpha_i \in \mathbb{N}_0$, $1 \leq i \leq m$. In an election (C, V) held under the scoring vector $\vec{\alpha}$, a candidate $c \in C$ gets α_i points from a voter $v \in V$ if he is placed at position i in his vote. The overall score of c is the sum of all of his points and will be denoted by $score_{(C,V)}(c)$. The winners of the election are those candidates having the highest score. The scoring rule with vector $\vec{\alpha} = (m-1, m-2, \dots, 1, 0)$ for an election with m candidates is known as Borda count or Borda rule, since it goes back to Borda [Bor81]. Another well-known scoring rule often used in political elections is plurality, where only the first candidate of each vote gets one point. The scoring vector for plurality is then $\vec{\alpha} = (1, 0, \dots, 0)$. In a veto election, which has the scoring vector $\vec{\alpha} = (1, \dots, 1, 0)$, in each vote all candidates except the last one get one point. In Section 3.2, we will also focus on the specific scoring rule with vector $\vec{\alpha} = (2, 1, \dots, 1, 0)$, where the first candidate gets two points, the last candidate gets zero points, and all remaining candidates get one point. It can be assumed without loss of generality that the last entry of a scoring vector is always zero, since if this is not the case for a scoring vector, one can easily transform it into a different one that satisfies it and preserves the result, see [HH07]. We will follow the approach of Betzler and Dorn [BD10] and focus on so-called pure scoring rules. A scoring rule is called *pure* if the scoring vector for an m -candidate election, $m \geq 2$, can be obtained from the scoring vector for an $(m-1)$ -candidate election by inserting one additional value satisfying that $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m$. The following example illustrates different scoring rules.

Example 2.1. *As an example for an election held under scoring rules, assume that the set of candidates is $\{a, b, c, d\}$ and that there are four voters v_1, v_2, v_3 , and v_4 . The preferences and the evaluation for the four above-mentioned scoring rules are shown in Table 2.1. If the election is held under Borda, candidates a, b , and d*

tie for winning. In the case of a plurality election candidate a is the unique winner and if the veto system is used candidates b and d are both winners. Under the specific scoring rule $(2, 1, 1, 0)$ candidates a and b win the election in conjunction.

Table 2.1: Example for the evaluation according to four different scoring rules

Preferences	Points															
	Borda				Plurality				Veto				$(2, 1, 1, 0)$			
	a	b	c	d	a	b	c	d	a	b	c	d	a	b	c	d
$v_1: a > d > b > c$	3	1	0	2	1	0	0	0	1	1	0	1	2	1	0	1
$v_2: b > d > a > c$	1	3	0	2	0	1	0	0	1	1	0	1	1	2	0	1
$v_3: a > d > b > c$	3	1	0	2	1	0	0	0	1	1	0	1	2	1	0	1
$v_4: c > b > d > a$	0	2	3	1	0	0	1	0	0	1	1	1	0	1	2	1
Total points:	7	7	3	7	2	1	1	0	3	4	1	4	5	5	2	4

Another class of voting rules is the family of *Copeland* $^\alpha$ elections. In the system proposed by Copeland [Cop51], the voters report again linear preferences over the set of candidates, but in contrast to the above defined scoring rules the candidates do not receive points independently from each voter but by the result of pairwise comparisons. All candidates are compared pairwise with each other and if a majority of voters prefers one candidate this candidate gets one point. In the case of a tie both voters get half a point. The candidates with the highest overall score are the winners of the election. A slightly different system goes back to Llull (see [HP01]) who proposed a system where the candidates also get one point for a win in such a head-to-head contest, and in case of a tie both candidates get also one point, instead of half a point. This approach was generalized by Faliszewski et al. [FHHR09a] as family of Copeland $^\alpha$ elections, where α is a rational number between zero and one, indicating the number of points both candidates get in case of a tie. Hence, Copeland $^{1/2}$ denotes the original Copeland system and Copeland 1 denotes the system proposed by Llull. For each candidate $c \in C$ we denote by $win(c)$ the number of candidates c beats in a pairwise comparison and by $tie(c)$ the number of candidates c ties with in a pairwise comparison. Then the Copeland $^\alpha$ score of a candidate c equals $win(c) + \alpha \cdot tie(c)$. The winners are again those candidates having the highest score. One important property of Copeland $^\alpha$ elections is that they respect *Condorcet winners*. A candidate is called Condorcet winner if he defeats all candidates in a pairwise comparison. Obviously such a candidate does not always exist but is unique if he exists. It is very natural to require that such a candidate is always a winner, but for example the Borda rule fails to satisfy

this criterion, since even if there is a Condorcet winner another candidate may be chosen as winner by this rule.

Example 2.2. In Table 2.2 we consider the same example as in Table 2.1 with the additional voter v_5 and evaluate the pairwise comparisons between the candidates. The column identifier $a?b$ stands for the head-to-head contest between candidates a and b . Since the number of candidates is odd there are no ties and the winners in all Copeland^α elections coincide. Obviously, candidate d is the winner in all these elections since he wins all pairwise comparisons and hence is even a Condorcet winner.

Table 2.2: Example for pairwise comparisons

Preferences	Pairwise comparisons					
	$a?b$	$a?c$	$a?d$	$b?c$	$b?d$	$c?d$
$v_1: a > d > b > c$	a	a	a	b	d	d
$v_2: b > d > a > c$	b	a	d	b	b	d
$v_3: a > d > b > c$	a	a	a	b	d	d
$v_4: c > b > d > a$	b	c	d	c	b	c
$v_5: d > c > a > b$	a	c	d	c	d	d
Winner:	a	a	d	b	d	d

Even though the concept of evaluating elections through majority in pairwise comparisons is very natural, the famous Condorcet paradox [Con85] shows that this may lead to cyclic common preferences even if the underlying individual preferences are all linear. Consider for example a situation with three candidates a , b , and c , and three voters with preferences $a > b > c$, $b > c > a$, and $c > a > b$. Then a majority of voters prefers a to b and b to c but also c to a .

A voting system where the voters do not report linear preferences is *approval voting*, which was studied by Brams and Fishburn [BF78]. Here the votes are so-called approval vectors. In an m -candidate election, the candidates have a fixed ordering and the voters report a vector from $\{0, 1\}^m$. A one at position i indicates that this voter approves of candidate i , and a zero indicates that this voter disapproves of candidate i . The candidates get one point for each approval and the winners are again all candidates having the maximum score. One variant of approval voting is k -approval, where each voter approves of exactly k candidates. This system can again be modeled as a scoring rule. The voters have to report linear preferences, the first k entries of the scoring vector are one and the remaining entries are zero.

Furthermore, we will consider a voting system that combines preference-based voting and approval voting. Various such systems were proposed by Brams and Sanver [BS06, BS09]. Here the voters do not only report a linear preference, but also an approval line. This line indicates that the voter approves of all candidates to the left of this line and disapproves of all candidates to the right of this line. An additional criterion is that votes must be *admissible* (see [BS06]), that means that each voter must approve of his first ranked candidate and disapprove of his last ranked candidates. The candidates again receive one point for each approval and the candidates with the highest score are the winners. But in contrast to approval vectors here the votes provide more information.

Example 2.3. *An example for an approval and a preference-based approval election is given in Table 2.3. Here the same votes are given in two forms, first as approval vectors for the fixed order (a, b, c, d) of candidates and then for the preference-based approval election as linear orders along with the approval line. The winner of this election is candidate b with 4 points.*

Table 2.3: Example for an approval and a preference-based approval election

	Approval	Preference-based	Points			
	votes	approval votes	a	b	c	d
v_1 :	$(1, 1, 0, 1)$	$a > d > b c$	1	1	0	1
v_2 :	$(0, 1, 0, 0)$	$b d > a > c$	0	1	0	0
v_3 :	$(1, 1, 0, 1)$	$a > d > b c$	1	1	0	1
v_4 :	$(0, 1, 1, 0)$	$c > b d > a$	0	1	1	0
v_5 :	$(1, 0, 1, 1)$	$d > c > a b$	1	0	1	1
Total points:			3	4	2	3

All voting systems that we consider allow for more than one winner. A unique winner can always be obtained by applying a tie-breaking rule. If not stated otherwise we will always assume that no tie-breaking rule is used and hence there may be more than one winner of the election. In contrast to the unique-winner case this setting will be referred to as the co-winner or the nonunique-winner case. For further information on various voting procedures we refer to the book chapter by Brams and Fishburn [BF02].

A good voting system should determine the winners by reasonable criteria. Therefore, a voting rule that always declares the most liked candidate of one specific voter to be the winner, hence a dictatorship, will not be considered as a good voting rule. Social choice theoretic properties like monotonicity, admissibility, unanimity, Pareto efficiency, or independence of irrelevant alternatives try to capture

the intuitive quality of voting rules. An important result in this field is Arrows Impossibility Theorem [Arr63], which says that every preference-based voting rule that satisfies some basic rational criteria must be a dictatorship¹ (see also the quote at the beginning of Chapter 1). Tideman [Tid06] provides an overview of several such social choice theoretic properties and shows which of them are fulfilled by some common voting rules.

It is commonly assumed to be undesirable that a voter reports an insincere preference, but there are cases in which a voter can benefit from strategic voting. Consider for example a Borda election with three candidates a , b , and c , and two voters of the form $a > b > c$ and one voter of the form $c > b > a$. Obviously candidate a wins with four points followed by candidate b who has only three points. The voter with preference $c > b > a$ is totally dissatisfied with this outcome since his most despised candidate is the winner. If he reports the insincere vote $b > c > a$ instead, he will be more satisfied with the outcome since then candidates a and b both have four points and win the election. An election system in which a voter cannot benefit from reporting an insincere preference is called strategy-proof. As shown here the Borda rule is not strategy-proof. A famous result shown independently by Gibbard [Gib73] and Satterthwaite [Sat75] shows that this is by no means only the case for Borda.

Theorem 2.1 (Gibbard [Gib73]; Satterthwaite [Sat75]). *If there are at least three candidates there is no preference-based voting system that fulfills the following properties at the same time:*

- *the voting system returns a unique winner,*
- *for every candidate there is a set of votes that makes him win,*
- *the voting system is strategy-proof, and*
- *the voting system is not a dictatorship.*

Since no candidate should be excluded from the set of winners a priori, all voting systems for at least three candidates that return a unique winner are not strategy-proof, and hence in principle manipulable, or the voting system is a dictatorship. This is rather unsatisfactory since both a dictatorship and manipulable voting rules are undesirable. Bartholdi et al. [BTT89] were the first to show that computational complexity can be used as a barrier against such undesired behavior. The idea is that, though an election is manipulable in principle, computational hardness of the problem determining whether there is a successful manipulative preference can be seen as a resistance against manipulation. More details on the manipulation

¹Arrow originally formulated this theorem in a slightly different formal context, this formulation goes back to Taylor [Tay05, Tay95].

problem are given in Chapter 3.1 and the background from complexity theory in Section 2.4.

The seminal papers by Bartholdi et al. [BTT89, BO91, BTT92] initiated the study of the computational complexity of problems related to elections. Besides the manipulation problem, also the complexity of the problem of determining whether a given candidate is the winner of the election is studied. Obviously it is important for a voting system that the winner can be found in a reasonable amount of time. The third type of problems introduced by Bartholdi et al. [BTT92] is control in elections. They defined different forms of (constructive) control where the chair of the election tries to make some distinguished candidate win by changing the structure of the election. Later the corresponding destructive control problems, where some candidate should be prevented from being a winner, were introduced by Hemaspaandra et al. [HHR07]. Faliszewski et al. [FHH09] initiated the study of bribery in elections.

In Chapter 3 we will focus on the computational complexity of different possible winner problems for elections. In the initial possible winner problem, which was introduced by Konczak and Lang [KL05], the voters report incomplete preferences and the question is whether these can be completed in a way that a desired candidate wins the election. The relationship between the possible winner problems at hand and bribery and control in elections is elaborated in Section 3.7.

More details on algorithmic and computational properties of voting systems can be found for example in the survey from Faliszewski et al. [FHHR09b], in the book chapter from Baumeister et al. [BEH⁺10], and in the German book from Rothe et al. [RBLR11]. The survey from Faliszewski et al. [FHH10] provides more details about how complexity can be employed to protect elections.

2.2 Solution Concepts

A common task in diverse areas of social sciences is to identify the “most desirable” elements of a given set of alternatives based on some binary dominance relation. Examples for applications are the selection of socially desired candidates in social choice (e.g., [Fis77]), the determination of winners in sports tournaments (e.g., [DL99]), and the search for valid arguments in argumentation theory (e.g., [Dun95]). A common dominance relation in social choice is the pairwise majority relation. An alternative x is said to dominate another alternative y if the number of individuals preferring x to y is greater than the number of individuals preferring y to x . Such a pairwise majority relation can be represented as a graph, where the vertices correspond to the alternatives, and there is a directed edge from alternative x to z if x dominates z .

McGarvey [McG53] proved that every asymmetric dominance relation can be realized via a particular profile of linear individual preferences. Consider for ex-

ample the dominance graph (A, \succ) shown in Figure 2.1. The set A consists of the four alternatives a , b , c , and d , and the dominance relation \succ is defined by the following dominances: a dominates c and is dominated by b and d ; additionally, d dominates b and is dominated by c .

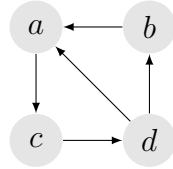


Figure 2.1: Dominance graph (A, \succ)

The profile of eight individual linear preferences shown in Table 2.4 yields exactly the described majority relation. Here, the first line indicates that there are two individuals preferring a to c , c to d , and d to b , which is denoted by the preference $a \succ c \succ d \succ b$. In addition there is one voter with the preference $b \succ c \succ a \succ d$, three voters with the preference $d \succ b \succ a \succ c$, and two voters with the preference $c \succ d \succ b \succ a$.

Table 2.4: Profile of eight individual preferences

#							
2	a	\succ	c	\succ	d	\succ	b
1	b	\succ	c	\succ	a	\succ	d
3	d	\succ	b	\succ	a	\succ	c
2	c	\succ	d	\succ	b	\succ	a

This example also illustrates the famous Condorcet paradox [Con85]. Even if the underlying individual preferences are linear, the majority relation may contain cycles. Here a is preferred to c , c is preferred to d but d is preferred to a in the majority relation. Hence, the natural concept of choosing a maximum element is often not applicable. For the case of nontransitive relations, various alternative solution concepts have been proposed (e.g., [Las97]). Covering relations are transitive subrelations of the dominance relation, and solution concepts based on such covering relations have some desirable properties [Fis77, Mil80, Dut88]. The two covering relations studied here are upward and downward covering relations.

- An alternative a upward covers another alternative b if a dominates b and all alternatives that dominate a also dominate b . Intuitively, a “strongly” dominates b , since there can be no alternative that dominates a but not b .
- An alternative a downward covers another alternative b if a dominates b and all alternatives that are dominated by b are also dominated by a . Here a

“strongly” dominates b , since there can be no alternative that is dominated by b but not by a .

A third natural covering relation, which will not be considered further, is the bidirectional covering, where an alternative a is said to cover b bidirectionally if a covers b upward and downward. In the case of complete dominance relations (which are also called tournaments), all three covering relations coincide.

The solution concepts based on the upward and downward covering relation (see [BF08, Dut88]) are an inclusion-minimal subset of the alternatives that satisfies certain notions of internal and external stability with respect to the upward or downward covering relation, and will be defined formally in Chapter 4. Furthermore the computational complexity of various problems regarding these two solution concepts will be investigated.

2.3 Judgment Aggregation

As the name suggests, judgment aggregation models the situation in trials where judges have to make a common decision on the guiltiness of a defendant. Such decisions do not only occur in trials, but also in other cases where a group has to make a common decision over some possibly interconnected propositions. Consider for example a controversial penalty situation in a soccer match with three referees having different views of the situation. According to the rules one gets a penalty if a player has been fouled in the penalty area. The first referee states that there has been a foul in the penalty area, and hence the team should get a penalty, but the other two referees decide that there is no penalty. The second referee says that what he observed in the penalty area in fact was a dive and the third one claims that there was a foul outside the penalty area. These three individual judgment sets and the evaluation according to the majority rule are shown in Table 2.5. The common decision is that there was a foul in the penalty area, but the penalty will be denied, which is an inconsistent outcome according to the rules.

Table 2.5: Example illustrating the doctrinal paradox

	penalty area	foul	penalty
Referee 1	yes	yes	yes
Referee 2	yes	no	no
Referee 3	no	yes	no
majority	yes	yes	no

The fact that the outcome of a judgment aggregation scenario can be inconsistent even if the underlying individual judgments are all consistent is called *doctrinal paradox* or *discursive dilemma* (see Kornhauser and Sager [KS86] for the original formulation and Pettit [Pet01] for a generalization). The situation that the collective outcome may be inconsistent is by no means special to the majority rule. Indeed List and Pettit [LP02] provide an impossibility result in judgment aggregation showing that there is no judgment aggregation procedure that satisfies some basic natural criteria that always returns complete and consistent outcomes. List and Pettit [LP04] state that the discursive dilemma can be seen as a generalization for the Condorcet paradox (see Section 2.1), since preferences can be seen as a kind of propositions. One way of circumventing the doctrinal paradox is the premise-based approach. Considering again the penalty example above, in the premise-based procedure for the majority rule, we will apply the majority rule only to the premises “foul” and “penalty area”, which results in a “yes” in both cases, and the outcome for the conclusion “penalty” is derived from the outcome of the premises, and hence the penalty will be given.

At first sight the aggregation of judgments and the aggregation of individual preferences seem to be very similar, but there is, however, one major difference. Although both fields are closely related, they consider different settings (for further details, see [LP04, DL07a]). In elections, the individuals report their personal preference over some given alternatives. If two voters have the preferences $a > b$ and $b > a$, this does not contradict, and even if both voters do not comprehend the others voter’s preferences on a and b , they should accept them. In the case of judgment aggregation the situation is different; here the judges report their individual judgment set over some given proposition. Considering again the penalty example, the first and the second referee have different opinions of whether it was a foul or not. Hence both individual judgment sets contradict, and the referees will simply believe that the other one is wrong. There are cases were it might even be possible to objectively determine the truth value of the proposition and decide who is right and who is wrong. In contrast, this would be impossible to say for an individual preference. However, a different line of research tries to implement voting as a truth-tracking mechanism, see for example [CS05].

Strategic behavior as described for voting can also be observed in judgment aggregation. An analogue of the Gibbard-Satterthwaite theorem for judgment aggregation is given by Dietrich and List [DL07c]. As described in Section 2.1 a common approach in computational social choice is to apply methods from theoretical computer science to avoid such undesired strategic behavior, by the corresponding task being a computationally intractable problem. In the case of judgment aggregation this approach was initiated by Endriss et al. [EGP10b], and in Chapter 5 we extend their results on the manipulation problem and additionally initiate the study of bribery and control in judgment aggregation.

For further information on judgment aggregation see for example the surveys

from List and Puppe [LP09] and List [Lis].

2.4 Computational Complexity

After the introduction into three fields of social choice we will now turn to the computer science part of computational social choice and introduce some basic concepts from computational complexity. Computational complexity classifies problems according to resources, e.g., time and space, that are needed to solve them. The computational models that will be used throughout this thesis are deterministic and nondeterministic Turing machines (see [Rot05, Pap95] for a formal definition). One common concept used in the analysis of algorithms is the \mathcal{O} notation.

Definition 2.1. For two functions $f : \mathbb{N} \rightarrow \mathbb{N}$ and $g : \mathbb{N} \rightarrow \mathbb{N}$, it holds that

$$f \in \mathcal{O}(g) \Leftrightarrow \exists c, n_0 \in \mathbb{N} \text{ such that } \forall n \geq n_0 : f(n) \leq c \cdot g(n).$$

Informally stated $f \in \mathcal{O}(g)$ means that f does not grow faster than g . Instead of computing exact running times of algorithms the running time can be estimated by the \mathcal{O} notation.

In what follows we will focus on decision problems that will be encoded as a language $L \subseteq \Sigma^*$ over an alphabet Σ . Two main complexity classes of decision problems are P and NP, where the class P contains all decision problems that can be decided by a deterministic Turing machine in polynomial time, whereas NP is the class of all decision problems that can be decided by a nondeterministic Turing machine in polynomial time. Intuitively P captures the tractable problems, whereas problems that can not be shown to be in P are seen as intractable. It is an open question whether $P = NP$, but it is widely believed that there are problems from NP that are not contained in P. The class coP (coNP, respectively) contains all problems whose complements are contained in P (NP, respectively). The notion of *many-one reduction* shows that one problem is at least as hard as another problem. Here we will mainly focus on polynomial-time many-one reductions that are defined as follows.

Definition 2.2. A is polynomial-time many-one reducible to B (denoted by $A \leq_m^p B$) if and only if there is a polynomial-time computable function $f : \Sigma^* \rightarrow \Sigma^*$ such that for each $x \in \Sigma^*$ it holds that

$$x \in A \Leftrightarrow f(x) \in B.$$

A polynomial-time many-one reduction from A to B implies that B is at least as hard as A . A problem A is called *hard* for a complexity class \mathcal{C} if all problems contained in \mathcal{C} can be reduced to A . If a problem is hard for \mathcal{C} and contained in \mathcal{C} it is called *complete* for \mathcal{C} . Hence the NP-complete problems are the “hardest”

problems from NP, for which it is very unlikely that a polynomial-time algorithm exists.

The first problem that was shown to be NP-complete is the boolean satisfiability problem, that is defined as follows.

SATISFIABILITY (SAT)	
Given:	A boolean formula φ in conjunctive normal form.
Question:	Is there a satisfying truth assignment for φ ?

A boolean formula in $\varphi(v_1, v_2, \dots, v_n) = c_1 \cap c_2 \cap \dots \cap c_r$, over the set of variables $V = \{v_1, v_2, \dots, v_n\}$ is in conjunctive normal form if every clause c_i , $1 \leq i \leq r$, is a disjunction of variables from V . An assignment to the variables in V is satisfying if it evaluates the formula to true.

To show that a problem A is NP-hard, it suffices to show that a known NP-hard problem reduces to A , due to the transitivity of the many-one reduction. To show NP-hardness in the following chapters we will use some set and partition problems as well as some common problems from graph theory. Formally an undirected graph $G = (V, E)$ consists of a finite set of vertices V and a finite set of edges E that are unordered pairs of vertices. Further information on NP-completeness and a compendium of NP-complete problems can be found in [GJ79].

In most cases NP-hardness is bad news due to the practical intractability. For example consider the “winner” problem for elections, where it is asked whether a given candidate is a winner of a given election. Obviously it is desired that this problem is polynomial-time computable. For the voting rules presented in Section 2.1, this is always the case. But we also study problems where NP-hardness is good news. In Section 2.1 the undesirable effect of manipulation in voting is mentioned. A good property of a voting system is if it is NP-hard to determine whether a successful manipulation is possible. When faced with the problem to decide which voting rule, solution concept, or judgment aggregation procedure should be used in a specific situation, the computational complexity of these problems should be taken into account, for example, to prevent strategic behavior. Hence, the study of the computational complexity of various problems associated with voting, solution concepts, and judgment aggregation are important tasks. But it should always be considered that NP-hardness is only the worst-case complexity, and that there may be easy instances though the problem is NP-hard in general.

P and NP are perhaps the most important complexity classes, but there are classes beyond NP. We will concern two specific classes of the polynomial hierarchy over NP (see [MS72]). The first one is $\Sigma_2^p = \text{NP}^{\text{NP}}$; this class contains all problems that are solvable by a nondeterministic Turing machine that has access to an NP oracle. The NP oracle can be seen as a black box that, given an instance for a problem from NP, returns the answer in a single step. The second class we consider

is $P_{\parallel}^{\text{NP}}$ and is equally defined through a Turing machine that has access to an NP oracle. All problems contained in this class can be solved by a deterministic Turing machine that may ask $\mathcal{O}(\log n)$ sequential queries to an NP oracle (see [PZ83]). This class is also known as Θ_2^p (see [KSW87]) that is the closure of NP under polynomial-time truth-table reductions. And it has been shown that this definition coincides with the class of problems that are solvable by a deterministic Turing machine that accesses its NP oracle in a parallel manner (see [Hem87, KSW87]). From the definitions it follows immediately that

$$P \subseteq NP \cap \text{coNP} \subseteq NP \cup \text{coNP} \subseteq \Theta_2^p \subseteq \Sigma_2^p$$

For further information on computational complexity see for example the textbooks [Pap95, Rot05].

In addition to classical complexity we will also investigate the parameterized complexity of problems related to judgment aggregation in Chapter 5. As mentioned before there may be instances for NP-hard problems that are nevertheless easy to solve. Parameterized complexity theory offers a more fine-grained multi-dimensional complexity analysis. If a certain parameter is fixed, an NP-complete problem may be easy (i.e., fixed-parameter tractable) with respect to this parameter. Then, in practice those fixed-parameter tractable problem can be solved efficiently when this parameter is reasonably small, despite its NP-hardness. A *parameterized decision problem* is formally a set $L \subseteq \Sigma^* \times \mathbb{N}$, and we say that it is *fixed-parameter tractable* (FPT) if there is a constant c such that for each input (x, k) of size $n = |(x, k)|$ we can determine in time $\mathcal{O}(f(k) \cdot n^c)$ whether (x, k) is in L , where f is a function depending only on the parameter k . The main hierarchy of parameterized complexity classes is:

$$\text{FPT} = \text{W}[0] \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{W}[\ell] \subseteq \text{XP}.$$

Here we only focus on the class $\text{W}[2]$, which refers to problems that are considered to be fixed-parameter intractable. To show hardness for parameterized problems the following parameterized reduction will be used.

Definition 2.3. *A parameterized reduces to B if each instance (x, k) of A can be transformed in time $\mathcal{O}(g(k) \cdot |x|^c)$ (for some function g and some constant c) into an instance (x', k') of B such that*

$$(x, k) \in A \Leftrightarrow (x', k') \in B,$$

where $k' = g(k)$.

For further details on parameterized complexity theory see the textbooks [DF99, FG06]. Betzler et al. [BBCN12] provide a review of parameterized complexity results in voting theory.

3 Possible Winner

In general, a *possible winner* of an election is a candidate that has, in some kind of incomplete-information election, the possibility to win in a complete extension of the election. In this chapter we will study the computational complexity of several different possible winner problems, where the uncertainty lies in the votes, in the set of participating candidates, in the weights of the votes or in the election system itself. In the case where voters report partial instead of linear preferences, the question whether a distinguished candidate can win in a complete extension is captured by the POSSIBLE WINNER problem, which was first defined by Konczak and Lang [KL05]. For the class of pure scoring rules this problem was studied by Betzler and Dorn [BD10]. Their result was one step away from a full dichotomy since the complexity for one specific scoring rule was left open. In Section 3.2 we prove that the missing case is also NP-complete and so complete a dichotomy result for the important class of pure scoring rules. These results have already been published in [BR10, BR12].

In the original POSSIBLE WINNER problem there is no restriction on the structure of the ballots. One variant of this problem is POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES, here the votes are partial in the sense that the same set of candidates does not occur in all the votes. Obviously this problem is a special case of POSSIBLE WINNER, hence polynomial time algorithms carry over. In Section 3.3 we show that POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES is NP-complete for a whole class of pure scoring rules if one new candidate is added. Furthermore we initiate the study of the weighted version of this problem. These results have already been published in [BRR11].

One further restriction on the form of the ballots are top and/or bottom truncated ballots. If the set of alternatives is too large one might ask the voters to specify only a ranking of their top and/or bottom candidates. POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM-TRUNCATED BALLOTS asks if there is an extension of those ballots into complete ones such that the distinguished candidate wins. We study the computational complexity of these problems in Section 3.4. Since these problems are closely related to the MANIPULATION and the original POSSIBLE WINNER problem, results for POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM-TRUNCATED BALLOTS can be obtained from known results for the just mentioned problems. In addition we prove that POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM-TRUNCATED BALLOTS can be solved in poly-

nomial time for k -approval. The results presented in Section 3.4 have already been published in [BFLR12].

So far we always assumed that the voters who take part in the election are unweighted. However there are situations in which voters do have weights. In Section 3.5 we study the POSSIBLE WINNER WITH UNCERTAIN WEIGHTS problem, where the question is whether the weights of the voters can be set such that a distinguished candidate wins. We study this problem for k -approval and Copeland $^\alpha$ elections and show polynomial-time solvability as well as NP-hardness. These results will appear in [BRR⁺12].

In contrast to the above problems where uncertainty is always associated with the votes, we also study the POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM problem, where the source of uncertainty lies in the voting rule used to aggregate the ballots. In Section 3.6 we show that this problem is NP-complete if the voting rule is chosen from a subclass of the scoring rules. Furthermore we show that it is polynomial-time solvable for preference-based approval voting and the family of Copeland $^\alpha$ elections. These results have already been published in [BRR11].

3.1 Framework and Basic Definitions

As described in Chapter 2.1, in voting theory it is commonly assumed that we have full information about the election and especially that the voters report linear preferences over the set of candidates. But there are situations in which it is reasonable to assume that the voters report only partial preferences. In elections with a high number of candidates it may be too demanding to expect linear preferences from the voters or the voters are unwilling to reveal their whole preference. Consider, for example, the situation in which a committee has to make a joint decision to fill a vacancy. Assume that after some committee members have reported their preferences (and then have gone on vacation), some new candidates apply for the job. In this situation it is very natural to ask whether a distinguished candidate from the initial applicants can get the job if the additional applicants are inserted into the partial votes. Partial preferences also occur in multi-agent systems where there are, for example, too many allocation of resources as if every agent could have a linear order over all of them. And especially for such large-scale elections the computational aspects of the related problems are important. Another reason for assuming partial preferences is that two alternatives may be incomparable for a voter. Given these examples it is natural to assume only partial preferences from the voters when defining computational problems related to voting.

Formally, a partial vote is a (strict) partial order (i.e., a transitive and asymmetric though not necessarily total binary relation) on the set of candidates. In contrast to the preference relation $>$ used for linear votes, we will denote the un-

derlying preference relation for partial votes by \succ . For the candidate set $\{a, b, c\}$ the partial vote $a \succ b$ indicates that a is preferred to b but the relation of a to c and of b to c is unknown. The aim of the original POSSIBLE WINNER problem defined by Konczak and Lang [KL05] is to determine whether the given partial votes can be extended to linear ones such that a desired candidate wins the election. A linear vote v' over a set of candidates C is an extension of a partial vote v over C if $v \subseteq v'$, i.e., for all $a, b \in C$, if $a \succ b$ in v then $a > b$ in v' . For the candidate set $\{a, b, c\}$, the partial vote $a \succ b$ may thus be extended to $a > b > c$, $a > c > b$, or $c > a > b$. A list $V' = (v'_1, v'_2, \dots, v'_n)$ of linear votes is an extension of a list $V = (v_1, v_2, \dots, v_n)$ of partial votes if for every i , $1 \leq i \leq n$, $v'_i \in V'$ is an extension of $v_i \in V$.

As a notational convenience we write for two sets $A, B \subseteq C$ of candidates $A \succ B$ if every candidate in A is preferred to every candidate in B , i.e. $a \succ b$ for all $a \in A$ and $b \in B$, and as a shorthand we write $a \succ B$ for $\{a\} \succ B$ and we write $A \succ b$ for $A \succ \{b\}$. For linear votes we also write $a > \dots > b$ if for all $c \in C \setminus \{a, b\}$ it holds that $a > c$ and $c > b$. For a set $C = D \cup \{c\}$ of candidates the linear vote $c > \vec{D}$ means that c is preferred to all candidates in D , and the candidates in D have a fixed order.

Now we are ready to give the formal definition of the POSSIBLE WINNER problem (see [KL05]) for a given voting system \mathcal{E} .

\mathcal{E} -POSSIBLE WINNER	
Given:	A set C of candidates, a list of votes V that are partial orders over C , and a designated candidate $c \in C$.
Question:	Is there an extension V' of the votes in V to linear orders over C such that c is a winner of election (C, V') under voting system \mathcal{E} ?

Observe that this problem is stated in the co-winner case, since we only ask whether c is a winner of the election. For the unique-winner case, the question is whether there is an extension of the votes such that c is the unique winner of the election. We will use the term \mathcal{E} -POSSIBLE WINNER as generic term for both problems, and denote the explicit variant for the unique-winner case by \mathcal{E} -PW, and for the co-winner case by \mathcal{E} -PCW. If the voting system is clear from the context or not relevant in the corresponding context, we will drop the prefix “ \mathcal{E} –”, and just write POSSIBLE WINNER, PCW, and PW. If it is not explicitly mentioned that we assume unique winners, we always consider the co-winner case.

Unless stated otherwise we will always assume that the number of both candidates and voters is unbounded and that voters are unweighted. For some proofs we will assume succinct representation of the list of votes. That means that there is not one ballot stored for each voter, but a list of ballots with binary integers giving their corresponding multiplicity. For more information on succinct representation for voting systems, see [FHH09].

As mentioned in Chapter 2.1, the famous Gibbard-Satterthwaite Theorem states that in principle every voting system is manipulable, and Bartholdi et al. [BTT89, BO91] proposed to study the computational complexity of the MANIPULATION problem since computational hardness of this problem is a kind of protection against manipulation. Here the question is, if a manipulator can ensure by strategic voting that his favorite candidate wins the election. NP-hardness of this problem implies that no successful manipulation can be found in polynomial time, though the election system is manipulable in principle. Conitzer et al. [CSL07, CS02] generalized this problem by introducing weights to the voters and by allowing a coalition of manipulative voters instead of a single manipulator. The formal definition of the manipulation problem in the unweighted coalitional form for a given voting system \mathcal{E} is as follows.

\mathcal{E} -MANIPULATION	
Given:	A set of candidates C , a list of nonmanipulative votes V that are linear orders over C , a list of manipulative votes W that are not specified yet, and a designated candidate $c \in C$.
Question:	Is there a way to set the votes in W such that c is a winner of election $(C, V \cup W)$ under voting system \mathcal{E} ?

This unweighted coalitional version of the manipulation problem will be denoted by UCM and the version where the weights of all manipulators are known initially in addition to the weights and preferences of the nonmanipulators will be denoted by WCM. A destructive variant of this problem was also introduced by Conitzer et al. [CSL07]. In contrast to the constructive variant, that is stated here, the question is whether the votes in W can be set such that c is not a winner of the election.

The reason for mentioning the MANIPULATION problem is that the POSSIBLE WINNER problem generalizes UCM, as an instance of UCM can be seen as a POSSIBLE WINNER instance in which all nonmanipulative votes are linear orders whereas the manipulative votes are empty, the question is again whether these empty votes can be extended to linear ones such that the designated candidate wins the election. Hence the reduction $\text{UCM} \leq_m^p \text{PCW}$ implies that on one hand hardness results for UCM carry over to PCW and on the other hand easiness results for PCW carry over to UCM.

Complexity results for UCM for various voting systems are due to Faliszewski et al. [FHS08, FHS10], Narodytska et al. [NWX11], Xia et al. [XCP10, XZP⁺09], and Zuckerman et al. [ZPR09, ZLR11]. The results for scoring rules are very sparse and only recently the long standing open problem of the complexity of UCM for Borda elections was solved independently by Betzler et al. [BNW11] and Davies et al. [DKNW11]. They showed NP-completeness even if there are only

two manipulators. For more details on UCM, see the surveys of Faliszewski et al. [FHH10, FP10].

In contrast, the complexity of WCM, the weighted version of the manipulation problem, is well studied (see [CSL07]). In the case of scoring rules even a dichotomy theorem by Hemaspaandra and Hemaspaandra [HH07] is known.

When studying the POSSIBLE WINNER problem it is natural to also ask whether some distinguished candidate is a winner in every extension of the partial votes to linear ones. This problem is called NECESSARY WINNER, and Xia and Conitzer [XC11] showed that it is solvable in polynomial time for all pure scoring rules. In this chapter we will focus only on the POSSIBLE WINNER problem and variants of it.

3.2 The Possible Winner Problem for Scoring Rules

After its introduction by Konczak and Lang [KL05] the POSSIBLE WINNER problem for pure scoring rules was first studied by Xia and Conitzer [XC11] and then by Betzler and Dorn [BD10]. The latter showed that POSSIBLE WINNER is solvable in polynomial time for plurality and veto, and NP-complete for all other pure scoring rules, except the one with the scoring vector $(2, 1, \dots, 1, 0)$, for which the complexity was left open. We will show that POSSIBLE WINNER is NP-complete for the pure scoring rule with the vector $(2, 1, \dots, 1, 0)$, and hence obtain a full dichotomy result for the class of pure scoring rules. Such dichotomy results are particularly important, since they completely characterize the complexity of a whole class of related problems with an easy-to-check condition that distinguishes the easy problems from the hard ones. Schaefer [Sch78] provided the first dichotomy result in computer science by giving a simple criterion to distinguish the easy instances of the satisfiability problem from the hard ones. Hemaspaandra and Hemaspaandra [HH07] established with the “diversity of dislike” criterion the first dichotomy theorem for voting systems. In the following section we will complete the dichotomy result for the POSSIBLE WINNER problem for pure scoring rules.

3.2.1 Final Step to a Full Dichotomy

In this section we will show that POSSIBLE WINNER is NP-complete for the pure scoring rule with the vector $(2, 1, \dots, 1, 0)$. In our proof we will adopt the notion of maximum partial score defined by Betzler and Dorn [BD10]. Fix any scoring rule, and let C be a set of candidates, with a distinguished candidate $c \in C$ which we want to make win the election. The list of votes over C is $V = V^l \cup V^p$, where V^l contains only linear votes and V^p contains partial (i.e., incomplete) votes, such that the score of c is fixed, i.e., for each vote $v \in V^p$, no matter to which linear vote v is extended, the exact number of points c receives from this vote remains

the same. For each candidate $d \in C \setminus \{c\}$, define the *maximum partial score of d with respect to c* (denoted by $s_p^{max}(d, c)$) to be the maximum number of points that d may get, if the partial votes in V^p are extended to linear ones, without defeating c in (C, V') for any extension V' of V to linear votes. Since the score of c is the same in any extension V' of V to linear votes, it holds that

$$s_p^{max}(d, c) = score_{(C, V')}(c) - score_{(C, V')}(d).$$

For the unique winner case each candidate $d \in C \setminus \{c\}$ must have strictly less points than c , hence in this case $s_p^{max}(d, c) = score_{(C, V')}(c) - score_{(C, V')}(d) - 1$ holds.

The notion of maximum partial scores is useful for the following lemma which shows that it is possible to construct a list of linear votes having some desired properties.

Lemma 3.1 (Betzler and Dorn [BD10]). *Let $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ be any scoring rule, let C be a set of $m \geq 2$ candidates with designated candidate $c \in C$, let V^p be a list of partial votes in which the score of c is fixed, and let $s_p^{max}(c', c)$ be the maximum partial score with respect to c for all $c' \in C \setminus \{c\}$. Suppose that the following two properties hold:*

1. *There is a candidate $d \in C \setminus \{c\}$ such that $s_p^{max}(d, c) \geq \alpha_1 |V^p|$.*
2. *For each $c' \in C \setminus \{c\}$, the maximum partial score of c' with respect to c can be written as a linear combination of the score values, $s_p^{max}(c', c) = \sum_{j=1}^m n_j \alpha_j$, with $m = |C|$, $n_j \in \mathbb{N}$, and $\sum_{j=1}^m n_j \leq |V^p|$.*

Then a list V^l of linear votes can be constructed in polynomial time such that for all $c' \in C \setminus \{c\}$, $score_{(C, V^l)}(c') = score_{(C, V^l)}(c) - s_p^{max}(c', c)$, where V^l is an arbitrary extension of V^p to linear votes.

In the following theorem we take the final step to a full dichotomy result using a reduction from the NP-complete problem HITTING SET (see, e.g., [GJ79]), which is defined as follows.

HITTING SET	
Given:	A finite set X , a collection $\mathcal{S} = \{S_1, \dots, S_n\}$ of nonempty subsets of X (i.e., $\emptyset \neq S_i \subseteq X$ for each i , $1 \leq i \leq n$), and a positive integer k .
Question:	Is there a subset $X' \subseteq X$ with $ X' \leq k$ such that X' contains at least one element from each subset in \mathcal{S} ?

Theorem 3.1. *PCW and PW are NP-complete for the pure scoring rule with the vector $(2, 1, \dots, 1, 0)$.*

Proof. Membership in NP is obvious. Our NP-hardness proof uses a reduction from the NP-complete HITTING SET problem. Let (X, \mathcal{S}, k) be a given HITTING SET instance with $X = \{e_1, e_2, \dots, e_m\}$ and $\mathcal{S} = \{S_1, S_2, \dots, S_n\}$. From (X, \mathcal{S}, k) we construct a POSSIBLE WINNER instance with candidate set

$$C = \{c, h\} \cup \{x_i, x_i^1, x_i^2, \dots, x_i^n, y_i^1, y_i^2, \dots, y_i^n, z_i^1, z_i^2, \dots, z_i^n \mid 1 \leq i \leq m\}$$

and designated candidate c . The list of votes $V = V^l \cup V^p$ consists of a list V^l of linear votes and a list V^p of partial votes. $V^p = V_1^p \cup V_2^p \cup V_3^p$ consists of three sublists:

1. V_1^p contains k votes of the form $h \succ C \setminus \{h, x_1, x_2, \dots, x_m\} \succ \{x_1, x_2, \dots, x_m\}$.
2. V_2^p contains the following $2n + 1$ votes for each i , $1 \leq i \leq m$:

$$\begin{aligned} v_i &: h \succ C \setminus \{h, x_i, y_i^1\} \succ \{x_i, y_i^1\}, \\ v_i^j &: y_i^j \succ C \setminus \{y_i^j, z_i^j, h\} \succ h \quad \text{for } 1 \leq j \leq n, \\ w_i^j &: x_i^j \succ C \setminus \{x_i^j, y_i^{j+1}, z_i^j\} \succ y_i^{j+1} \quad \text{for } 1 \leq j \leq n-1 \\ w_i^n &: x_i^n \succ C \setminus \{x_i^n, z_i^n, h\} \succ h. \end{aligned}$$

3. V_3^p contains the vote $T_j \succ C \setminus \{T_j, h\} \succ h$ for each j , $1 \leq j \leq n$, where

$$T_j = \{x_i^j \mid e_i \in S_j\}.$$

For each i , $1 \leq i \leq m$, and j , $1 \leq j \leq n$, the maximum partial scores with respect to c are set as follows:

$$\begin{aligned} s_p^{\max}(x_i, c) &= |V^p| - 1 \\ s_p^{\max}(x_i^j, c) &= |V^p| + 1 \\ s_p^{\max}(y_i^j, c) &= s_p^{\max}(z_i^j) = |V^p| \\ s_p^{\max}(h, c) &\geq 2|V^p|. \end{aligned}$$

This means that each x_i must take at least one last position, which is possible in the votes from V_1^p and the votes v_i , $1 \leq i \leq m$, from V_2^p . Since the candidates x_i^j can never take a last position, they may take at most one first position. For y_i^j and z_i^j , the maximum partial scores with respect to c are set such that for each first position they take, they must also take at least one last position. Finally, h can never beat c . By Lemma 3.1, we can construct a list of votes V^l such that all candidates other than c can get only their maximum partial scores with respect to c in the partial votes.

We claim that (X, \mathcal{S}, k) is a yes-instance of HITTING SET if and only if c is a possible winner in (C, V) , using the scoring rule with vector $(2, 1, \dots, 1, 0)$.

From left to right, suppose there exists a hitting set $X' \subseteq X$ with $|X'| \leq k$ for \mathcal{S} . The partial votes in V^p can then be extended to linear votes such that c wins the election as follows:

	$e_i \in X'$	$e_i \notin X'$
V_1^p :	$h > \dots > x_i$	
V_2^p :	v_i : $h > \dots > x_i > y_i^1$	$h > \dots > y_i^1 > x_i$
	$v_i^j, 1 \leq j \leq n$: $y_i^j > \dots > z_i^j$	$z_i^j > y_i^j > \dots > h$
	$w_i^j, 1 \leq j \leq n$: $z_i^j > x_i^j > \dots > y_i^{j+1}$	$x_i^j > \dots > y_i^{j+1} > z_i^j$
	w_i^n : $z_i^n > x_i^n > \dots > h$	$x_i^n > \dots > h > z_i^n$
V_3^p :	$x_i^j > \dots > h$ for some $j \in \{\ell \mid e_i \in S_\ell\}$.	

Every x_i takes one last position and get his maximum partial score with respect to c . For $e_i \in X'$, all y_i^j take exactly one first, one last, and a middle position in all remaining votes. For $e_i \notin X'$, all y_i^j take middle positions only. So they always get their maximum partial scores with respect to c . The candidates z_i^j also get their maximum partial scores with respect to c , since they always get one first position, one last position, and a middle position in all remaining votes. Every candidate x_i^j gets at most one first position and therefore does not exceed his maximum partial score with respect to c . Since no candidate exceeds his maximum partial score with respect to c , candidate c is a winner in this extension of the list V^p of partial votes.

Conversely, assume that c is a possible winner for (C, V) . Then no candidate may get more points in V^p than his maximum partial score with respect to c . Since at most k different x_i may take a last position in V_1^p , at least $n - k$ different x_i must take a last position in v_i . Fix any i such that x_i is ranked last in v_i . We now show that it is not possible that a candidate x_i^j then takes a first position in any vote of V_3^p . Since x_i takes the last position in v_i , candidate y_i^1 takes a middle position in this vote and gets one point. The only vote in which the score of y_i^1 is not fixed is v_i^1 . Without the points from this vote, y_i^1 already gets $|V^p| - 1$ points, so y_i^1 cannot get two points in v_i^1 , and z_i^1 takes the first position in v_i^1 . Without the points from w_i^1 , z_i^1 gets $|V^p|$ points and must take the last position in w_i^1 . The first position in w_i^1 is then taken by x_i^1 , so x_i^1 cannot take a first position in any vote from V_3^p . Candidate y_i^2 gets one point in w_i^1 , and by a similar argument as above, x_i^2 is placed at the first position in w_i^2 . Repeating this argument, we have that for each j , $1 \leq j \leq n$, x_i^j is placed at the first position in w_i^j and thus cannot take a first position in a vote from V_3^p . This means that all first positions in the votes of V_3^p must be taken by those x_i^j for which x_i takes the last position in a vote from V_1^p . This is possible only if the x_i^j are not at the first position in w_i^j . Thus z_i^j must take this position. Due to z_i^j 's maximum partial score with respect to c , this is possible only if z_i^j takes the last position in v_i^j . Then y_i^j takes the first position in this vote. This is possible, since y_i^j can take a middle position in v_i for $j = 1$, and in v_i^j for $2 \leq j \leq n$. Hence all x_i^j , where x_i takes the last position in the votes of V_1^p , may take the first position in the votes of V_3^p . Thus, by the definition of V_3^p (which, recall, contains the vote $T_j \succ C \setminus \{T_j, h\} \succ h$ for each j , $1 \leq j \leq n$, where

$T_j = \{x_i^j \mid e_i \in S_j\}$, the elements e_i corresponding to those x_i must form a hitting set of size at most k for \mathcal{S} .

Note that this proof holds for the co-winner case and the unique-winner case at the same time due to the definition of the maximum partial scores as argued by Betzler and Dorn [BD10]. \square

This completes the dichotomy for the POSSIBLE WINNER problem in pure scoring rules with the result that it is NP-complete for all pure scoring rules except plurality and veto.

3.3 The Possible Winner Problem with Respect to the Addition of New Alternatives for Scoring Rules

This section deals with a variant of the POSSIBLE WINNER problem, called POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES, that was introduced by Chevaleyre et al. [CLMM10]. As the name suggests, this problem captures the situation where some additional candidates enter the election after the votes have already been cast. Chevaleyre et al. [CLMM10] and Xia et al. [XLM11] argue that such situations often occur in real-life, for example if a meeting has to be scheduled, and after the participants reported their preferences over the given dates a new time-slot becomes available. The POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES problem asks if the given linear preferences over a set of initial candidates can be extended to linear preferences over the initial and the new candidates such that a distinguished candidate from the set of initial candidates wins the election. Here we will use the terms alternatives and candidates interchangeably. The formal definition of this problem for a given voting system \mathcal{E} is as follows.

\mathcal{E} -POSSIBLE WINNER W.R.T. THE ADDITION OF NEW ALTERNATIVES

- Given:** A set C of candidates, a list of votes V that are linear orders over C , a set C' with $|C'| = k$, $k \in \mathbb{N}$, of new candidates, and a designated candidate $c \in C$.
- Question:** Is there an extension V' of the votes in V to linear orders over $C' \cup C$ such that c is a winner of election $(C \cup C', V')$ under voting system \mathcal{E} ?
-

As for the POSSIBLE WINNER problem we will drop the prefix “ \mathcal{E} -” if appropriate and we denote the explicit co-winner variant by PCWNA and the unique-winner variant by PWNA. In contrast to the POSSIBLE WINNER problem we will also study PCWNA for weighted elections. Here every voter $v_i \in V$ has a predefined

weight $w_i \in \mathbb{N}$ and for the evaluation of the election this is counted as if there were w_i unweighted votes of type v_i .

3.3.1 State of the Art

The complexity of PCWNA has been studied for different voting rules by Chevaleyre et al. [CLMM10, CLM⁺12] and Xia et al. [XLM11]. Since we focus on pure scoring rules, Table 3.1 summarizes the known results for pure scoring rules from earlier work [CLMM10, CLM⁺12, XLM11]. It is again always assumed that the voters are unweighted and that the number of initial candidates is unbounded, unless stated otherwise.

Table 3.1: Previous results on the complexity of PCWNA for pure scoring rules

Scoring rule	PCWNA
Plurality	in P (see [CLMM10])
Veto	in P (see [CLMM10])
Borda	in P (see [CLMM10])
2-Approval	in P (see [CLM ⁺ 12])
k -Approval, $ C' \leq 2$	in P (see [CLMM10, CLM ⁺ 12])
k -Approval, $k \geq 3$, $ C' \geq 3$	NP-complete (see [CLMM10, CLM ⁺ 12])
$(\alpha_i - \alpha_{i+1}) \leq (\alpha_{i+1} - \alpha_{i+2})$, $1 \leq i \leq m - 2$	in P (see [CLMM10])
$(3, 2, 1, 0, \dots, 0)$, $ C' = 1$	NP-complete (see [CLMM10])

In particular, PCWNA is in P for the Borda rule for any fixed number of candidates, yet it is NP-complete for the scoring vector $(3, 2, 1, 0, \dots, 0)$ when the number of candidates is unbounded. Thus, this NP-completeness result is about a more general problem and does not contradict the polynomial-time solvability of Borda in the restricted case of four candidates. Chevaleyre et al. [CLMM10] raised the question whether there are more general results for the class of pure scoring rules for PCWNA as it is the case for PCW (see the dichotomy result by Betzler and Dorn [BD10] and Baumeister and Rothe [BR12] which is partly presented in Section 3.2). In the next section we will make one step further in this direction by showing NP-completeness for a whole class of pure scoring rules if one new candidate is added. Additionally, we initiate the study of the weighted PCWNA problem. Note that since PCWNA is a special case of PCW, no hardness results carry over, but the membership in P for PCWNA under plurality and veto directly follows from the corresponding PCW problems.

3.3.2 New Results for Scoring Rules

Unweighted Voters We will extend the result of Chevaleyre et al. [CLMM10] that PCWNA is NP-complete for pure scoring rules with vector $(3, 2, 1, 0, \dots, 0)$ when one new candidate is added by showing that NP-completeness of PCWNA holds even for the class of pure scoring rules of the form $(\alpha_1, \alpha_2, 1, 0, \dots, 0)$ with $\alpha_1 > \alpha_2 > 1$, if one new candidate is added. The reduction will be from the NP-complete problem THREE-DIMENSIONAL MATCHING (see, e.g., [GJ79]) which is defined as follows.

THREE-DIMENSIONAL MATCHING (3-DM)	
Given:	A set $M \subseteq W \times X \times Y$, with $W = \{w_1, \dots, w_q\}$, $X = \{x_1, \dots, x_q\}$, and $Y = \{y_1, \dots, y_q\}$.
Question:	Is there a subset $M' \subseteq M$ with $ M' = q$, such that no two elements of M' agree in any coordinate?

Theorem 3.2. *PCWNA is NP-complete for pure scoring rules of the form $(\alpha_1, \alpha_2, 1, 0, \dots, 0)$ with $\alpha_1 > \alpha_2 > 1$, if one new candidate is added.*

Proof. Membership in NP is obvious and the proof of NP-hardness is by a reduction from the NP-complete 3-DM problem. Let $M \subseteq W' \times X' \times Y'$ be an instance of 3-DM with $W' = \{w'_1, \dots, w'_q\}$, $X' = \{x'_1, \dots, x'_q\}$, and $Y' = \{y'_1, \dots, y'_q\}$, where $m = |M|$. Let $p(s)$ be the number of elements in M in which $s \in W' \times X' \times Y'$ occurs.

Construct an instance of the PCWNA problem with the election (C, V) having the set $C = W \cup X \cup Y \cup \{b, c\} \cup D$ of candidates with $W = \{w_1, \dots, w_q\}$, $X = \{x_1, \dots, x_q\}$, and $Y = \{y_1, \dots, y_q\}$. The new candidate to be added is a , so $C' = \{a\}$. D contains only dummy candidates needed to pad the votes so as to make the reduction work. Table 3.2 shows the list $V = V_1 \cup V_2 \cup V_3 \cup V_4$ of votes. Note that only the first three candidates of each vote will be specified, since all other candidates do not receive any points. The numbers behind each vote denote their multiplicity. All places that need to be filled by a dummy candidate will be indicated by d (with no explicit subscript defined). Note that it is possible to substitute the d 's by a polynomial number of dummy candidates such that none of them receives more than $q \cdot \alpha_1$ points.

The scores of the single candidates in the election (C, V) are:

$$\begin{aligned}
 \text{score}_{(C,V)}(c) &= (q + m)\alpha_1 + \alpha_2 \\
 \text{score}_{(C,V)}(w_i) &= (q + m + 1)\alpha_1, \quad 1 \leq i \leq q \\
 \text{score}_{(C,V)}(x_i) &= (q + m)\alpha_1 + 2\alpha_2 - 1, \quad 1 \leq i \leq q \\
 \text{score}_{(C,V)}(y_i) &= (q + m)\alpha_1 + \alpha_2 + 1, \quad 1 \leq i \leq q \\
 \text{score}_{(C,V)}(b) &= (q + 2m)\alpha_1 + 2\alpha_2 \\
 \text{score}_{(C,V)}(d) &< (q + m)\alpha_1 + \alpha_2, \quad \forall d \in D
 \end{aligned}$$

Table 3.2: Construction for the proof of Theorem 3.2

V_1	$w_i > x_j > y_k > \dots$	$1, \forall (w'_i, x'_j, y'_k) \in M$
V_2	$w_i > d > d > \dots$ $d > d > x_i > \dots$ $d > d > y_i > \dots$	$q + m + 1 - p(w'_i), \forall w_i \in W$ $(q + m)\alpha_1 + (2 - p(x'_i))\alpha_2 - 1, \forall x_i \in X$ $(q + m)\alpha_1 + \alpha_2 + 1 - p(y'_i), \forall y_i \in Y$
V_3	$c > d > d > \dots$ $d > c > d > \dots$	$q + m$ 1
V_4	$d > d > b > \dots$	$(q + 2m)\alpha_1 + 2\alpha_2$

Note that $score_{(C,V)}(d) < score_{(C,V)}(c)$ for all dummy candidates $d \in D$.

We claim that c is a possible winner (i.e., a can be inserted such that c wins in the election held over the candidates $C \cup C'$) if and only if there is a matching M' for the 3-DM instance M .

For the direction from right to left assume that there exists a matching M' for M . Extend the votes in V to V' , where a is inserted at a position with zero points in all votes of V_2 and V_3 , and the votes in V_1 and V_4 are extended as follows:

V_1 :	$a > w_i > x_j > y_k$	$1, \forall (w'_i, x'_j, y'_k) \in M'$
	$w_i > x_j > y_k > a$	$1, \forall (w'_i, x'_j, y'_k) \in M \setminus M'$
V_4 :	$d > d > a > b$	$m\alpha_1 + \alpha_2$
	$d > d > b > a$	$(q + m)\alpha_1 + \alpha_2$

Then all candidates except the dummy candidates have exactly $(q + m)\alpha_1 + \alpha_2$ points. Hence c has the highest score and is a winner of the election.

Conversely assume that c is a winner of the election $(C \cup C', V')$, where V' is an extension of the linear votes in V . This implies that the scores of all other candidates in this election are less than or equal to the score of c . The score of c will always be $(q + m)\alpha_1 + \alpha_2$, since c gets all of his points from the voters in V_3 , where he is placed at the top position in $m + q$ votes and at a second position in one vote.

Since $score_{(C,V)}(w_i) = (q + m + 1)\alpha_1$, each of the candidates w_i , $1 \leq i \leq q$, must lose at least $\alpha_1 - \alpha_2$ points when inserting a . Due to the requirement that $\alpha_1 > \alpha_2$, each w_i has to take at least one second position in a vote where he was ranked first originally. For the candidates x_i , $1 \leq i \leq q$, we have $score_{(C,V)}(x_i) = (q + m)\alpha_1 + 2\alpha_2 - 1$. Again, since $\alpha_2 > 1$, each x_i must lose at least $\alpha_2 - 1$ points, and since $score_{(C,V)}(y_i) = (q + m)\alpha_1 + \alpha_2$, each y_i must lose at least one point so as to not beat c .

The new candidate a can get at most $(q + m)\alpha_1 + \alpha_2$ points, since otherwise a would beat c .

To prevent w_i , $1 \leq i \leq q$, from beating c , a must be placed in a first position in q votes from V_1 or V_2 . Then a can get at most $m\alpha_1 + \alpha_2$ points from the remaining votes without beating c . In the current situation b would beat c by $m\alpha_1 + \alpha_2$ points. So a must take $m\alpha_1 + \alpha_2$ third positions in these votes such that b has a score of $(q + m)\alpha_1 + \alpha_2$. Then the score of a is $(q + m)\alpha_1 + \alpha_2$. Since we assumed that c is a winner of the election, every x_i , $1 \leq i \leq q$, must end up having $\alpha_2 - 1$ points less, and every y_i , $1 \leq i \leq q$, must end up having one point less. This is possible only if a is at the first position in some vote from V_1 . Hence the q first positions of a must shift every candidate x_i and y_i by one position to the right. Then the triples corresponding to the three elements w_i , x_j , and y_k corresponding to these q votes must form a matching for the 3-DM instance M . \square

Weighted Voters Now we will study the PCWNA problem in the case of weighted voters. Obviously all NP-hardness results from the unweighted case carry over to the weighted case. But there is no direct transfer from the polynomial-time algorithms from the unweighted case to the weighted case. In particular we will show that for some voting rules where the problem is in P for unweighted voters, it is NP-complete for weighted voters. Specifically, we will consider the plurality rule for weighted voters in this section. For this rule, polynomial-time algorithms are known for the original POSSIBLE WINNER problem in the case of unweighted voters (see [BD10]), and for MANIPULATION both in the unweighted-voters and in the weighted-voters case (see [CSL07, CS02]). In contrast, we now show that PCWNA is NP-complete for plurality in the case of weighted voters, even if there are only two initial candidates and one new candidate to be added. The proof will be by a reduction from the NP-complete PARTITION problem (see, e.g., [GJ79]).

PARTITION

Given: A nonempty, finite sequence (s_1, s_2, \dots, s_n) of positive integers.

Question: Is there a subset $A' \subset A = \{1, 2, \dots, n\}$ such that

$$\sum_{i \in A'} s_i = \sum_{i \in A \setminus A'} s_i?$$

Theorem 3.3. *PCWNA is NP-complete for plurality in the case of weighted voters, even if there are only two initial candidates and one new candidate to be added.*

Proof. Membership in NP is obvious. To show NP-hardness of PCWNA for plurality in the case of weighted voters, we now give a reduction from the NP-complete PARTITION problem. For a given PARTITION instance (s_1, s_2, \dots, s_n) , let

$\sum_{i \in A} s_i = 2K$, where $A = \{1, 2, \dots, n\}$. We construct an election (C, V) with the set of candidates $C = \{c, d\}$, where c is the distinguished candidate and the list of votes $V = V_1 \cup V_2$ with the corresponding weights as follows:

V_1 : $c > d$ one vote of weight K
V_2 : $d > c$ one vote of weight s_i for each $i \in A$

The new candidate to be added is a , so $C' = \{a\}$. In the initial situation, the score of candidate c is K , and candidate d receives $2K$ points and hence wins the election. We now show that c can be made a winner by introducing candidate a into the election if and only if there is a valid partition for the given PARTITION instance.

For the direction from right to left assume that there is a subset $A' \subset A$ such that $\sum_{i \in A'} s_i = \sum_{i \in A \setminus A'} s_i$. If the new candidate a is placed at the first position in each of those votes from V_2 that correspond to the $i \in A'$, and at the last position in all remaining votes, then the score of all three candidates is exactly K , and c is a co-winner of the election.

Conversely assume that c is a winner of the election, after candidate a has been introduced. It must hold that candidates a and d receive at most K points, since c gets K points from the vote in V_1 . Hence candidate d must lose K points due to inserting candidate a . This is possible only if a is placed at the first position in some votes from V_2 with a total weight of K . These votes now correspond to a valid partition. □

Next, we study 2-approval and in Theorem 3.4 we give a result for the case of weighted voters and an unbounded number of candidates.

Theorem 3.4. *PCWNA is NP-complete for 2-approval in the case of weighted voters, where the number of candidates is unbounded and one new candidate is to be added.*

Proof. To prove NP-hardness of the problem, we again give a reduction from PARTITION. Let (s_1, \dots, s_n) be an instance of PARTITION with $\sum_{i \in A} s_i = 2K$, where $A = \{1, 2, \dots, n\}$.

We introduce a set C of $n + 3$ candidates:

- c (the candidate we want to win the election),
- b (the candidate who wins the original election), and
- a set $\{d_0, d_1, \dots, d_n\}$ of dummy candidates.

The votes are as follows. Note that we specify only the first two candidates, since in 2-approval the ranking of the remaining candidates cannot influence the outcome of the election.

- For each s_j , we define a vote $d_j > b > \dots$ with weight s_j .
- There is one vote $c > d_0 > \dots$ with weight K .

Since $\sum_{j \in A} s_j = 2K$, candidate b has a score of $2K$ and wins the election. We now prove that c can be made a winner by adding one new candidate, namely a , if and only if there is a subset $A' \subset A$ that induces a valid partition for the given instance.

For the direction from right to left assume that we have a partition $A' \subset A$. By putting a at the first position of each vote having a weight of s_i and for which $i \in A'$, a will get exactly K points. Furthermore, b loses these K points, since he moves to the third position in these votes. Now, there is a tie between a , b , c , and d_0 , each having K points. Since $s_j \leq K$, $1 \leq j \leq n$, no candidate d_j , $1 \leq j \leq n$, has a higher score. Thus, c is a co-winner of the election.

For the direction from left to right assume that c can be made a winner by adding candidate a . It follows that b has to lose at least K points. Hence a has to be added in the votes of the form $d_j > b > \dots$ at first or second position. Thus, a gets each point that b loses. But since c is made a winner by inserting a , the new candidate a can get no more than K points. Therefore, we have to insert a in a subset of votes such that the weights of these votes sum up to exactly K . Consequently there exists a partition.

Since PARTITION is NP-complete, this proves NP-hardness. Membership in NP is straightforward. Thus PCWNA is NP-complete for 2-approval. \square

It is easy to see that the proof of Theorem 3.4 can be transferred to k -approval: In each vote $k - 2$ dummy candidates are added at the first $k - 2$ positions, which gives a total number of $(k - 1)(n + 1) + 2$ initial candidates and one new candidate. Thus we can state the following corollary.

Corollary 3.1. *PCWNA is NP-complete for k -approval in the case of weighted voters where the number of candidates is unbounded and one new candidate is to be added.*

Note that, in Corollary 3.1, the k in k -approval cannot depend on the number of candidates, since the proof is for an *unbounded* number of candidates. The results of this section for PCWNA in the case of weighted voters are summarized in Table 3.3.

Table 3.3: New results on the complexity of PCWNA in the case of weighted voters

Scoring rule	PCWNA
Plurality, $ C = 2$, $ C' = 1$	NP-complete (see Theorem 3.3)
k -Approval, $ C' = 1$	NP-complete (see Corollary 3.1)

3.4 The Possible Winner Problem with Truncated Ballots

In this section we consider a possible winner problem where the voters again report partial instead of linear votes but the partial votes all have a common structure, they are either *doubly*, *top*, or *bottom* truncated. For two nonnegative integers t and b we say that a partial preference order \succ on a set of candidates C is a (t, b) -doubly-truncated vote, if there is a permutation π over $\{1, \dots, |C|\}$ such that \succ is of the form $c_{\pi(1)} \succ \dots \succ c_{\pi(t)} \succ \{c_{\pi(t+1)}, \dots, c_{\pi(m-b)}\} \succ c_{\pi(m-b+1)} \succ \dots \succ c_{\pi(m)}$. Hence it holds that each candidate in the set $\{c_{\pi(t+1)}, \dots, c_{\pi(m-b)}\}$ is ranked strictly below candidate $c_{\pi(t)}$, and strictly above $c_{\pi(m-b+1)}$, but the voter is indifferent among the members of the set. We refer to the candidates $c_{\pi(1)}, \dots, c_{\pi(t)}, c_{\pi(m-b+1)} \dots c_{\pi(m)}$ as *ranked* candidates and to the remaining ones as *unranked* candidates. Furthermore, we define $top(\succ) = t$ to be the number of candidates ranked at the top of a vote, and $bottom(\succ) = b$ to be the number of candidates ranked at the bottom of a vote. A top-truncated vote is a preference order that has only top-ranked candidates but no bottom ranked candidates, hence it is a $(t, 0)$ -doubly-truncated vote. And accordingly a bottom-truncated vote is $(0, b)$ -doubly-truncated and has only bottom-ranked candidates. We call a preference order *doubly-truncated* (*top-truncated*, *bottom-truncated*) if there are values t and b for which it is (t, b) -doubly-truncated (t -top-truncated, b -bottom-truncated). Accordingly an election (C, V) is doubly-truncated (top-truncated, bottom-truncated) if each vote $v \in V$ is doubly-truncated (top-truncated, bottom-truncated). Furthermore, we consider a variant where there is either an upper or a lower bound on the number of ranked candidates.

The motivation to consider these special forms of ballots is that it is a reasonable assumption that voters are able to rank some of their top candidates and/or some of their bottom candidates, as these are the most liked and disliked candidates, but the voters are indifferent among the unranked candidates. Top-truncated ballots may be demanded, for example, if the number of candidates is large and the voters should only declare some of their most liked candidates. Indeed, there are adaptations of the Borda rule to top-truncated ballots that are actually used for

political elections, for example the Irish green party uses the modified Borda count to choose its leader, see [Eme]. In an m -candidate modified Borda count election the points for a t -top-truncated vote $c_{\pi(1)} \succ \cdots \succ c_{\pi(t)} \succ \{c_{\pi(t+1)} \cdots c_{\pi(m)}\}$, are $m - i$ points for the ranked candidates $c_{\pi(i)}$, $1 \leq i \leq t$, and the remaining un-ranked candidates $c_{\pi(j)}$, $t + 1 \leq j \leq m$, all get $m - t - 1$ points. In the POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM-TRUNCATED BALLOTS problem, however, the question is whether the given doubly/top/bottom-truncated ballots may be extended to linear ones such that a distinguished candidate wins the election. Hence, there is no need to adapt the voting rules to the special forms of truncated ballots here. Baumeister et al. [BFLR12] additionally study campaigning problems for truncated ballots, where an adaption of existing voting rules is needed. The formal definition of POSSIBLE WINNER WITH DOUBLY-TRUNCATED BALLOTS for a given voting system \mathcal{E} is as follows.

\mathcal{E} -POSSIBLE WINNER WITH DOUBLY-TRUNCATED BALLOTS	
Given:	A set C of candidates, a list of votes V that are possibly doubly-truncated partial orders over C , and a designated candidate $c \in C$.
Question:	Is there an extension V' of the votes in V to linear orders over C such that c is a winner of election (C, V') under voting system \mathcal{E} ?

We will again drop the prefix “ \mathcal{E} -” if the voting system is clear from the context or not relevant in the corresponding context. The unique-winner variant will be referred to by PWDTB and the co-winner variant by PCWDTB. The problem definitions for top-truncated ballots and bottom-truncated ballots are analogously, and we will refer to these problems by PWTTB and PCWTTB in the case of top-truncated ballots, and by PWBTB and PCWBTB in the case of bottom-truncated ballots. In addition to these problems we will also consider a restricted variant where the number of ranked candidates is bounded. There is either an upper bound that indicates the maximal number of candidates to be ranked or a lower bound that indicates the least number of candidates to be ranked. For a fixed positive constant k , PCWDTBU(k), PCWTTBU(k), and PCWBTBU(k) denote the versions with an upper bound and PCWDTBL(k), PCWTTBL(k), and PCWBTBL(k) denote the version with a lower bound. Truncated ballots with an upper bound on the number of ranked candidates can be seen as heavily truncated, while those with a lower bound can be seen as being only moderately truncated.

In this section we will focus on the co-winner case, but the following obviously also holds for the corresponding unique-winner problems. Since the definitions of all described variants of the possible winner problem with truncated ballots are new to this section, these problems have not been studied before. However, these problems are related to the unweighted coalitional version of the MANIPULATION problem (UCM) and to the original POSSIBLE WINNER problem. Specifically, UCM is

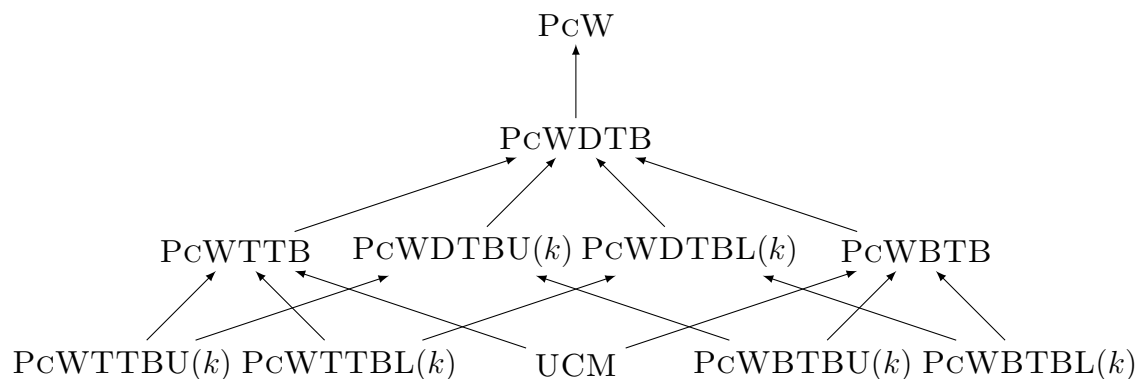


Figure 3.1: A hierarchy of possible winner problems

a special case of PcWTTB and PcWBTB . Since top- and bottom-truncated ballots are special forms of doubly-truncated ballots, PcWTTB and PcWBTB are a special case of PcWDTB , which in turn is a special case of PcW . However since in UCM the ballots are either full or empty, none of the problems with an upper or lower bound on the ranked candidates is a more general problem than UCM . All these relations are stated in Proposition 3.1 and shown in Figure 3.1, where $A \rightarrow B$ means that A (polynomial-time many-one) reduces to B .

Proposition 3.1. *Among the possible winner problems with and without truncated ballots defined above and the unweighted coalitional manipulation problem, we have the reductions shown in Figure 3.1.*

The reductions stated above immediately imply that PcWDTB , PcWTTB , and PcWBTB inherit any membership in P result from PcW and any NP-hardness result from UCM . Summarizing the results for common voting rules from [CSL07, KL05, BD10, XC11], they can be classified into three groups:

1. **POSSIBLE WINNER** is in P for e.g., plurality, veto, Condorcet, and plurality with runoff (in the co-winner case);
2. UCM is NP-hard for, e.g., Copeland, STV, maximum, ranked pairs, most scoring rules, and all rules for which winner determination is NP-hard;
3. **POSSIBLE WINNER** is NP-hard and UCM is in P for, e.g., Bucklin, voting trees, plurality with runoff (in the unique-winner case), and k -approval.

Accordingly, the complexity for **POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM-TRUNCATED BALLOTS** is open only for those voting rules from the third group, e.g., Bucklin, voting trees, plurality with runoff (in the unique-winner case), and k -approval. In the following section we will show

that all three problems are in P for k -approval elections, whereas Betzler and Dorn [BD10] showed that POSSIBLE WINNER for k -approval is NP-complete for all values except 1 and $m - 1$ if there are m candidates.

3.4.1 Results for k -approval

In this section we will show that for k -approval the possible winner problem with truncated ballots is solvable in polynomial time for all three kinds of truncated ballots without an upper or lower bound on the number of ranked candidates. A useful technique to design polynomial time algorithms are flow networks. In the context of voting this was first applied by Faliszewski [Fal08], see also [FHHR09a, BHN09, BD10]. Formally, a flow network is a directed graph $G = (V, E)$, where the set of vertices V contains a source s and a sink t . Furthermore, every edge $(u, v) \in E$ has a capacity $c(u, v) \in \mathbb{N}_0$ indicating the maximum amount of flow that can pass through this edge. A flow in such a network is a function $f : V \times V \rightarrow \mathbb{Z}$ that satisfies the following criteria:

- capacity constraint: $f(u, v) \leq c(u, v)$ for all $u, v \in V$ (i.e., the flow may not exceed the capacity),
- flow conservation: $\sum_{v \in V \setminus \{u\}} f(u, v) = \sum_{v \in V \setminus \{u\}} f(v, u)$ for all $u \in V \setminus \{s, t\}$ (i.e. for each node (except source and sink) the flow entering a node must equal the flow exiting a node).

The value of a flow f is defined as $\sum_{u \in V \setminus \{s\}} f(s, u)$. The *maximum flow problem* seeks to find a maximum flow from the source to the sink. It is well known that this problem is solvable in polynomial time, for example by linear programming. For further details on flow networks, see the textbook [AMO93].

Theorem 3.5. *For k -approval, the problems PCWDTB and, a fortiori, PCWTTB and PCWBTB are in P.*

Proof. Let $V = (v_1, \dots, v_n)$ be a given list of doubly-truncated ballots over a set C of m candidates, with the given values $\text{top}(v_i)$ and $\text{bottom}(v_i)$ for each voter $v_i \in V$. Let $c \in C$ be the designated candidate. To decide whether c is a possible winner, we transform the given instance into the following network flow problem:

1. For each i , $1 \leq i \leq n$, if $\text{top}(v_i) < k$, and the position of c is not revealed in v_i (i.e., c is among the unranked candidates in v_i), then add c at position $\text{top}(v_i) + 1$ in v_i . Let $V' = (v'_1, \dots, v'_n)$ be the corresponding modified profile with adjusted values $\text{top}(v'_i)$, $1 \leq i \leq n$.
2. For each i , $1 \leq i \leq n$:

- if $\text{top}(v'_i) \geq k$, then let Z_i be the set containing the first k candidates of v'_i ;
 - if $\text{top}(v'_i) < k$ and $\text{bottom}(v'_i) \leq m - k$, then let Z_i be the set containing the first $\text{top}(v'_i)$ candidates of v'_i ;
 - if $\text{top}(v'_i) < k$ and $\text{bottom}(v'_i) > m - k$, then let Z_i be the set containing the first $\text{top}(v'_i)$ candidates plus the first $\text{bottom}(v'_i) - m + k$ candidates which are ranked at the bottom of v'_i .
3. For each $d \in C$, let $S(d) = |\{i \mid d \in Z_i\}|$.
4. The flow network contains $n + m + 1$ nodes:
- one node d for each candidate $d \in C \setminus \{c\}$,
 - one node v'_i for each voter $v'_i \in V'$,
 - a source s , and
 - a sink t .
5. The flow network contains the following edges:
- there is an edge from s to every $d \in C \setminus \{c\}$ with capacity $S(c) - S(d)$;
 - there is an edge from $d \in C \setminus \{c\}$ to $v'_i \in V'$ with capacity 1 if and only if the position of d is not revealed in v'_i ;
 - there is an edge from every $v'_i \in V'$ to t with capacity

$$\begin{cases} 0 & \text{if } \text{top}(v'_i) \geq k; \\ k - \text{top}(v'_i) & \text{if } \text{top}(v'_i) < k \text{ and } \text{bottom}(v'_i) \leq m - k; \\ m - \text{top}(v'_i) - \text{bottom}(v'_i) & \text{if } \text{top}(v'_i) < k \text{ and } \text{bottom}(v'_i) > m - k; \end{cases}$$

We claim that c is a possible winner in the k -approval election (C, V) if and only if there is a flow of value $\sum_{i=1}^n a_i$ in the network constructed above, where

- $a_i = 0$ if $\text{top}(v'_i) \geq k$,
- $a_i = k - \text{top}(v'_i)$ if $\text{top}(v'_i) < k$ and $\text{bottom}(v'_i) \leq m - k$, and
- $a_i = m - \text{top}(v'_i) - \text{bottom}(v'_i)$ if $\text{top}(v'_i) < k$ and $\text{bottom}(v'_i) > m - k$.

Assume that c is a possible k -approval winner for (C, V) . That means that there is an extension of the list of truncated ballots V into a list W of complete ones such that c is a k -approval winner of election (C, W) . Without loss of generality, we can assume that c is placed at the first possible position in each vote v_i where its position is unrevealed. Let (C, V') be the profile thus modified. The points every candidate gets in the profile (C, V') correspond to the values $S(d)$ of the above

construction. We now show that there is a flow of value $\sum_{i=1}^n a_i$ in the network. First note that $\sum_{i=1}^n a_i$ is the sum of the unranked candidates among the first k positions in all votes. Since no candidate gets more points than c in (C, W) , there is a flow of value at most $S(c) - S(d)$ from s to every node d for each candidate $d \in C \setminus \{c\}$. If in the list of complete ballots candidate d takes a position in a vote v'_i that was unrevealed in V' , there is a flow of value one from d to v'_i . Further, from each node v'_i , $1 \leq i \leq n$, there is a flow to the sink t whose value corresponds to the number of unrevealed candidates among the first k positions. Hence there is a flow of the desired value in this network.

Now assume that there is a flow of value $\sum_{i=1}^n a_i$ in the network. For the given election (C, V) , we again first place candidate c at the first possible position in the votes where its position was unrevealed before, and we refer to the modified profile by V' . If there is a flow of value one from node d to v'_i , candidate d is placed among the first k positions in vote v'_i . The sum of all a_i ensures that all first k positions are taken in all votes, and the capacity of $S(c) - S(d)$ from the source s to the nodes corresponding to the candidates $d \in C \setminus \{p\}$ ensures that no candidate can get more points than c . Hence, completing the profile V' as described results in a k -approval election in which c is a winner. \square

The proof will be illustrated by the following example.

Example 3.1. *We consider an election with the set $C = \{a, b, c, d, e\}$ of five candidates and the following four doubly-truncated ballots:*

$$\begin{array}{l}
 \hline
 v_1 : \{a, b, c\} > d > e \\
 v_2 : d > \{a, b, e\} > c \\
 v_3 : c > a > \{b, e\} > d \\
 v_4 : a > b > \{c, d\} > e \\
 \hline
 \end{array}$$

The election system is 3-approval and the candidate we want to make win the election is c . In Step 1 we consider all votes where the first three positions are not yet filled and the position of candidate c is unrevealed. In these votes c is placed at the first possible position. This results in the following list of votes:

$$\begin{array}{l}
 \hline
 v'_1 : c > \{a, b\} > d > e \\
 v'_2 : d > \{a, b, e\} > c \\
 v'_3 : c > a > \{b, e\} > d \\
 v'_4 : a > b > c > d > e \\
 \hline
 \end{array}$$

In the next step we compute $Z_1 = \{c\}$, $Z_2 = \{d\}$, $Z_3 = \{a, c\}$, and $Z_4 = \{a, b, c\}$, and obtain $S(a) = 2$, $S(b) = 1$, $S(c) = 3$, $S(d) = 1$, and $S(e) = 0$. This leads to the following capacities of the arcs of the flow network: $(s, a) \mapsto 1$, $(s, b) \mapsto 2$, $(s, d) \mapsto 2$, $(s, e) \mapsto 3$, $(v'_1, t) \mapsto 2$, $(v'_2, t) \mapsto 2$, $(v'_3, t) \mapsto 1$, $(v'_4, t) \mapsto 0$, and the

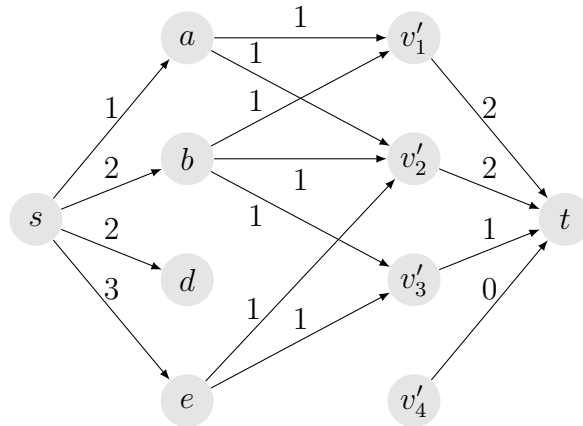


Figure 3.2: Flow network for Example 3.1

capacities of the arcs from $d \in C \setminus \{c\}$ to $v'_i \in V'$ where the position of d is not revealed in v'_i is 1. The resulting flow network is presented in Figure 3.2. The maximal flow of 5 is obtained for example by assigning the following values to the edges: $(s, a) : 1$, $(s, b) : 2$, $(s, e) : 2$, $(a, v'_1) : 1$, $(b, v'_1) : 1$, $(b, v'_2) : 1$, $(e, v'_2) : 1$, $(e, v'_3) : 1$, $(v'_1, t) : 2$, $(v'_2, t) : 2$, $(v'_3, t) : 1$, and the value for the remaining edges is always zero. This corresponds for example to the following extension of the votes $v'_1 : c > a > b > d > e$, $v'_2 : d > b > e > a > c$, $v'_3 : c > a > e > b > d$, and $v'_4 : a > b > c > d > e$, where a , b , and c tie for winning with 3 points. Therefore c is a possible co-winner.

3.5 The Possible Winner Problem with Uncertain Weights

So far the possible winner problem has mostly been studied for unweighted elections. Lang et al. [LPR⁺12] studied the possible winner problem for weighted profiles for Schwartz winners and balanced trees and Baumeister et al. [BRR11] (see also Section 3.3 of this thesis) obtained some NP-hardness results for the weighted version of the POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES problem. Obviously all NP-hardness results from the original POSSIBLE WINNER problem carry over to a weighted version of this problem. But here we investigate another POSSIBLE WINNER variant where the uncertainty lies in the voters' weights.

Consider for example a university ranking based on different criteria (e.g., equipment, third-party funds, graduation rates, etc.). This can be seen as an election where the universities are the candidates and the linear order of the universities for each criteria are the votes. Assume that the voting rule is fixed, but is it possible

to make a distinguished university be on top of the ranking by carefully choosing the weights for the single criteria?

Formally, this problem can be stated as follows for a given voting system \mathcal{E} .

\mathcal{E} -POSSIBLE WINNER WITH UNCERTAIN WEIGHTS	
Given:	A set C of candidates, two lists V_0 and V_1 of votes that are linear orders over C , where the weights of the voters in V_0 are not specified yet and weight zero is allowed for them, yet all voters in V_1 have weight one, and a designated candidate $c \in C$.
Question:	Is there an assignment of weights $w_i \in \mathbb{N}_0$ to the votes $v_i \in V_0$ such that c is a winner of election $(C, V_0 \cup V_1)$ under voting system \mathcal{E} when v_i 's weight is w_i for $1 \leq i \leq V_0 $?

We again stated the problem in the co-winner case, use \mathcal{E} -POSSIBLE WINNER WITH UNCERTAIN WEIGHTS as a generic term, refer to the explicit unique-winner variant by \mathcal{E} -PWUW, and to the explicit co-winner variant by \mathcal{E} -PCWUW. The prefix “ \mathcal{E} -” will as well be dropped if the voting system used is clear from the context. Note that for inputs where V_0 is empty we obtain the ordinary unweighted winner problem for \mathcal{E} , where we simply ask whether a distinguished candidate is a winner for a given election. Hence PCWUW is a generalization of the winner problem, but since the winner problem is in P for all voting systems considered in this section, no results carry over. That we allow weight zero for the voters in V_0 corresponds in some sense to control by deleting candidates (see [BTT92, HHR07]). The relationship to control by adding candidates will be discussed in more detail later in this section. Distinguishing between votes with unit-weight in V_1 and with uncertain weight in V_0 in our problem instances captures these problems in their full generality; just as the original possible winner problem allows for linear votes. And the restriction that the weights of the votes in V_1 are normalized to unit-weight is a restriction (that doesn't hurt) and is chosen at will. This will somewhat simplify our proofs.

Although most proofs can be easily adapted for the unique-winner case, we focus on the co-winner case and in addition to PCWUW we will study several restrictions of this problem.

- We add regions (i.e., intervals) $R_i \subseteq \mathbb{N}_0$, $1 \leq i \leq |V_0|$, to the problem instance and require that in addition each weight w_i must be chosen from R_i . This variant will be denoted by PCWUW-RW.
- We add a positive bound $B \in \mathbb{N}_0$ to the problem instance and require that the sum of the weights does not exceed B , i.e., $\sum_{i=1}^{|V_0|} w_i \leq B$. This variant will be denoted by PCWUW-BW.

- We combine both variants by adding regions (i.e., intervals) $R_i \subseteq \mathbb{N}_0$, $1 \leq i \leq |V_0|$, and a positive bound $B \in \mathbb{N}_0$ to the problem instance and require that each weight w_i must be chosen from R_i and that $\sum_{i=1}^{|V_0|} w_i \leq B$. This variant will be denoted by PCWUW-BW-RW.

Baumeister et al. [BRR⁺12] also define a variant of the POSSIBLE WINNER WITH UNCERTAIN WEIGHTS problem where the weights are not integers, as assumed here, but positive rationals. Furthermore one could allow sets of intervals for each weight or define a destructive variant of all these problems, but we focus on the four problems defined above. Among these problems the following reductions trivially hold:

$$\text{PCWUW-RW} \leq_m^{\text{P}} \text{PCWUW-BW-RW} \quad (3.1)$$

$$\text{PCWUW-BW} \leq_m^{\text{P}} \text{PCWUW-BW-RW} \quad (3.2)$$

The first one holds by setting the bound on the total weight to the sum of the highest possible weights, and the second one by setting the intervals to $[0, B]$, where B is the bound on the total weight.

As mentioned above, this problem is closely related to constructive control by deleting voters, since weight zero is allowed for voters from V_0 . Furthermore PCWUW is related to constructive control by adding voters (see [BTT92]), CCAV for short, since raising the weight for a voter from V_0 corresponds to adding such a vote to the election. An instance of CCAV consists of a set of candidates C with a distinguished candidate $c \in C$, a list V of registered voters, a list V' of yet unregistered voters, and a positive integer k . The control action is successful if c can be made win the election by adding at most k voters from V' to the election. Obviously, there is a polynomial-time many-one reduction from CCAV to PCWUW-BW-RW. Here, the voters from V_1 are the registered voters from V , and the unregistered voters V' are those from V_0 . The regions to choose the weights from is $[0, 1]$ for all votes in V_0 , and the bound on the total weight B is set to k , the maximum number of voters that can be added. Assuming succinct representation also leads to a polynomial-time many-one reduction from PCWUW-BW-RW to CCAV. The registered voters are those from V_1 and the list of unregistered voters contains the voters from V_0 , where each vote is added according to its maximal weight in the PCWUW instance. The maximum number k of voters that can be added equals the bound B on the total weight. Having reductions in both directions, all known results for CCAV directly carry over to PCWUW-BW-RW. For the voting systems k -approval and Copeland ^{α} that we consider here, this implies that PCWUW-BW-RW is NP-complete for Copeland⁰ and Copeland¹, and in P for plurality (see [FHHR09a, BTT92]). We nevertheless prove these existing results, since our proofs always cover other variants of our problem at the same time. Conversely the reductions imply that all results for PCWUW-BW-RW obtained here directly hold for CCAV if we assume succinct representation.

3.5.1 Results

In this section we study the complexity of PCWUW and its variants for k -approval and Copeland ^{α} elections. The first theorem shows polynomial time solvability under plurality, veto, and 2-approval for all variants, and for two variants under k -approval, $k \geq 1$.

Proposition 3.2. *1. Each of the four variants of plurality-PCWUW, veto-PCWUW and 2-approval-PCWUW studied in this section is in P.*

2. For each $k \geq 1$, k -approval-PCWUW and k -approval-PCWUW-RW are in P.

Proof. For plurality-PCWUW the optimal strategy is to set the weights of the voters having the distinguished candidate c not on the top position to the minimum possible value, and the weight for those having c on the first position to the maximum value needed without exceeding the total bound on the weights. Similar arguments show that veto-PCWUW is solvable in polynomial time.

To see that all variants of 2-approval-PCWUW are also in P, we will give a max-flow instance that solves the problem 2-approval-PCWUW-BW-RW where the ranges of the weights for the single votes are $\{0, 1\}$ and the bound on the total weight is B . This proof can easily be adapted for other ranges. Given a 2-approval-PCWUW-BW-RW instance as defined above, first note that the only votes from V_0 that have to be considered are those having c among the top two positions. This set will be denoted by V'_0 . And without loss of generality we may assume that the bound on the total weight satisfies $B \leq |V'_0|$, since otherwise, the optimal strategy is to let the weights of the votes in V'_0 be 1 and to let the weights of all other votes be 0. Now construct a network with the set $\{s, s', t\} \cup V'_0 \cup (C \setminus \{c\})$ of vertices, and the following edges and capacities:

1. There is an edge $s \rightarrow s'$ with capacity B ,
2. there is an edge from s to each node in V'_0 with capacity 1,
3. there is an edge from a node L in V'_0 to a node d in $C \setminus \{c\}$ with capacity 1 if and only if d is ranked among the top two positions in L (note that for each vote in V'_0 there is only one such candidate, since c is also ranked among the top two positions),
4. there is an edge from each node $d \in C \setminus \{c\}$ to t with capacity $B + score_{(C, V_1)}(c) - score_{(C, V_1)}(d)$, where $score_{(C, V_1)}(e)$ is the 2-approval score of any $e \in C$ in vote list V_1 .

Now we ask if there is a maximum flow whose value is B . We note that in the PCWUW instance, it is always optimal to choose B votes in V'_0 and to let their

weights be 1. The claim for 2-approval-PCWUW-RW and 2-approval-PCWUW-BW then follows from the reductions (3.1) and (3.2) stated above. The results for the corresponding problems without ranges for the weights and a bound on the total weight follow from the second statement.

For the problems k -approval-PCWUW and k -approval-PCWUW-RW for $k \geq 1$, it suffices to maximize the weights of the votes in V_0 that rank c among their top k positions, and to minimize the weights of the other votes. \square

In particular, it is open whether 3-approval-PCWUW-BW-RW and 3-approval-PCWUW-BW are also in P. For $k \geq 4$ however, we can show that these problems are NP-complete. The proof is by a reduction from the NP-complete problem EXACT-COVER BY 3-SETS (see, e.g., [GJ79]) which is defined as follows.

EXACT COVER BY 3-SETS (X3C)	
Given:	A set $\mathcal{B} = \{b_1, \dots, b_{3q}\}$ and a collection $\mathcal{S} = \{S_1, \dots, S_n\}$ with $ S_i = 3$ and $S_i \subseteq \mathcal{B}$, $1 \leq i \leq n$.
Question:	Does \mathcal{S} contain an exact cover for \mathcal{B} , i.e. a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that every element of \mathcal{B} occurs in exactly one member of \mathcal{S}' ?

Theorem 3.6. *For each $k \geq 4$, k -approval-PCWUW-BW-RW and k -approval-PCWUW-BW are NP-complete.*

Proof. It is easy to see that both problems belong to NP. For proving NP-hardness, we give a proof for k -approval-PCWUW-BW by a reduction from the NP-complete problem X3C. Given an X3C instance $(\mathcal{B}, \mathcal{S})$ with $\mathcal{B} = \{b_1, \dots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$, we construct an instance of k -approval-PCWUW-BW as follows. The set of candidates is $C = \{c, b_1, \dots, b_{3q}, b_1^1, \dots, b_{3q}^1, b_1^2, \dots, b_{3q}^2, b_1^3, \dots, b_{3q}^3\}$, where c is the distinguished candidate. The set V_0 contains n votes of the form $c > \vec{S}_i > \dots$, and V_1 contains $q - 1$ votes of the form $b_j > b_j^1 > b_j^2 > b_j^3 > \dots$ for each j , $1 \leq j \leq 3q$. The bound on the total weight of the votes in V_0 is $B = q$. Recall that the votes in V_1 all have fixed weight one, and those of the votes in V_0 are from \mathbb{N}_0 . We show that \mathcal{S} has an exact cover for \mathcal{B} if and only if we can set the weights of the voters in this election such that c is a winner.

Assume that there is an exact cover $\mathcal{S}' \subseteq \mathcal{S}$ for \mathcal{B} . By setting the weights of the votes $c > \vec{S}_i > \dots$ to one for those q subsets S_i contained in \mathcal{S}' , and to zero for all other votes in V_0 , c is a winner of the election, as c and all b_j , $1 \leq j \leq 3q$, receive exactly q points, whereas b_j^1 , b_j^2 , and b_j^3 , $1 \leq j \leq 3q$, receive $q - 1$ points each.

Conversely, assume that c can be made a winner of the election by choosing the weights of the votes in V_0 appropriately. Note that the bound on the total weight for the votes in V_0 is $B = q$. Every b_j gets $q - 1$ points from the votes in V_1 , and c gets points only from the votes in V_0 . Since there are always some b_j getting points if a vote from V_0 has weight one, there are at least three b_j having q points

if a vote from V_0 has weight one. Hence c must get q points from the votes in V_0 by setting the weight of q votes to one. Furthermore, every b_j can occur only once in the votes having weight one in V_0 , as otherwise c would not win. Thus, the S_i corresponding to the votes of weight one in V_0 must form an exact cover for \mathcal{B} .

By adding dummy candidates to fill the positions receiving points, this proof can be adapted for k -approval for any fixed $k > 4$. NP-hardness for k -approval-PCWUW-BW-RW, $k \geq 4$, then follows from the trivial reduction (3.2) stated above. \square

Now we consider Copeland $^\alpha$ elections and show that all PCWUW variants studied in this section are NP-complete, for each rational value of α , $0 \leq \alpha \leq 1$. To do so we will use the representation of weighted majority graphs for our election. A weighted majority graph for an election $E = (C, V)$ contains one node for each candidate and there is a directed edge from candidate c to d , with weight $N(c, d)$, where $N(c, d)$ is the number of voters who prefer c to d minus the number of voters who prefer d to c . Since $N(c, d) = -N(d, c)$ it is enough to give only one of these two edges explicitly. Note that the parity is the same for all weights, and whether it is odd or even depends on the parity of the number of votes. In the next proof we apply *McGarvey's trick* [McG53] to weighted majority graphs to construct a profile corresponding to a given weighted majority graph, see [Deb87, MPS08]. Similar as we used Lemma 3.1 in Section 3.2 to construct a profile with some given properties, we will use this trick to just specify the weighted majority graph instead of the whole list of votes. Originally, McGarvey showed that for every unweighted majority graph there is a particular list of preferences that results in this graph. This trick can be extended to weighted majority graphs since adding the two votes $c > d > c_3 > \dots > c_m$ and $c_m > c_{m-1} > \dots > c_3 > c > d$ to a list of votes increases the weight on the edge $c \rightarrow d$ by 2 and hence decreases the weight on the edge $d \rightarrow c$ by 2, while the weights on all other edges remain unchanged. Now we will make use of this trick to show NP-hardness for Copeland $^\alpha$ -PCWUW in the following theorem.

Theorem 3.7. *For each rational α , $0 \leq \alpha \leq 1$, every variant of Copeland $^\alpha$ -PCWUW studied in this section is NP-complete.*

Proof. NP-membership is easy to see for all problem variants. We first prove NP-hardness for Copeland $^\alpha$ -PCWUW, and then show how to modify the proof for the variants of the problem. Given an X3C instance $(\mathcal{B}, \mathcal{S})$ with $\mathcal{B} = \{b_1, \dots, b_{3q}\}$ and $\mathcal{S} = \{S_1, \dots, S_n\}$. Without loss of generality we assume that $q \geq 4$. Now we construct the following PCWUW instance for Copeland $^\alpha$, where the set of candidates is $\mathcal{B} \cup \{c, d, e\}$. We are asked whether c can be made a winner.

The votes on C are defined as follows. V_0 will encode the X3C instance and V_1 will be used to implement McGarvey's trick. V_0 consists of the following n votes:

For each j , $1 \leq j \leq n$, there is a vote $d > e > \vec{S}_j > c > \dots$. V_1 is the list of votes whose weighted majority graph has the following edges:

- $c \rightarrow d$ with weight $q + 1$, $d \rightarrow e$ with weight $q + 1$, and $e \rightarrow c$ with weight $q + 1$.
- For every i , $1 \leq i \leq 3q$, $d \rightarrow b_i$ and $e \rightarrow b_i$ each with weight $q + 1$, and $b_i \rightarrow c$ with weight $q - 3$.
- The weight on any other edge not defined above is no more than 1.

It follows that no matter what the weights of the votes in V_0 are, d beats e and e beats c in pairwise elections, and both d and e beat all candidates in \mathcal{B} in pairwise elections. For c to be a winner, c must beat d in the pairwise election, which means that the total weight of the votes in V_0 is no more than q . On the other hand, c must beat all candidates in \mathcal{B} . This happens if and only if the votes in V_0 that have positive weights correspond to an exact cover of \mathcal{B} , and all of these votes must have weight one. This means that Copeland ^{α} -PCWUW is NP-hard.

For the BW and BW-RW variants, we let $B = q$; for the RW and BW-RW variants, we let the range of each vote in V_0 be $\{0, 1\}$. □

Furthermore Baumeister et al. [BRR⁺12] studied PCWUW for the voting systems ranked pairs, Bucklin, and fallback voting that are not considered here, and obtained NP-completeness for all variants. In addition they studied the PCWUW in the case where the weights are positive rationals instead of integers as defined here and they obtained polynomial time algorithms for voting systems that can be represented by linear inequalities (for example scoring rules, Bucklin, fallback voting, and plurality with runoff).

3.6 The Possible Winner Problem under Uncertain Voting System

In all of the above-defined problems the uncertainty is in the preferences. In this section we study a possible winner problem where uncertainty is in the voting system itself, POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM. The motivation to study this problem is that uncertainty about the voting system may give an incentive to vote truthfully for the voters, since reporting an insincere preference may result in a worse outcome. Consider for example the following situation. There are three candidates a , b , and c , three sincere votes $c > a > b$, two sincere votes $b > a > c$, and two strategic voters with the true preference $a > b > c$. If the strategic voters know for sure that the election is held under plurality, their favorite candidate a cannot win and instead of wasting their votes

by voting sincerely and letting c win the election the may benefit from reporting the insincere preference $b > a > c$ as then candidate b , who is preferred to c by the strategic voters, wins the election. However if the election was held under the Borda rule, candidate a wins the election with 9 points if the strategic voters report their sincere preference, as candidate b and c have only 6 points. But if the strategic voters report the insincere preference $b > a > c$ candidate b wins with 8 points in front of candidate a with 7 and candidate c with 6 points. Hence uncertainty about the voting systems may give a strong incentive to reveal the true preference. For a given class \mathcal{V} of voting systems, this problem is formally defined as follows.

\mathcal{V} -POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM

Given: A set of candidates C , a list of voters V consisting of linear orders over C , and a designated candidate $c \in C$.

Question: Is there a voting system \mathcal{E} in \mathcal{V} such that c is a winner of the election held under \mathcal{E} ?

As for the other possible winner problems we will use POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM as a generic term and drop the prefix \mathcal{V} - if appropriate. The explicit co-winner variant will be denoted by PCWUVS and the explicit unique-winner variant by PWUVS.

Uncertainty about the voting rule that will be used to determine the winners of an election was also studied by Elkind and Erdélyi [EE12] yet not with respect to winner determination but with respect to manipulation. They investigate the problem whether a successful manipulation for a single manipulator or a group of manipulators is possible if the voting rule that will be used is not fixed in advance but will be chosen from a given list of voting rules. They obtained NP-hardness results as well as polynomial-time solvability.

3.6.1 Results

This section studies PWUVS and PCWUVS with respect to the family of scoring rules, Copeland ^{α} elections, and preference-based approval voting.

Scoring Rules The first family of voting rules we consider is the family of scoring rules. Recall that c is the designated candidate we want to make a winner in the given m -candidate election, by specifying the values α_i of the scoring vector $(\alpha_1, \dots, \alpha_m)$ appropriately. In the proof of Theorem 3.8 we will need the following notions.

Definition 3.1. For an election $E = (C, V)$, let $pos_i(x)$ denote the total number of times candidate $x \in C$ is at position i , $1 \leq i \leq |C|$, in the list V of votes, and for all $a \in C \setminus \{c\}$, let $plus_{(c,i)}(a) = pos_i(a) - pos_i(c)$.

If the election is held under scoring vector $(\alpha_1, \dots, \alpha_m)$, candidate c wins if and only if for each $a \in C \setminus \{c\}$, we have $\sum_{i=1}^{|C|} plus_{(c,i)}(a) \cdot \alpha_i \leq 0$ in the co-winner case. For the unique-winner case, replace the zero on the right-hand side of the inequality by one.

To prove our result for the POSSIBLE WINNER problem in Section 3.2, we used a lemma by Betzler and Dorn [BD10] (stated here as Lemma 3.1) to construct a list of linear votes with specific properties and in Section 3.5 we used McGarvey's trick to construct a list of votes from a given majority graph. Similar to this, we will now give another lemma to construct a list of votes needed in the proof of Theorem 3.8. Specifically, in the following lemma we will show how to construct a list of votes for given values $plus_{(c,i)}(a)$ under some conditions. Let $M_{(d,i)}$ denote a circular block of $|C| - 1$ votes, where candidate d is always at position i and all other candidates take all the remaining positions exactly once, by shifting them in a circular way. For example, for the set $C = \{d, c_1, \dots, c_m\}$ of candidates the circular block $M_{(d,m+1)}$ looks as follows:

$$\begin{array}{cccccc} c_1 & > & c_2 & > & \dots & > & c_{m-1} & > & c_m & > & d \\ c_2 & > & c_3 & > & \dots & > & c_m & > & c_1 & > & d \\ \vdots & & \vdots & & \vdots & & \vdots & & \vdots & & \vdots \\ c_m & > & c_1 & > & \dots & > & c_{m-2} & > & c_{m-1} & > & d \end{array}$$

Lemma 3.2. *Let C be a set of m candidates, $c \in C$ be a distinguished candidate, $d \in C$ be a dummy candidate, and let the values $plus_{(c,i)}(a) \in \mathbb{Z}$, $1 \leq i \leq m - 1$, for all candidates a in $C \setminus \{c, d\}$ be given. Let $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m)$ be an arbitrary scoring vector with $\alpha_m = 0$. One can construct in time polynomial in m a list V of votes satisfying that*

1. *every candidate $a \in C \setminus \{c, d\}$ has the given values $plus_{(c,i)}(a)$, $1 \leq i \leq m - 1$, in election (C, V) and*
2. *candidate d cannot beat c in election (C, V) .*

Proof. Let $m = |C|$ be the number of candidates. For each positive value $plus_{(c,i)}(a)$, $1 \leq i \leq m - 1$, $a \in C \setminus \{c, d\}$, we construct two types of circular blocks of votes. The first block is of type $M_{(d,i)}$, except that in the vote in which candidate a is at position m , the positions of a and d are swapped. For this block it holds that $plus_{(c,i)}(a) = 1$, and all other values $plus_{(c,j)}(b)$, $b \in C \setminus \{c, d, a\}$, and $plus_{(c,j)}(a)$, $1 \leq j \leq m - 1$, remain unchanged. These blocks will be added with multiplicity $plus_{(c,i)}(a)$. To ensure that candidate d has no chance to beat candidate c , we add the votes of the circular block $M_{(d,m)}$ with multiplicity $m \cdot plus_{(c,j)}(a)$. Clearly, this block does not affect the values $plus_{(c,j)}(b)$, $1 \leq j \leq m - 1$, $b \in C \setminus \{c, d\}$.

If $plus_{(c,i)}(a)$ is negative, we add the block of type $M_{(d,m)}$, where the places of a and d are swapped in the vote in which a is at position i , with multiplicity

$-plus_{(c,i)}(a)$. The effect is that $plus_{(c,i)}(a)$ is decreased by 1 for each of these blocks. Again, to ensure that candidate d will not be able to beat candidate c , we add the circular block $M_{(d,m)}$ with multiplicity $-plus_{(c,i)}(a) + 1$.

By construction, the values $plus_{(c,i)}(d)$, $1 \leq i \leq m - 1$, are never positive, so obviously d has no chance to beat or tie with c in the election whatever scoring rule will be used. Since the votes can be stored as a list of binary integers representing their corresponding multiplicities, these votes can be constructed in time polynomial in m . \square

To make use of Lemma 3.2, we assume succinct representation of the election, see the comments made in Section 3.1 for more details on succinct representation. The following NP-hardness proof is by reduction from the NP-complete problem INTEGER KNAPSACK (see, e.g., [GJ79]) which is defined as follows.

INTEGER KNAPSACK	
Given:	A finite set of elements $U = \{u_1, \dots, u_n\}$, two mappings $s, v : U \rightarrow \mathbb{N}$, and two positive integers, b and k .
Question:	Is there a mapping $c : U \rightarrow \mathbb{N}$ such that
$\sum_{i=1}^n c(u_i) \cdot s(u_i) \leq b \text{ and } \sum_{i=1}^n c(u_i) \cdot v(u_i) \geq k?$	

Theorem 3.8. *Let \mathcal{S} be the class of scoring rules with $m \geq 4$ candidates that are defined by a scoring vector of the form $\vec{\alpha} = (\alpha_1, \dots, \alpha_{m-4}, x_1, x_2, x_3, 0)$, with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$. \mathcal{S} -PCWUVS and \mathcal{S} -PWUVS are NP-complete (assuming succinct representation).*

Proof. Membership in NP is obvious, and the proof of NP-hardness will be by a reduction from the NP-complete problem INTEGER KNAPSACK.

We first focus on the co-winner case and then show how to transfer the proof to the unique-winner case. Let (U, s, v, b, k) be an instance of INTEGER KNAPSACK with $U = \{u_1, \dots, u_n\}$ and let $c : U \rightarrow \mathbb{N}$ be a mapping. Then it holds that

$$\begin{aligned} \sum_{i=1}^n c(u_i) \cdot s(u_i) &\leq b \\ \sum_{i=1}^n c(u_i) \cdot v(u_i) &\geq k \end{aligned} \tag{3.3}$$

$$\begin{aligned}
 &\Leftrightarrow \begin{pmatrix} s(u_1) & s(u_2) & \cdots & s(u_n) \\ -v(u_1) & -v(u_2) & \cdots & -v(u_n) \end{pmatrix} \begin{pmatrix} c(u_1) \\ c(u_2) \\ \vdots \\ c(u_n) \end{pmatrix} \leq \begin{pmatrix} b \\ -k \end{pmatrix} \\
 &\Leftrightarrow \begin{pmatrix} -b' \\ k' \\ nb \\ A & (n-1)b \\ \vdots \\ b \end{pmatrix} \begin{pmatrix} c'(u_1) \\ c'(u_2) \\ \vdots \\ c'(u_n) \\ 1 \end{pmatrix} \leq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \tag{3.4}
 \end{aligned}$$

with

$$\begin{aligned}
 A &= \begin{pmatrix} s(u_1) & s(u_2) & \cdots & s(u_n) \\ -v(u_1) & -v(u_2) & \cdots & -v(u_n) \\ -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & & & \\ 0 & \cdots & 0 & -1 \end{pmatrix} \in \mathbb{Z}^{(n+2) \times n}, \\
 c'(u_i) &= c(u_i) + (n-i+1)b, \quad 1 \leq i \leq n, \\
 b' &= b + \sum_{i=1}^n b \cdot s(u_i) \cdot (n-i+1), \text{ and} \\
 k' &= k + \sum_{i=1}^n k \cdot v(u_i) \cdot (n-i+1).
 \end{aligned}$$

The last n rows of the matrix ensure that $c'(u_i) \geq (n-i+1)b$, $1 \leq i \leq n$, and so there are no new solutions added for which the values $c(u_i)$ may be negative. Furthermore, since $c(u_i) \leq b$, it is now ensured that $c'(u_1) \geq c'(u_2) \geq \cdots \geq c'(u_n) \geq b$. Hence it still holds that c is a solution for the given INTEGER KNAPSACK instance if and only if c' is a solution for (3.4).

We will now build an election $E = (C, V)$ with candidate set $C = \{c, d, e, f, g_1, \dots, g_n\}$, where c is the distinguished candidate and d is a dummy candidate who cannot beat c in the election whatever scoring rule will be used. The list of voters will be built using Lemma 3.2 according to the matrix in (3.4). The $n+2$ rows in the matrix correspond to the candidates e , f , and g_1, \dots, g_n . Since the matrix has only $n+1$ columns, the positions $n+2$ and $n+3$ in the votes will have no effect on the outcome of the election, and thus the corresponding $plus_{(c,i)}(a)$ values, $n+2 \leq i \leq n+3$, can be set to zero for all candidates $a \in \{e, f, g_1, \dots, g_n\}$. The corresponding values in the scoring vector can be set to

either zero or one, respecting the conditions for a valid scoring vector. Hence, the votes in V have to fulfill the following properties:

$$\begin{aligned} plus_{(c,i)}(e) &= \begin{cases} s(u_i) & \text{for } 1 \leq i \leq n \\ -b' & \text{for } i = n + 1 \\ 0 & \text{for } n + 2 \leq i \leq n + 3, \end{cases} \\ plus_{(c,i)}(f) &= \begin{cases} -v(u_i) & \text{for } 1 \leq i \leq n \\ k' & \text{for } i = n + 1 \\ 0 & \text{for } n + 2 \leq i \leq n + 3, \end{cases} \\ plus_{(c,i)}(g_j) &= \begin{cases} -1 & \text{for } 1 \leq i \leq n, i = j \\ (n - i + 1)b & \text{for } i = n + 1, 1 \leq j \leq n \\ 0 & \text{for } 1 \leq i \leq n + 3, 1 \leq j \leq n, i \neq j. \end{cases} \end{aligned}$$

According to Lemma 3.2, these votes can be constructed in polynomial time such that the dummy candidate d has no influence on c being a winner of the election or not, whatever scoring rule of type $\vec{\alpha} = \{\alpha_1, \dots, \alpha_n, 1, \alpha_{n+2}, \alpha_{n+3}, 0\}$ will be used.

Since the $plus_{(c,i)}(a)$ values assigned to the candidates $a \in C \setminus \{c, d\}$ are set according to the matrix in (3.4), it holds that c can be a winner in election $E = (C, V)$ by choosing a scoring rule of the form $\vec{\alpha} = \{\alpha_1, \dots, \alpha_n, 1, \alpha_{n+2}, \alpha_{n+3}, 0\}$ if and only if for each $a \in C \setminus \{c\}$, we have

$$\sum_{i=1}^n plus_{(c,i)}(a) \cdot c(u_i) + plus_{(c,n+1)}(a) \leq 0.$$

As described above, the values in the scoring vector for positions $n + 2$ and $n + 3$, have no effect on the outcome of the election, hence by switching rows in the matrix we can extend the set of possible scoring rules to scoring rules of the form $\vec{\alpha} = (c(u_1), \dots, c(u_n), x_1, x_2, x_3, 0)$ with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$. Hence c can be made a winner of the election $E = (C, V)$ if and only if there is a solution to (3.4). Since we have shown above that there is a solution to (3.3) if and only if there is a solution to (3.4), it holds that there is a solution c to our INTEGER KNAPSACK instance if and only if there is a scoring rule $\vec{\alpha}$, of the form described above, under which c wins the election $E = (C, V)$.

To see that this reduction also settles the unique-winner case, note that (3.4) is equivalent to the following inequality:

$$\begin{pmatrix} -b' + 1 \\ k' + 1 \\ nb + 1 \\ A & (n - 1)b + 1 \\ & \vdots \\ & b + 1 \end{pmatrix} \begin{pmatrix} c'(u_1) \\ c'(u_2) \\ \vdots \\ c'(u_n) \\ 1 \end{pmatrix} \leq \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}. \quad (3.5)$$

The election we need to construct has the same candidate set as above and the voters are constructed according to the values $plus_{(c,n+1)}(a)$ for $a \in C \setminus \{c, d\}$ in the matrix of (3.5). Thus, c is the unique winner of the modified election if and only if for each $a \in C \setminus \{c\}$, we have

$$\sum_{i=1}^n plus_{(c,i)}(a) \cdot c(u_i) + plus_{(c,n+1)}(a) \leq 1.$$

By a similar argument as above, there is a scoring rule of the form $\vec{\alpha} = (\alpha_1, \dots, \alpha_n, x_1, x_2, x_3, 0)$ with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$ in which c wins the election if and only if there is a solution c for the given INTEGER KNAPSACK instance. \square

Copeland $^\alpha$ Elections In this paragraph we consider the POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM problem with respect to the family of Copeland $^\alpha$ elections. Recall that the parameter α is a rational number from the interval $[0, 1]$ that specifies how ties are rewarded in the pairwise comparisons between candidates. In contrast to the considered subclass of scoring rules, where it is NP-hard to determine whether there exists a scoring rule from the given class that makes a distinguished candidate win, we show that the problem is solvable in polynomial time when the voting rule is chosen from the family of Copeland $^\alpha$ elections.

Theorem 3.9. *\mathcal{C} -PCWUVS and \mathcal{C} -PWUVS are polynomial-time solvable for the family of Copeland $^\alpha$ elections:*

$$\mathcal{C} = \{\text{Copeland}^\alpha \mid \alpha \text{ is a rational number in } [0, 1]\}.$$

Proof. To decide whether a distinguished candidate c can be made a winner of the election by choosing the parameter α after all votes have been cast, we do the following. In the co-winner case, for each $d \in C \setminus \{c\}$, compute

$$f(d) = \begin{cases} \frac{win(c) - win(d)}{tie(c) - tie(d)} & \text{if } tie(c) \neq tie(d) \\ win(c) - win(d) & \text{otherwise.} \end{cases}$$

If $f(d) \geq 0$ for all $d \in C$, c can be made a winner of the election by setting $\alpha = \min_{d \in C} \{f(d), 1\}$, and otherwise c cannot be made a winner. So \mathcal{C} -PCWUVS is in P.

In the unique-winner case, for c to be the unique winner of the election, it must hold that $f(d) > 0$, for all $d \in C \setminus \{c\}$, and α is set to a value greater than $\min_{d \in C} \{f(d)\}$ if this value is less than one, or else to one. Otherwise c cannot be made the unique winner of the election. So \mathcal{C} -PWUVS is in P. \square

Preference-Based Approval Voting In approval voting the situation is a bit different, since approval voting is not a class of voting systems, and the voters usually do not report linear preferences but approval vectors. But in preference-based approval voting, the voters report a strict preference order along with an approval line. If we assume that the approval lines are not set by the voters (who thus only report their linear orders) but are set by the voting system itself (after all votes have been cast), we obtain (for m candidates and n voters) a class $\mathcal{A}_{m,n}$ of $(m - 1)^n$ voting systems. The winners in each such system are the candidates with the highest number of approvals. Note that these voting systems are not very natural (as they do not let the voters themselves choose their approval strategies) and do not possess generally desirable social-choice properties (e.g., the systems in $\mathcal{A}_{m,n}$ are not even anonymous, as changing the order of votes may result in a different outcome).

In this setting, given an election where voters report their preference orders, setting the approval lines afterwards corresponds to choosing a system from $\mathcal{A}_{m,n}$. It is easy to see that PCWUVS and PWUVS are polynomial-time solvable for this class. To make the distinguished candidate c win the election, choose the system that sets the approval line in each vote that does not rank c at the last position right behind c , and in the votes that do rank c last right behind the top candidate. If c is not a winner (unique winner) of this election, c cannot win (be a unique winner of) the election whatever system from the class is chosen. Thus PCWUVS and PWUVS are polynomial-time solvable for this class of preference-based approval voting systems.

Elkind et al. [EFS09] study the related problem of “mixed bribery”, which is a variant of SWAP BRIBERY defined for SP-AV, which is also a voting system that combines preference based voting and approval voting. Here the briber may ask the voters to swap adjacent candidates and/or to move the approval line. In contrast to our result, they showed NP-hardness even if the bribery is only allowed to move the approval line.

3.7 Related Problems

The relation of POSSIBLE WINNER and MANIPULATION has already been elaborated in Section 3.1, by showing that UCM is a special case of PW and in Section 3.4 the connections to various POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM TRUNCATED BALLOTS problems have been studied. The POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES problem studied in Section 3.3 problem in turn is a special case of the original POSSIBLE WINNER problem. Another problem related to POSSIBLE WINNER is the SWAP BRIBERY problem defined by Elkind et al. [EFS09].

\mathcal{E} -SWAP BRIBERY	
Given:	A set of candidates C , a list of votes V that are linear orders over C , a list of swap-bribery price functions π , a budget $B \in \mathbb{N}$, and a designated candidate $c \in C$
Question:	Is there a sequence of admissible swaps such that the sum of the prices is lower than B such that c is a winner of the election under voting system \mathcal{E} ?

A swap-bribery price function π_i for voter $v_i \in V$ indicates for each pair (c_i, c_j) , $c_i, c_j \in C$, the price the briber has to pay for a swap of the given candidates in his vote. A swap is called admissible, if the considered candidates are adjacent in the vote. To swap candidates that are further away the briber has to execute a sequence of admissible swaps.

Elkind et al. [EFS09] show that POSSIBLE WINNER is a special case of SWAP BRIBERY. Summing up, this leads to the following hierarchy of problems shown in Figure 3.3, where an arrow from A to B means that there is a polynomial-time many-one reduction from A to B .

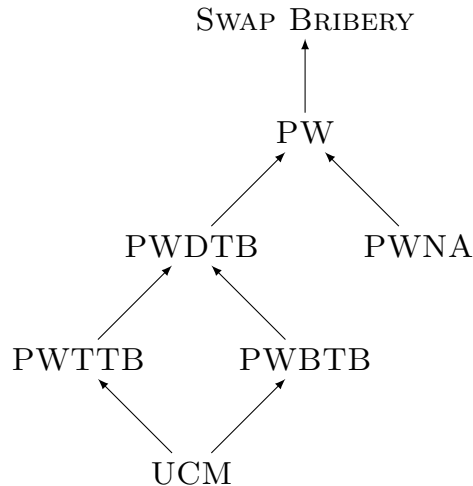


Figure 3.3: A hierarchy of possible winner and related problems

Having these reductions at hand, it immediately follows that all NP-hardness results are inherited upwards, and all polynomial-time algorithms are inherited downwards. For example the NP-hardness for SWAP BRIBERY for the pure scoring rule with the vector $(2, 1, \dots, 1, 0)$ follows immediately from the NP-hardness of PW for this scoring rule, showed in Section 3.2.

Another problem that is related to POSSIBLE WINNER WITH RESPECT TO THE ADDITION OF NEW ALTERNATIVES is the problem of CONSTRUCTIVE CONTROL BY ADDING CANDIDATES, (CCAC for short), see Bartholdi et al. [BTT92] and

Hemaspaandra et al. [HHR07]. Here a set C of candidates and a set D of spoiler candidates are given, along with a list of votes V that are linear orders over $C \cup D$ and a designated candidate $c \in C$. The question is whether there is a subset D' of the spoiler candidates that can be added into the election such that c is a winner of the election $(C \cup D', V)$. In contrast to the PCWNA problem not all of the new candidates must be added to the votes, and the positions of the new candidates are fixed in advance due to the given linear orders over $C \cup D$.

Also related to PCWNA but yet different is cloning in elections, see Elkind et al. [EFS10]. In cloning problems the question is whether the victory of a distinguished candidate can be achieved by cloning candidates, where all clones of one existing candidate will have subsequent positions in the votes. Hence the new candidates cannot be inserted at any position in the votes as it is the case for PCWNA.

The possible winner variants where the voters' weights are unknown are not directly related to the above mentioned problems. However, in Section 3.5 we showed that one variant of POSSIBLE WINNER WITH UNCERTAIN WEIGHTS is related to CONSTRUCTIVE CONTROL BY ADDING VOTERS.

In the last possible winner problem, considered in Section 3.6, the voting system used to determine the winner is initially unknown, and there is no obvious connection to the other possible winner problems studied in this chapter. But as mentioned earlier a specific variant of the SWAP BRIBERY problem is related to our study of PWUVS with respect to preference-based approval elections. Furthermore, Elkind and Erdélyi [EE12] study manipulation problems where the voting rule is unknown.

3.8 Summary

This chapter concludes with a short summary of the obtained results. Table 3.4 provides an overview of all results that are new to this chapter.

In Section 3.2 we completed the dichotomy result from Betzler and Dorn [BD10] by showing that POSSIBLE WINNER is NP-hard for the pure scoring rule with the vector $(2, 1, \dots, 1, 0)$. The POSSIBLE WINNER problem is NP-hard for all pure scoring rules except plurality and veto. Chevaleyre et al. [CLMM10] raised the question if such a dichotomy result could also be obtained for the related PCWNA problem. Until now the complexity is not yet completely settled and the question remains open. In Section 3.3 we made one step further by showing NP-completeness for PCWNA for the class of pure scoring rules defined by the vector $(\alpha_1, \alpha_2, 1, 0, \dots, 0)$ if one new candidate is added. Furthermore, we started to study the weighted version of PCWNA, and obtained NP-completeness for k -approval elections for each $k \geq 1$, if one new candidate is added. In the case of plurality, NP-completeness still holds if there are only two initial and one new candidate.

Section 3.4 introduces the POSSIBLE WINNER WITH DOUBLY/TOP/BOTTOM TRUNCATED BALLOTS problems and provides a polynomial time algorithm for k -approval elections, for each $k \geq 1$.

The POSSIBLE WINNER WITH UNCERTAIN WEIGHTS problem is investigated in several variants in Section 3.5. It was shown that for plurality, veto, and 2-approval elections all studied variants are solvable in polynomial time. The variants PCWUW and PCWUW-RW are polynomial-time solvable even for k -approval for each $k \geq 1$, whereas PCWUW-BW-RW and PCWUW-BW are NP-complete for k -approval for each $k \geq 4$. However, the complexity of PCWUW-BW-RW and PCWUW-BW for 3-approval remains open. For Copeland $^\alpha$ elections all variants of PCWUW are shown to be NP-complete for each rational value of α , $0 \leq \alpha \leq 1$.

Finally, Section 3.6 deals with the POSSIBLE WINNER UNDER UNCERTAIN VOTING SYSTEM problem. This problem is NP-complete for the subclass of scoring rules for an m -candidate election defined through vectors of the form $(\alpha_1, \dots, \alpha_{m-4}, x_1, x_2, x_3, 0)$, where $x_i = 1$ for at least one $i \in \{1, 2, 3\}$, and $m \geq 4$, if succinct representation is assumed. For the class of Copeland $^\alpha$, $0 \leq \alpha \leq 1$, and preference-based approval voting polynomial time algorithms are provided.

As a future research direction we propose to tackle the problem of finding a dichotomy theorem for the class of pure scoring rules for the unweighted version of the COALITIONAL MANIPULATION problem and the PCWNA problem. The study of the weighted version of the PCWNA problem has been initiated, but there are many voting rules for which the complexity is still unknown. Furthermore for most problems we only studied the co-winner case. Do the results carry over to the unique-winner variants or are they different? For PCWNA we only showed NP-hardness in the case where one new candidate is to be added. Chevaleyre et al. [CLMM10] remarked that the problem becomes easy if an unbounded number of candidates is to be added. But what about adding more than one new candidate?

We introduced PCWUVS and PWUVS, where the uncertainty does not lie in the candidates and/or preferences but in the voting system itself. We obtained results for a subclass of scoring rules, Copeland $^\alpha$ elections, and preference-based approval voting. Studying this problem for other natural classes of pure scoring rules would be very interesting. For example, for all voting systems sharing some important social choice theoretic property, e.g., the class of all systems that respect Condorcet winners.

Table 3.4: Overview of results for various possible winner problems

Problem Type	Voting System	Result
PW, PCW	$(2, 1, \dots, 1, 0)$	NP-comp., Thm. 3.1
PCWNA	$(\alpha_1, \alpha_2, 1, 0, \dots, 0),$ $\alpha_1 > \alpha_2 > 1, C' = 1$	NP-comp., Thm. 3.2
	plurality, $ C = 2, C' = 1,$ weighted voters	NP-comp., Thm. 3.3
	2-approval, $ C' = 1,$ weighted voters	NP-comp., Thm. 3.4
	k -approval, $k \geq 3, C' = 1,$ weighted voters	NP-comp., Cor. 3.1
PCWDTB	k -approval, $k \geq 1$	in P, Thm. 3.5
PCWTTB	k -approval, $k \geq 1$	in P, Thm. 3.5
PCWBTB	k -approval, $k \geq 1$	in P, Thm. 3.5
PCWUW	plurality, veto, k -approval, $k \geq 1$	in P, Prop. 3.2
	Copeland ^{α}	NP-comp., Thm. 3.7
PCWUW-BW	plurality, veto, 2-approval	in P, Prop. 3.2
	k -approval, $k \geq 4$	NP-comp., Thm. 3.6
	Copeland ^{α}	NP-comp., Thm. 3.7
PCWUW-RW	plurality, veto, k -approval, $k \geq 1$	in P, Prop. 3.2
	Copeland ^{α}	NP-comp., Thm. 3.7
PCWUW-BW-RW	plurality, veto, 2-approval	in P, Prop. 3.2
	k -approval, $k \geq 4$	NP-comp., Thm. 3.6
	Copeland ^{α}	NP-comp., Thm. 3.7
PCWUVS, PWUVS	$(\alpha_1, \dots, \alpha_{m-4}, x_1, x_2, x_3, 0),$ $\exists i \in \{1, 2, 3\}$ such that $x_i = 1$	NP-comp., Thm. 3.8
	Copeland ^{α}	in P, Thm. 3.9

4 Upward and Downward Covering

In this chapter we will introduce the solution concepts based on the upward and downward covering relations and study the computational complexity of several problems related to these solution concepts. The results have been published in [BBF⁺10] and are accepted for publication in [BBF⁺12].

The computational complexity of problems related to solution concepts has been addressed recently [Woe03, Alo06, Con06, BFH09, BFHM10, BF08]. Considering upward and downward covering Brandt and Fischer [BF08] showed that it is NP-hard to decide whether a given alternative is contained in some minimal upward covering set and whether a given alternative is contained in some minimal downward covering set. The upper bound was left open in both cases. We improve the lower bound to the θ_2^p level of the polynomial hierarchy, and provide an upper bound of Σ_2^p . In addition to these two problems we will also introduce and study other problems related to minimal and minimum-size upward and downward covering sets. Brandt and Fischer [BF08] also studied bidirectional covering sets and have shown that they are computable in polynomial time, whereas our results imply that neither minimal upward covering sets nor minimal downward covering sets (even when guaranteed to exist) can be found in polynomial time, unless $P = NP$. Even though the corresponding decision problem for minimal upward covering sets is trivially in P , since minimal upward covering sets are guaranteed to exist.

Parts of our results are obtained by applying Wagner's method [Wag87], which has also been applied in various other contexts (see, e.g., [Wag87, HHR97a, HR98, HW02, HRS06]). But to the best of our knowledge we apply this method for the first time to problems defined in terms of minimality rather than minimum-size.

4.1 Definitions and Notation

We will study the computational complexity of several problems related to the minimal upward covering set and to the minimal downward covering set. Before the solution concepts of minimal upward and downward covering sets can be introduced some further definitions are required.

For a finite set A of alternatives, we denote by the asymmetric and irreflexive relation $\succ \subseteq A \times A$ the dominance relation on A . Transitivity or completeness of the relation is not needed in general. We say that an alternative x dominates an alternative y if $(x, y) \in \succ$ (or equivalently $x \succ y$), which means that alternative

x is strictly preferred to alternative y . In the case of the majority relation, x dominates y if a strict majority prefers x to y . As mentioned in Chapter 2.2 a common representation of a dominance relation \succ on a set A of alternatives is the dominance graph denoted by (A, \succ) , where the edges are the alternatives of A , and for each $x, y \in A$ with $x \succ y$ there is a directed edge from x to y . The definition of minimal upward and downward covering sets relies on the upward and downward covering relation and on the uncovered set.

Definition 4.1 (Upward and Downward Covering Relation). *Let A be a finite set of alternatives, let $B \subseteq A$, and let $\succ \subseteq A \times A$ be a dominance relation on A . For any two alternatives x and y in B , define the following covering relations (see, e.g., [Fis77, Mil80, Bor83]):*

- x upward covers y in B , denoted by $x C_u^B y$, if $x \succ y$ and for all $z \in B$, $z \succ x$ implies $z \succ y$, and
- x downward covers y in B , denoted by $x C_d^B y$, if $x \succ y$ and for all $z \in B$, $y \succ z$ implies $x \succ z$.

When the subset B is clear from the context, we simply write $x C_u y$ and $x C_d y$ rather than $x C_u^B y$ and $x C_d^B y$ and omit mentioning “in B ” explicitly.

Definition 4.2 (Uncovered Set). *Let A be a set of alternatives, let $B \subseteq A$ be any subset, let \succ be a dominance relation on A , and let C be a covering relation on A based on \succ . The uncovered set of B with respect to C is defined as*

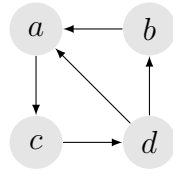
$$UC_C(B) = \{x \in B \mid y C x \text{ for no } y \in B\}.$$

For the sake of readability we will denote the upward uncovered set of B by $UC_u(B)$, and the downward uncovered set of B by $UC_d(B)$.

Figure 4.1 shows the same graph as Figure 2.1 in Chapter 2.2. In this example b upward covers a since the only alternative which dominates b is d , and d also dominates a . Furthermore d downward covers b since a is the only alternative dominated by b and is also dominated by d . The upward, respectively downward, uncovered set contains all alternatives which are not upward, respectively downward covered by another alternatives. This implies that in our example, the upward uncovered set is $UC_u(A) = \{b, c, d\}$ and the downward uncovered set is $UC_d(A) = \{a, c, d\}$.

Since the upward and the downward covering relation are transitive, the corresponding uncovered sets are nonempty for every nonempty set of alternatives.

The solution concepts we are interested in are minimal covering sets based on the upward and downward covering relation. Such minimal covering sets have two stability properties, external and internal stability. Internal stability means that there is no reason against any alternative in the minimal covering set to exclude

Figure 4.1: Dominance graph (A, \succ)

it, and external stability means that there is no reason to include any alternative outside the set. The stability criteria for the minimal upward and downward covering sets are defined through the corresponding uncovered sets.

Definition 4.3 (Minimal Covering Set). *Let A be a set of alternatives, let \succ be a dominance relation on A , and let C be a covering relation based on \succ . A subset $B \subseteq A$ is a covering set for A under C if the following two properties hold:*

- Internal stability: $UC_C(B) = B$.
- External stability: For all $x \in A \setminus B$, $x \notin UC_C(B \cup \{x\})$.

A covering set M for A under C is said to be (inclusion-)minimal if no $M' \subset M$ is a covering set for A under C .

Minimal upward covering sets are guaranteed to exist, and every upward uncovered set contains one or more minimal upward covering sets. In contrast, minimal downward covering sets may not always exist [BF08]. In tournaments (i.e., complete dominance graphs) the set of alternatives dominating a given alternative x are exactly those alternatives that are not dominated by x , hence both notions of covering coincide in this case. For the graph shown in Figure 4.1 the unique minimal upward covering set is $\{b, c\}$, and the unique minimal downward covering set is $\{a, c, d\}$.

Brandt and Fischer [BF08] considered the inclusion-minimal upward and downward covering sets, we also consider minimum-size upward and downward covering sets. Obviously every minimum-size upward, respectively downward, covering set is also an inclusion-minimal upward, respectively downward, covering set. Many classical problems in complexity theory are also defined through cardinality, like maximum-size independent set, or minimum-size dominating set for example. Some standard techniques used to study the complexity of problems where minimality is defined through cardinality are not directly applicable to set-inclusion minimal problems. To the best of our knowledge we are the first to apply Wagner's technique (see [Wag87]) to set-inclusion minimal problems. Regarding the results, for some of our problems we obtain different complexities for the minimum-size and inclusion-minimal variants of the problem.

Now we are ready to formally define the different problems we will study for upward and downward covering sets. The problem of deciding whether a given

alternative is contained in some minimal upward or some minimal downward covering set was considered in [BF08]. For minimal upward covering sets, the problem is formally defined as follows.

MC _u -MEMBER	
Given:	A set A of alternatives, a dominance relation \succ on A , and a distinguished element $d \in A$.
Question:	Is d contained in some minimal upward covering set for A ?

The corresponding problem MC_d-MEMBER is defined analogously for minimal downward covering sets. In addition to this problem we will also study the complexity of five further problems, which we will now define for minimal upward covering sets. The first one is MC_u-MEMBER-ALL and asks if a given alternative is contained in all minimal upward covering sets.

MC _u -MEMBER-ALL	
Given:	A set A of alternatives, a dominance relation \succ on A , and a distinguished element $d \in A$.
Question:	Is d contained in all minimal upward covering set for A ?

MC_u-SIZE asks if there is a minimal upward covering set of at most a given size.

MC _u -SIZE	
Given:	A set A of alternatives, a dominance relation \succ on A , and a positive integer k .
Question:	Does there exist some minimal upward covering set for A containing at most k alternatives?

The problem of deciding whether there is a unique minimal upward covering set is called MC_u-UNIQUE.

MC _u -UNIQUE	
Given:	A set A of alternatives and a dominance relation \succ on A .
Question:	Does there exist a unique minimal upward covering set for A ?

The last decision problem we consider is MC_u-TEST, the problem of deciding whether a given subset of the alternatives is a minimal upward covering set.

MC _u -TEST	
Given:	A set A of alternatives, a dominance relation \succ on A , and a subset $M \subseteq A$.
Question:	Is M a minimal upward covering set for A ?

In addition to the above defined decision problems we will also consider the search problem $\text{MC}_u\text{-FIND}$.

$\text{MC}_u\text{-FIND}$	
Given:	A set A of alternatives and a dominance relation \succ on A .
Find:	A minimal upward covering set for A .

The corresponding problems for downward covering sets are defined analogously and will be denoted by $\text{MC}_d\text{-MEMBER-ALL}$, $\text{MC}_d\text{-SIZE}$, $\text{MC}_d\text{-UNIQUE}$, $\text{MC}_d\text{-TEST}$, and $\text{MC}_d\text{-FIND}$. In addition to the problems defined for inclusion-minimal upward and downward covering sets we will also study these problems for minimum-size upward and downward covering sets. The minimum-size upward covering set problems will be denoted by $\text{MSC}_u\text{-MEMBER}$, $\text{MSC}_u\text{-MEMBER-ALL}$, $\text{MSC}_u\text{-SIZE}$, $\text{MSC}_u\text{-UNIQUE}$, $\text{MSC}_u\text{-TEST}$, and $\text{MSC}_u\text{-FIND}$, and the minimum-size downward covering set problems by $\text{MSC}_d\text{-MEMBER}$, $\text{MSC}_d\text{-MEMBER-ALL}$, $\text{MSC}_d\text{-SIZE}$, $\text{MSC}_d\text{-UNIQUE}$, $\text{MSC}_d\text{-TEST}$, and $\text{MSC}_d\text{-FIND}$. Thus we study 24 different problems in total. We will not only show hardness and completeness for NP and coNP, but also for the class Θ_2^p . To obtain the Θ_2^p -hardness results we will apply Wagner's Lemma, which is stated here as Lemma 4.1.

Lemma 4.1. *Let S be some NP-complete problem and let T be any set. If there exists a polynomial-time computable function f such that, for all $m \geq 1$ and all strings x_1, x_2, \dots, x_{2m} satisfying that if $x_j \in S$ then $x_{j-1} \in S$, $1 < j \leq 2m$, we have*

$$|\{i \mid x_i \in S\}| \text{ is odd} \Leftrightarrow f(x_1, x_2, \dots, x_m) \in T, \quad (4.1)$$

then T is Θ_2^p -hard.

In addition Wagner [Wag87] proved appropriate analogs of Lemma 4.1 for each level of the boolean hierarchy. In particular, the analogous criterion for DP-hardness is obtained by using the wording of Lemma 4.1 except with the value of $m = 1$ being fixed. Note that DP is the second level of the boolean hierarchy over NP (see Cai et al. [CGH⁺88, CGH⁺89]).

The sufficient condition provided by Wagner for proving Θ_2^p -hardness was useful in various other contexts. For example, the problem of testing whether the size of a maximum clique in a given graph is an odd number, the problem of deciding whether two given graphs have minimum vertex covers of the same size, and the problem of recognizing those graphs for which certain heuristics yield good approximations for the size of a maximum independent set or for the size of a minimum vertex cover each are known to be complete for Θ_2^p (see [Wag87, HR98, HRS06]). Hemaspaandra and Wechsung [HW02] proved that the minimization problem for boolean formulas is Θ_2^p -hard. In the field of computational social choice, the winner problems for Dodgson [Dod76], Young [You77],

and Kemeny [Kem59] elections have been shown to be Θ_2^p -complete in the nonunique-winner model [HHR97a, RSV03, HSV05], and also in the unique-winner model [HHR08].

In contrast with those previous results, however, one subtlety in our construction is due to the fact that we consider not only minimum-size but also (inclusion-)minimal covering sets. To the best of our knowledge, our Construction 4.2 and Construction 4.4, which will be presented in Chapter 4.3 and Chapter 4.4, for the first time apply Wagner’s technique [Wag87] to problems defined in terms of minimality/maximality rather than minimum/maximum size of a solution. For example, recall Wagner’s Θ_2^p -completeness result for testing whether the size of a maximum clique in a given graph is an odd number [Wag87]. One key ingredient in his proof is to define an associative operation on graphs, \bowtie , such that for any two graphs G and H , the size of a maximum clique in $G \bowtie H$ equals the sum of the sizes of a maximum clique in G and one in H . This operation is quite simple: Just connect every vertex of G with every vertex of H . In contrast, since minimality for minimal upward covering sets is defined in terms of set inclusion, it is not at all obvious how to define a similarly simple operation on dominance graphs such that the minimal upward covering sets in the given graphs are related to the minimal upward covering sets in the connected graph in a similarly useful way.

4.2 Results and Discussion

Results: Brandt and Fischer [BF08] proved that it is NP-hard to decide whether a given alternative is contained in some minimal unidirectional covering set. Using the notation of this chapter, their results state that the problems $\text{MC}_u\text{-MEMBER}$ and $\text{MC}_d\text{-MEMBER}$ are NP-hard. The question of whether these two problems are NP-complete or of higher complexity was left open in [BF08]. Our contribution is

1. to raise Brandt and Fischer’s NP-hardness lower bounds for $\text{MC}_u\text{-MEMBER}$ and $\text{MC}_d\text{-MEMBER}$ to Θ_2^p -hardness and to provide (simple) Σ_2^p upper bounds for these problems, and
2. to extend the techniques we developed to apply also to the 22 other covering set problems defined in Section 4.1, in particular to the search problems.

Our results are stated in the following theorem.

Theorem 4.1. *The complexity of the covering set problems defined in Section 4.1 is as shown in Table 4.1 for the upward covering set problems and as shown in Table 4.2 for the downward covering set problems.*

Problem Type	MC_u	MSC_u
SIZE	NP-complete	NP-complete
MEMBER	Θ_2^p -hard and in Σ_2^p	Θ_2^p -complete
MEMBER-ALL	coNP-complete [BF08]	Θ_2^p -complete
UNIQUE	coNP-hard and in Σ_2^p	coNP-hard and in Θ_2^p
TEST	coNP-complete	coNP-complete
FIND	not in polynomial time unless $P = NP$	not in polynomial time unless $P = NP$

Table 4.1: Results for minimal and minimum-size upward covering set problems. As indicated, one result is due to Brandt and Fischer [BF08]; all other results will be proven in this thesis.

Problem Type	MC_d	MSC_d
SIZE	NP-complete	NP-complete
MEMBER	Θ_2^p -hard and in Σ_2^p	coNP-hard and in Θ_2^p
MEMBER-ALL	coNP-complete [BF08]	coNP-hard and in Θ_2^p
UNIQUE	coNP-hard and in Σ_2^p	coNP-hard and in Θ_2^p
TEST	coNP-complete	coNP-complete
FIND	not in polynomial time unless $P = NP$ (follows from [BF08])	not in polynomial time unless $P = NP$

Table 4.2: Results for minimal and minimum-size downward covering set problems. As indicated, two results are due to Brandt and Fischer [BF08]; all other results will be proven in this thesis.

The detailed proofs of the single results collected in Theorem 4.1 will be presented in Section 4.3.2 for the upward covering set problems and in Section 4.4.2 for the downward covering set problems.

Discussion: We consider the problems of *finding* minimal and minimum-size upward and downward covering sets ($\text{MC}_u\text{-FIND}$, $\text{MC}_d\text{-FIND}$, $\text{MSC}_u\text{-FIND}$, and $\text{MSC}_d\text{-FIND}$) to be particularly important and natural.

Regarding upward covering sets, we stress that our results (see Theorem 4.1) that, assuming $P \neq NP$, $\text{MC}_u\text{-FIND}$ and $\text{MSC}_u\text{-FIND}$ are hard to compute does not seem to follow directly from the NP-hardness of $\text{MC}_u\text{-MEMBER}$ in any obvious way. The decision version of $\text{MC}_u\text{-FIND}$ is: Given a dominance graph, does it contain a minimal upward covering set? However, this question has always an affirmative answer, so the decision version of $\text{MC}_u\text{-FIND}$ is trivially in P . Note also that $\text{MC}_u\text{-FIND}$ can be reduced in a “disjunctive truth-table” fashion to the search version of $\text{MC}_u\text{-MEMBER}$ (“Given a dominance graph (A, \succ) and an alternative $d \in A$, find some minimal upward covering set for A that contains d ”) by asking this oracle set about all alternatives in parallel.¹ So $\text{MC}_u\text{-FIND}$ is no harder (with respect to disjunctive truth-table reductions) than that problem. The converse, however, is not at all obvious. Brandt and Fischer’s result only implies the hardness of finding an alternative that is contained in *all* minimal upward covering sets [BF08]. Our reduction that raises the lower bound of $\text{MC}_u\text{-MEMBER}$ from NP-hardness to Θ_2^P -hardness, however, also allows us to prove that $\text{MC}_u\text{-FIND}$ and $\text{MSC}_u\text{-FIND}$ cannot be solved in polynomial time unless $P = NP$.

Regarding downward covering sets, the result that $\text{MC}_d\text{-FIND}$ cannot be computed in polynomial time unless $P = NP$ is an immediate consequence of Brandt and Fischer’s result that it is NP-complete to decide whether there exists a minimal downward covering set [BF08, Thm. 9]. We provide an alternative proof based on our reduction showing that $\text{MC}_d\text{-MEMBER}$ is Θ_2^P -hard (see the proof of Theorem 4.16). In contrast to Brandt and Fischer’s proof, our proof shows that $\text{MC}_d\text{-FIND}$ is hard to compute even when the existence of a (minimal) downward covering set is guaranteed. As indicated in Table 4.2, coNP-hardness of $\text{MC}_u\text{-MEMBER-ALL}$ and $\text{MC}_d\text{-MEMBER-ALL}$ was also shown previously by Brandt and Fischer [BF08].

As mentioned above, the two problems $\text{MC}_u\text{-MEMBER}$ and $\text{MC}_d\text{-MEMBER}$ were already known to be NP-hard [BF08] and are here shown to be even Θ_2^P -hard. One

¹This type of reduction was introduced by Ladner et al. [LLS75]. Informally stated, a *disjunctive truth-table reduction* between two decision problems X and Y computes, given an instance x , in polynomial time k queries y_1, y_2, \dots, y_k such that $x \in X$ if and only if $y_i \in Y$ for at least one i , $1 \leq i \leq k$. This reduction can be adapted straightforwardly to function problems F and G : F *disjunctively truth-table reduces to* G if, given an instance x , in polynomial time we can compute k queries y_1, y_2, \dots, y_k such that $F(x)$ can be computed from $G(y_i)$ for at least one i , $1 \leq i \leq k$.

may naturally wonder whether raising their (or any problem's) lower bound from NP-hardness to Θ_2^p -hardness gives us any more insight into the problem's inherent computational complexity. After all, $P = NP$ if and only if $P = \Theta_2^p$. However, this question is a bit more subtle than that and has been discussed carefully by Hemaspaandra et al. [HHR97b]. They make the case that the answer to this question crucially depends on what one considers to be the most natural computational model. In particular, they argue that raising NP-hardness to Θ_2^p -hardness potentially (i.e., unless longstanding open problems regarding the separation of the corresponding complexity classes could be solved) is an improvement in terms of randomized polynomial time and in terms of unambiguous polynomial time [HHR97b].

4.3 Minimal and Minimum-Size Upward Covering Sets

We will first focus on problems related to minimal and minimum-size upward covering sets. In Section 4.3.1 we provide the constructions that will be used in Section 4.3.2 to prove our results for minimal and minimum-size upward covering set problems.

4.3.1 Constructions for Minimal and Minimum-Size Upward Covering Sets

The results for the different problems are all obtained by the constructions presented in this chapter. Since the new constructions relies on the proof of the following theorem by Brandt and Fischer [BF08], we first give a proof sketch of their theorem.

Theorem 4.2 (Brandt and Fischer [BF08]). *Deciding whether a designated alternative is contained in some minimal upward covering set for a given dominance graph is NP-hard. That is, MC_u -MEMBER is NP-hard.*

Proof Sketch. To show NP-hardness, a reduction from the NP-complete boolean satisfiability problem, SAT, is given (for a formal definition of this problem, see Chapter 2.4). Based on a boolean formula in conjunctive normal form, $\varphi(v_1, v_2, \dots, v_n) = c_1 \wedge c_2 \wedge \dots \wedge c_r$, over the set $V = \{v_1, v_2, \dots, v_n\}$ of variables, we construct an instance of MC_u -MEMBER with the dominance graph (A, \succ) and the distinguished alternative d , whose membership in some minimal upward covering set for A is to be decided. The set of alternatives is

$$A = \{x_i, \bar{x}_i, x'_i, \bar{x}'_i \mid v_i \in V\} \cup \{y_j \mid c_j \text{ is a clause in } \varphi\} \cup \{d\},$$

and the dominance relation is defined by:

- For each i , $1 \leq i \leq n$ there is a cycle $x_i \succ \bar{x}_i \succ x'_i \succ \bar{x}'_i \succ x_i$;
- if variable v_i occurs in clause c_j as a positive literal, then $x_i \succ y_j$;
- if variable v_i occurs in clause c_j as a negative literal, then $\bar{x}_i \succ y_j$; and
- for each j , $1 \leq j \leq r$, we have $y_j \succ d$.

Then it holds that there is a satisfying assignment for the boolean formula φ if and only if there is a minimal upward covering set for (A, \succ) which contains alternative d .

We will not give a formal proof, but rather give an example for the reduction. Consider the boolean formula $(v_1 \vee \neg v_2) \wedge (v_1 \vee \neg v_2 \vee \neg v_3)$. The resulting dominance graph with the set of alternatives $A = \{x_i, \bar{x}_i, x'_i, \bar{x}'_i \mid 1 \leq i \leq 3\} \cup \{y_1, y_2\} \cup \{d\}$ and the dominance relation \succ is shown in Figure 4.2. A satisfying truth assignment for φ is obtained for example by setting v_1, v_2 , and v_3 to false. The corresponding minimal upward covering set for A contains the alternatives $\{\bar{x}_1, \bar{x}'_1, \bar{x}_2, \bar{x}'_2, \bar{x}_3, \bar{x}'_3\} \cup \{d\}$, and as desired, the alternative d is contained in this minimal upward covering set. \square

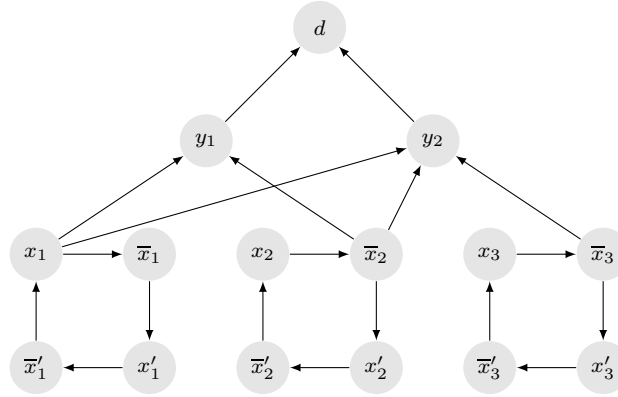


Figure 4.2: Dominance graph (A, \succ) for Theorem 4.2 resulting from the boolean formula $(v_1 \vee \neg v_2) \wedge (v_1 \vee \neg v_2 \vee \neg v_3)$

Our construction used to show coNP- and Θ_2^P -hardness for upward covering set problems relies on the construction used by Brandt and Fischer [BF08] in the proof of Theorem 4.2, therefore we will first analyze the minimal upward covering sets of the resulting dominance graph. Independent of the underlying boolean formula, Brandt and Fischer [BF08] showed, that each minimal covering set for A contains exactly two of the four alternatives corresponding to any of the variables. More precisely, it holds that either x_i and x'_i , or \bar{x}_i and \bar{x}'_i , $1 \leq i \leq n$, are contained in each minimal upward covering set for A . Since we will also analyze minimum-size upward covering sets we will have a closer look at the number of alternatives

contained in each minimal upward covering set for A . We now assume that if φ is not satisfiable then for each truth assignment to the variables of φ , at least two clauses are unsatisfied (which can be ensured, if needed, by adding two dummy variables). Due to the property described above, each minimal upward covering set contains $2n$ alternatives from the cycles corresponding to the variables. If φ is not satisfiable, at least 2 more alternatives from the unsatisfied clauses are also contained in each minimal upward covering set. Whereas each minimal upward covering set that contains the alternative d consists of exactly $2n + 1$ alternatives. Thus φ is satisfiable if and only if every minimum-size upward covering set consists of $2n + 1$ alternatives. Additionally, these minimum-size upward covering sets always include alternative d .

Now we will present a construction that can be used to show coNP-hardness for $\text{MC}_u\text{-MEMBER}$ and other upward covering problems. By merging this construction with the one by Brandt and Fischer [BF08] presented above, we can apply Wagner's Lemma to show Θ_2^P hardness for upward covering set problems.

Construction 4.1 (for coNP-hardness of upward covering set problems). *We again build a dominance graph (A, \succ) based on a boolean formula $\varphi(w_1, w_2, \dots, w_k) = f_1 \wedge f_2 \wedge \dots \wedge f_\ell$, over the set $W = \{w_1, w_2, \dots, w_k\}$ of variables, in conjunctive normal form. Without loss of generality, we may assume that if φ is satisfiable then it has at least two satisfying assignments. This can be ensured, if needed, by adding two dummy variables. For the constructed dominance graph, the set of alternatives is $A = \{u_i, \bar{u}_i, u'_i, \bar{u}'_i \mid w_i \in W\} \cup \{e_j, e'_j \mid f_j \text{ is a clause in } \varphi\} \cup \{a_1, a_2, a_3\}$, and the dominance relation \succ is defined by:*

- For each i , $1 \leq i \leq k$, there is a cycle $u_i \succ \bar{u}_i \succ u'_i \succ \bar{u}'_i \succ u_i$;
- if variable w_i occurs in clause f_j as a positive literal, then $u_i \succ e_j$, $u_i \succ e'_j$, $e_j \succ \bar{u}_i$, and $e'_j \succ \bar{u}_i$;
- if variable w_i occurs in clause f_j as a negative literal, then $\bar{u}_i \succ e_j$, $\bar{u}_i \succ e'_j$, $e_j \succ u_i$, and $e'_j \succ u_i$;
- if variable w_i does not occur in clause f_j , then $e_j \succ u'_i$ and $e'_j \succ \bar{u}'_i$;
- for each j , $1 \leq j \leq \ell$, we have $a_1 \succ e_j$ and $a_1 \succ e'_j$; and
- there is a cycle $a_1 \succ a_2 \succ a_3 \succ a_1$.

To analyze the minimal and minimum-size upward covering sets, Figure 4.3 shows some parts of the resulting dominance graph. In particular, Figure 4.3(a) shows that part of the graph that corresponds to some variable w_i occurring in clause f_j as a positive literal; Figure 4.3(b) shows that part of the graph that corresponds to some variable w_i occurring in clause f_j as a negative literal; and

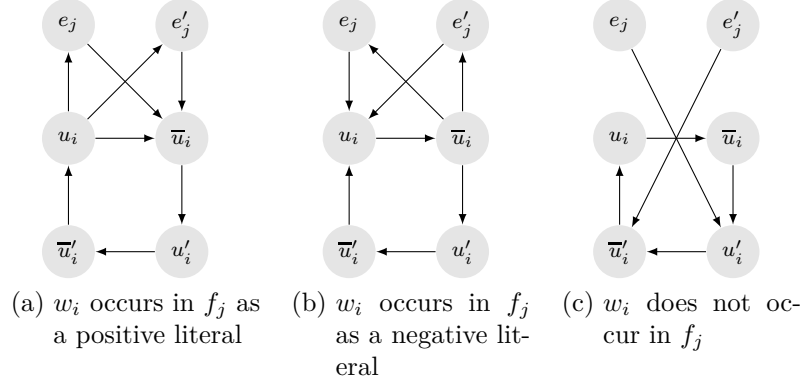

 Figure 4.3: Parts of the dominance graph (A, \succ) defined in Construction 4.1

Figure 4.3(c) shows that part of the graph that corresponds to some variable w_i not occurring in clause f_j .

Figure 4.4 shows the complete resulting dominance graph (A, \succ) for the boolean formula $(w_1 \wedge \neg w_2) \vee (\neg w_2 \wedge \neg w_3)$. This boolean formula can be satisfied for example by setting each of w_1 , w_2 , and w_3 to false. The corresponding minimal upward covering set for A is $M = \{\bar{u}_1, \bar{u}'_1, \bar{u}_2, \bar{u}'_2, \bar{u}_3, \bar{u}'_3, a_1, a_2, a_3\}$. Note that in M neither e_1 nor e_2 occurs, and none of them occurs in any other minimal upward covering set for A either. For alternative e_1 in the example shown in Figure 4.4, this can be seen as follows. First observe that the alternatives a_1 , a_2 , and a_3 are contained in every minimal upward covering set for A . If there were a minimal upward covering set M' for A containing e_1 (and thus also e'_1 , since they both are dominated by the same alternatives), then neither u_1 nor \bar{u}_2 (which dominate e_1) must upward cover e_1 in M' , so all alternatives corresponding to the variables w_1 and w_2 (i.e., $\{u_i, u'_i, \bar{u}_i, \bar{u}'_i \mid 1 \leq i \leq 2\}$) would also have to be contained in M' . Due to $e_1 \succ u'_3$ and $e'_1 \succ \bar{u}'_3$, all alternatives corresponding to w_3 (i.e., $\{u_3, u'_3, \bar{u}_3, \bar{u}'_3\}$) are in M' as well. Note that e_2 and e'_2 are no longer upward covered and must also be in M' . But then M' is not minimal because the upward covering set M , which corresponds to the satisfying assignment of the boolean formula stated above, is a strict subset of M' . Hence there can be no minimal upward covering set for A which contains alternative e_1 .

In the following we establish several properties of Construction 4.1, which will be used in the proofs of Section 4.3.2. In the example above we have already seen that if e_1 is contained in some minimal upward covering set, all other alternatives are in this minimal upward covering set as well. In its general version this property is stated in Claim 4.1.

Claim 4.1. *Consider the dominance graph (A, \succ) created by Construction 4.1, and fix any j , $1 \leq j \leq \ell$. For each minimal upward covering set M for A , if M contains the alternative e_j , then all other alternatives are contained in M as well*

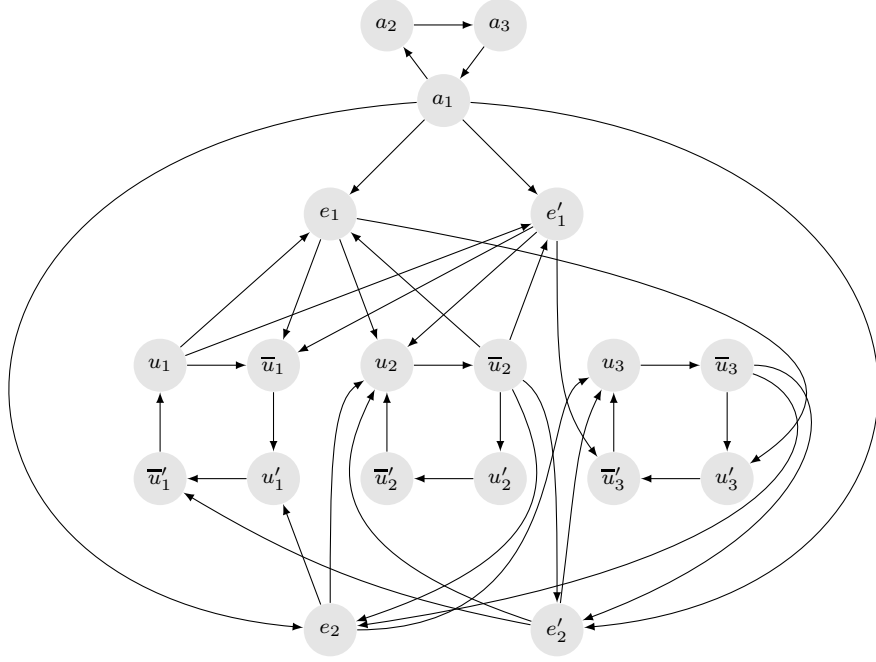


Figure 4.4: Dominance graph (A, \succ) from Construction 4.1 for the boolean formula $(w_1 \wedge \neg w_2) \vee (\neg w_2 \wedge \neg w_3)$

(i.e., $A = M$).

Proof. To simplify notation we prove the claim only for the case of $j = 1$. However, since there is nothing special about e_1 in our argument, the same property can be shown by an analogous argument for each j , $1 \leq j \leq \ell$.

The three alternatives a_1 , a_2 , and a_3 form an undominated three-cycle and thus are contained in every minimal upward covering set M for A .

Let M be any minimal upward covering set for A , and suppose that $e_1 \in M$. First note that the dominators of e_1 and e'_1 are always the same (albeit e_1 and e'_1 may dominate different alternatives). This implies that for each minimal upward covering set for A , either both e_1 and e'_1 are in it, or they both are not. Thus, since $e_1 \in M$, we have $e'_1 \in M$ as well. As noted above, $\{a_1, a_2, a_3\} \subseteq M$, and since a_1 is a dominator of e_j and e'_j , $1 \leq j \leq \ell$, but none of the alternatives dominated by e_j or e'_j is dominated by a_1 , no other alternative in M can be upward covered by e_j or e'_j . In particular, no alternative in any of the k four-cycles $u_i \succ \bar{u}_i \succ u'_i \succ \bar{u}'_i \succ u_i$ can be upward covered by an alternative e_j or e'_j , and so they must be upward covered within their cycle. In the construction by Brandt and Fischer [BF08] in their proof of Theorem 4.2 they showed that for each four-cycle it holds that either $\{x_i, \bar{x}_i\}$ or $\{x'_i, \bar{x}'_i\}$, $1 \leq i \leq n$, are contained in each minimal upward covering set, but not both. Analogously, in our construction every minimal upward covering set for A must contain at least one of the sets $\{u_i, \bar{u}_i\}$ and $\{u'_i, \bar{u}'_i\}$, since at least one is

needed to upward cover the other one. In contrast to the construction in the proof of Theorem 4.2, our construction also allows for both $\{u_i, \bar{u}_i\}$ and $\{u'_i, \bar{u}'_i\}$ being contained in some minimal upward covering set for A . Informally stated, the reason is that, unlike the four cycles in Figure 4.2, our four-cycles $u_i \succ \bar{u}_i \succ u'_i \succ \bar{u}'_i \succ u_i$ also have incoming edges.

Since $e_1 \in M$ and by internal stability, we have that no alternative from M upward covers e_1 . In addition to a_1 , the alternatives dominating e_1 are u_i (for each i such that w_i occurs as a positive literal in f_1) and \bar{u}_i (for each i such that w_i occurs as a negative literal in f_1).

First assume that, for some i , w_i occurs as a positive literal in f_1 . Suppose that $\{u_i, u'_i\} \subseteq M$. If $\bar{u}'_i \notin M$, then e_1 would be upward covered by u_i , which is impossible. Thus $\bar{u}'_i \in M$. But then $\bar{u}_i \in M$ as well, since u_i , the only alternative that could upward cover \bar{u}_i , is itself dominated by \bar{u}'_i . For the latter argument, recall that \bar{u}_i cannot be upward covered by any e_j or e'_j . Thus, we have shown that $\{u_i, u'_i\} \subseteq M$ implies $\{\bar{u}_i, \bar{u}'_i\} \subseteq M$. Conversely, suppose that $\{\bar{u}_i, \bar{u}'_i\} \subseteq M$. Then u'_i is no longer upward covered by \bar{u}_i and hence must be in M as well. The same holds for the alternative u_i , so $\{\bar{u}_i, \bar{u}'_i\} \subseteq M$ implies $\{u_i, u'_i\} \subseteq M$. Summing up, if $e_1 \in M$ then $\{u_i, u'_i, \bar{u}_i, \bar{u}'_i\} \subseteq M$ for each i such that w_i occurs as a positive literal in f_1 .

By symmetry of the construction, an analogous argument shows that if $e_1 \in M$ then $\{u_i, u'_i, \bar{u}_i, \bar{u}'_i\} \subseteq M$ for each i such that w_i occurs as a negative literal in f_1 .

Now, consider any i such that w_i does not occur in f_1 . We have $e_1 \succ u'_i$ and $e'_1 \succ \bar{u}'_i$. Again, none of the sets $\{u_i, u'_i\}$ and $\{\bar{u}_i, \bar{u}'_i\}$ alone can be contained in M , since otherwise either u_i or \bar{u}'_i would remain upward uncovered. Thus, $e_1 \in M$ again implies that $\{u_i, u'_i, \bar{u}_i, \bar{u}'_i\} \subseteq M$.

Now it is easy to see that, since $\bigcup_{1 \leq i \leq k} \{u_i, u'_i, \bar{u}_i, \bar{u}'_i\} \subseteq M$ and since a_1 cannot upward cover any of the e_j and e'_j , $1 \leq j \leq \ell$, external stability of M enforces that $\bigcup_{1 \leq j \leq \ell} \{e_j, e'_j\} \subseteq M$. Summing up, we have shown that if e_1 is contained in any minimal upward covering set M for A , then $M = A$. \square

The next claim shows a central property for establishing coNP-hardness for upward covering set problems.

Claim 4.2. *Consider Construction 4.1. The boolean formula φ is satisfiable if and only if there is no minimal upward covering set for A that contains any of the e_j , $1 \leq j \leq \ell$.*

Proof. It is enough to prove the claim for the case $j = 1$, since the other cases can be proven analogously. From left to right, suppose there is a satisfying assignment $\alpha : W \rightarrow \{0, 1\}$ for φ . Define the set

$$B_\alpha = \{a_1, a_2, a_3\} \cup \{u_i, u'_i \mid \alpha(w_i) = 0\}.$$

Since every upward covering set for A must contain $\{a_1, a_2, a_3\}$ and at least one of the sets $\{u_i, u'_i\}$ and $\{\bar{u}_i, \bar{u}'_i\}$ for each i , $1 \leq i \leq k$, B_α is a (minimal) upward covering set for A . Let M be an arbitrary minimal upward covering set for A . By Claim 4.1, if e_1 were contained in M , we would have $M = A$. But since $B_\alpha \subset A = M$, this contradicts the minimality of M . Thus $e_1 \notin M$.

From right to left, let M be an arbitrary minimal upward covering set for A and suppose $e_1 \notin M$. By Claim 4.1, if any of the e_j , $2 \leq j \leq \ell$, were contained in M , it would follow that $e_1 \in M$, a contradiction. Thus, $\{e_j \mid 1 \leq j \leq \ell\} \cap M = \emptyset$. It follows that each e_j must be upward covered by some alternative in M . It is easy to see that for each j , $1 \leq j \leq \ell$, and for each i , $1 \leq i \leq k$, e_j is upward covered in $M \cup \{e_j\} \supseteq \{u_i, u'_i\}$ if w_i occurs in f_j as a positive literal, and e_j is upward covered in $M \cup \{e_j\} \supseteq \{\bar{u}_i, \bar{u}'_i\}$ if w_i occurs in f_j as a negative literal. It can never be the case that all four alternatives, $\{u_i, u'_i, \bar{u}_i, \bar{u}'_i\}$, are contained in M , because then either e_j would no longer be upward covered in M or the resulting set M was not minimal. Now, M induces a satisfying assignment for φ by setting, for each i , $1 \leq i \leq k$, $\alpha(w_i) = 1$ if $u_i \in M$, and $\alpha(w_i) = 0$ if $\bar{u}_i \in M$. \square

Note that in Construction 4.1 every minimal upward covering set for A obtained from any satisfying assignment for φ contains exactly $2k+3$ alternatives, and there is no minimal upward covering set of smaller size for A when φ is satisfiable.

Now we turn to the case when the underlying boolean formula is not satisfiable, and show that in this case there is a unique minimal upward covering set for the resulting dominance graph.

Claim 4.3. *Consider Construction 4.1. The boolean formula φ is not satisfiable if and only if there is a unique minimal upward covering set for A .*

Proof. Recall that we assumed in Construction 4.1 that if φ is satisfiable then it has at least two satisfying assignments. From left to right, suppose there is no satisfying assignment for φ . By Claim 4.2, there must be a minimal upward covering set for A containing one of the e_j , $1 \leq j \leq \ell$, and by Claim 4.1 this minimal upward covering set for A must contain all alternatives. By reasons of minimality, there cannot be another minimal upward covering set for A . From right to left, suppose there is a unique minimal upward covering set for A . Due to our assumption that if φ is satisfiable then there are at least two satisfying assignments, φ cannot be satisfiable, since if it were, there would be two distinct minimal upward covering sets corresponding to these assignments (as argued in the proof of Claim 4.2). \square

Now we are ready to present the construction which will be used to apply Wagner's Lemma to upward covering set problems. In this construction, we define a dominance graph based on Construction 4.1 and the construction presented in the proof sketch of Theorem 4.2 such that Lemma 4.1 can be applied, making use of the properties established in Claims 4.1, 4.2, and 4.3.

Construction 4.2 (for applying Lemma 4.1 to upward covering set problems). We apply Wagner’s Lemma with the NP-complete problem $S = \text{SAT}$ and construct a dominance graph. Fix an arbitrary $m \geq 1$ and let $\varphi_1, \varphi_2, \dots, \varphi_{2m}$ be $2m$ boolean formulas in conjunctive normal form such that if φ_j is satisfiable then so is φ_{j-1} , for each j , $1 < j \leq 2m$. Without loss of generality, we assume that for each j , $1 \leq j \leq 2m$, the first variable of φ_j does not occur in all clauses of φ_j . Furthermore, we require φ_j to have at least two unsatisfied clauses if φ_j is not satisfiable, and to have at least two satisfying assignments if φ_j is satisfiable. It is easy to see that if φ_j does not have these properties, it can be transformed into a formula that does have them in polynomial time, without affecting the satisfiability of the formula.

We now define a polynomial-time computable function f , which maps the given $2m$ boolean formulas to a dominance graph (A, \succ) with useful properties for upward covering sets. Define $A = \bigcup_{j=1}^{2m} A_j$ and the dominance relation \succ on A by

$$\left(\bigcup_{j=1}^{2m} \succ_j \right) \cup \left(\bigcup_{i=1}^m \{(u'_{1,2i}, d_{2i-1}), (\bar{u}'_{1,2i}, d_{2i-1})\} \right) \cup \left(\bigcup_{i=2}^m \{(d_{2i-1}, z) \mid z \in A_{2i-2}\} \right)$$

where we use the following notation:

1. For each i , $1 \leq i \leq m$, let (A_{2i-1}, \succ_{2i-1}) be the dominance graph that results from the formula φ_{2i-1} according to Brandt and Fischer’s construction [BF08] given in the proof sketch of Theorem 4.2. We use the same names for the alternatives in A_{2i-1} as in that proof sketch, except that we attach the subscript $2i - 1$. For example, alternative d from the proof sketch of Theorem 4.2 now becomes d_{2i-1} , x_1 becomes $x_{1,2i-1}$, y_1 becomes $y_{1,2i-1}$, and so on.
2. For each i , $1 \leq i \leq m$, let (A_{2i}, \succ_{2i}) be the dominance graph that results from the formula φ_{2i} according to Construction 4.1. We use the same names for the alternatives in A_{2i} as in that construction, except that we attach the subscript $2i$. For example, alternative a_1 from Construction 4.1 now becomes $a_{1,2i}$, e_1 becomes $e_{1,2i}$, u_1 becomes $u_{1,2i}$, and so on.
3. For each i , $1 \leq i \leq m$, connect the dominance graphs (A_{2i-1}, \succ_{2i-1}) and (A_{2i}, \succ_{2i}) as follows. Let $u_{1,2i}, \bar{u}_{1,2i}, u'_{1,2i}, \bar{u}'_{1,2i} \in A_{2i}$ be the four alternatives in the cycle corresponding to the first variable of φ_{2i} . Then both $u'_{1,2i}$ and $\bar{u}'_{1,2i}$ dominate d_{2i-1} . The resulting dominance graph is denoted by (B_i, \succ_i^B) .
4. Connect the m dominance graphs (B_i, \succ_i^B) , $1 \leq i \leq m$, as follows: For each i , $2 \leq i \leq m$, d_{2i-1} dominates all alternatives in A_{2i-2} .

The dominance graph (A, \succ) is sketched in Figure 4.5. Clearly (A, \succ) is computable in polynomial time.

Before turning to the results for upward covering set problems, we will again show some useful properties of the constructed dominance graph. We first consider the

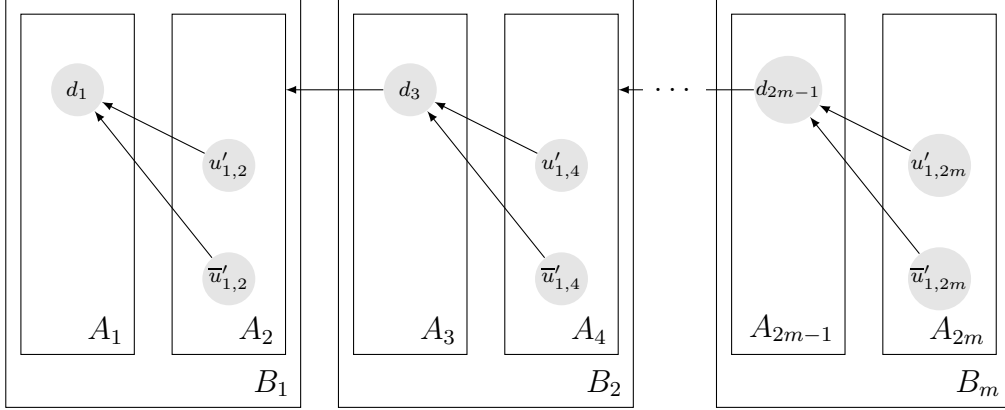


Figure 4.5: Dominance graph (A, \succ) from Construction 4.2. Most alternatives, and all edges between pairs of alternatives in A_j , $1 \leq j \leq 2m$, have been omitted. All edges between alternatives in A_i and alternatives in A_j for $i \neq j$ are shown. An edge incident to a set of alternatives represents an edge incident to *each* alternative in the set.

dominance graph (B_i, \succ_i^B) (see Step 3 in Construction 4.2) separately, for any fixed i with $1 \leq i \leq m$. Doing so will simplify our argument for the whole dominance graph (A, \succ) . In addition our argument about (B_i, \succ_i^B) can be used to show, in effect, DP-hardness of upward covering set problems (see the comments made after Wagner's Lemma, which is stated here as Lemma 4.1).

Claim 4.4. *Consider Construction 4.2. Alternative d_{2i-1} is contained in some minimal upward covering set for (B_i, \succ_i^B) if and only if φ_{2i-1} is satisfiable and φ_{2i} is not satisfiable.*

Proof. Distinguish the following three cases.

Case 1: $\varphi_{2i-1} \in \text{SAT}$ and $\varphi_{2i} \in \text{SAT}$. Since φ_{2i} is satisfiable, it follows from the proof of Claim 4.2 that for each minimal upward covering set M for (B_i, \succ_i^B) , either $\{u_{1,2i}, u'_{1,2i}\} \subseteq M$ or $\{\bar{u}_{1,2i}, \bar{u}'_{1,2i}\} \subseteq M$, but not both, and that none of the $e_{j,2i}$ and $e'_{j,2i}$ is in M . If $\bar{u}'_{1,2i} \in M$ but $u'_{1,2i} \notin M$, then $d_{2i-1} \notin \text{UC}_u(M)$, since $\bar{u}'_{1,2i}$ upward covers d_{2i-1} within M . Hence, by internal stability, d_{2i-1} is not contained in M .

Case 2: $\varphi_{2i-1} \notin \text{SAT}$ and $\varphi_{2i} \notin \text{SAT}$. Since $\varphi_{2i-1} \notin \text{SAT}$, it follows from the proof of Theorem 4.2 that each minimal upward covering set M for (B_i, \succ_i^B) contains at least one alternative $y_{j,2i-1}$ (corresponding to some clause of φ_{2i-1}) that upward covers d_{2i-1} . Thus d_{2i-1} cannot be in M , again by internal stability.

Case 3: $\varphi_{2i-1} \in \text{SAT}$ and $\varphi_{2i} \notin \text{SAT}$. Since $\varphi_{2i-1} \in \text{SAT}$, it follows from the proof of Theorem 4.2 that there exists a minimal upward covering set M' for (A_{2i-1}, \succ_{2i-1}) that corresponds to a satisfying truth assignment for φ_{2i-1} . In particular, none of the $y_{j,2i-1}$ is in M' . On the other hand, since $\varphi_{2i} \notin \text{SAT}$, it follows from Claim 4.3 that A_{2i} is the only minimal upward covering set for (A_{2i}, \succ_{2i}) . Define $M = M' \cup A_{2i}$. It is easy to see that M is a minimal upward covering set for (B_i, \succ_i^B) , since the only edges between A_{2i-1} and A_{2i} are those from $\bar{u}'_{1,2i}$ and $u'_{1,2i}$ to d_{2i-1} , and both $\bar{u}'_{1,2i}$ and $u'_{1,2i}$ are dominated by elements in M not dominating d_{2i-1} .

Note that by our assumption on how the formulas are ordered, the fourth case (i.e., $\varphi_{2i-1} \notin \text{SAT}$ and $\varphi_{2i} \in \text{SAT}$) cannot occur, thus the proof is complete. \square

Now we show that the minimal upward covering sets for (B_i, \succ_i^B) are contained in the minimal upward covering sets for the whole dominance graph (A, \succ) .

Claim 4.5. *Consider Construction 4.2. For each i , $1 \leq i \leq m$, let M_i be a minimal upward covering set for (B_i, \succ_i^B) according to the cases in the proof of Claim 4.4. Then each of the sets M_i must be contained in every minimal upward covering set for (A, \succ) .*

Proof. The minimal upward covering set M_m for (B_m, \succ_m^B) must be contained in every minimal upward covering set for (A, \succ) , since no alternative in $A \setminus B_m$ dominates any alternative in B_m . On the other hand, for each i , $1 \leq i < m$, no alternative in B_i can be upward covered by d_{2i+1} (which is the only element in $A \setminus B_i$ that dominates any of the elements of B_i), since d_{2i+1} is dominated within every minimal upward covering set for B_{i+1} (and, in particular, within M_{i+1}). Thus, each of the sets M_i , $1 \leq i \leq m$, must be contained in every minimal upward covering set for (A, \succ) . \square

Finally, we show the key property of our construction needed to apply Wagner's Lemma.

Claim 4.6. *Consider Construction 4.2. It holds that*

$$\begin{aligned} & |\{i \mid \varphi_i \in \text{SAT}\}| \text{ is odd} \\ & \iff d_1 \text{ is contained in some minimal upward covering set } M \text{ for } A. \end{aligned} \quad (4.2)$$

Proof. To show (4.2) from left to right, suppose $|\{i \mid \varphi_i \in \text{SAT}\}|$ is odd. Recall that for each j , $1 < j \leq 2m$, if φ_j is satisfiable then so is φ_{j-1} . Thus, there exists some i , $1 \leq i \leq m$, such that $\varphi_1, \dots, \varphi_{2i-1} \in \text{SAT}$ and $\varphi_{2i}, \dots, \varphi_{2m} \notin \text{SAT}$. In Case 3 in the proof of Claim 4.4 we have seen that there is some minimal upward covering set for (B_i, \succ_i^B) —call it M_i —that corresponds to a satisfying assignment of φ_{2i-1} and that contains all alternatives of A_{2i} . Note that M_i contains d_{2i-1} . For

each $j \neq i$, $1 \leq j \leq m$, let M_j be some minimal upward covering set for (B_j, \succ_j^B) according to Case 1 (if $j < i$) and Case 2 (if $j > i$) in the proof of Claim 4.4.

In Case 1 in the proof of Claim 4.4 we have seen that d_{2i-3} is upward covered either by $\bar{u}'_{1,2i-3}$ or by $u'_{1,2i-3}$. This is no longer the case, since d_{2i-1} is in M_i and it dominates all alternatives in A_{2i-2} but not d_{2i-3} . By assumption, φ_{2i-3} is satisfiable, so there exists a minimal upward covering set, which contains d_{2i-3} as well. Thus, setting

$$M = \{d_1, d_3, \dots, d_{2i-1}\} \cup \bigcup_{1 \leq j \leq m} M_j,$$

it follows that M is a minimal upward covering set for (A, \succ) containing d_1 .

To show (4.2) from right to left, suppose that $|\{i \mid \varphi_i \in \text{SAT}\}|$ is even. For a contradiction, suppose that there exists some minimal upward covering set M for (A, \succ) that contains d_1 . If $\varphi_1 \notin \text{SAT}$ then we immediately obtain a contradiction by the argument in the proof of Theorem 4.2. On the other hand, if $\varphi_1 \in \text{SAT}$ then our assumption that $|\{i \mid \varphi_i \in \text{SAT}\}|$ is even implies that $\varphi_2 \in \text{SAT}$. It follows from the proof of Claim 4.1 that every minimal upward covering set for (A, \succ) (thus, in particular, M) contains either $\{u_{1,2i}, u'_{1,2i}\}$ or $\{\bar{u}_{1,2i}, \bar{u}'_{1,2i}\}$, but not both, and that none of the $e_{j,2i}$ and $e'_{j,2i}$ is in M . By the argument presented in Case 3 in the proof of Claim 4.4, the only way to prevent d_1 from being upward covered by an element of M , either $u'_{1,2}$ or $\bar{u}'_{1,2}$, is to include d_3 in M as well.² By applying the same argument $m - 1$ times, we will eventually reach a contradiction, since $d_{2m-1} \in M$ can no longer be prevented from being upward covered by an element of M , either $u'_{1,2m}$ or $\bar{u}'_{1,2m}$. Thus, no minimal upward covering set M for (A, \succ) contains d_1 , which completes the proof of (4.2). \square

Furthermore, it holds that $|\{i \mid \varphi_i \in \text{SAT}\}|$ is odd if and only if d_1 is contained in all minimum-size upward covering sets for A . This is true since the minimal upward covering sets for A that contain d_1 are those that correspond to some satisfying assignment for all satisfiable formulas φ_i , and as we have seen in the analysis of Construction 4.1 and the proof sketch of Theorem 4.2, these are the minimum-size upward covering sets for A .

4.3.2 Proofs for Minimal and Minimum-Size Upward Covering Sets

The results for problems related to minimal and minimum-size upward covering sets stated in Theorem 4.1 and Table 4.1 will be proven in this section by making use of the constructions presented in Section 4.3.1.

²This implies that d_1 is not upward covered by either $u'_{1,2}$ or $\bar{u}'_{1,2}$, since d_3 dominates them both but not d_1 .

Theorem 4.3. *It is NP-complete to decide, given a dominance graph (A, \succ) and a positive integer k , whether there exists a minimal/minimum-size upward covering set for A of size at most k . That is, both $\text{MC}_u\text{-SIZE}$ and $\text{MSC}_u\text{-SIZE}$ are NP-complete.*

Proof. This result can be proven by using the construction of Theorem 4.2. Let φ be a given boolean formula in conjunctive normal form, and let n be the number of variables occurring in φ . Setting the bound k for the size of a minimal/minimum-size upward covering set to $2n + 1$ proves that both problems are hard for NP. Indeed, as we have seen in the paragraph after the proof sketch of Theorem 4.2, there is a size $2n + 1$ minimal upward covering set (and hence a minimum-size upward covering set) for A if and only if φ is satisfiable. Both problems are NP-complete, since they can obviously be decided in nondeterministic polynomial time. \square

Theorem 4.4. *Deciding whether a designated alternative is contained in some minimal upward covering set for a given dominance graph is hard for Θ_2^p and in Σ_2^p . That is, $\text{MC}_u\text{-MEMBER}$ is hard for Θ_2^p and in Σ_2^p .*

Proof. Θ_2^p -hardness follows directly from Claim 4.6, which applies Wagner's Lemma to upward covering set problems. Specifically, this claim shows that in Construction 4.2 the alternative d_1 is contained in some minimal upward covering set for A if and only if the number of underlying boolean formulas that are satisfiable is odd. For the upper bound, let (A, \succ) be a dominance graph and d a designated alternative in A . First, observe that we can verify in polynomial time whether a subset of A is an upward covering set for A , simply by checking whether it satisfies internal and external stability. Now, we can guess an upward covering set $B \subseteq A$ with $d \in B$ in nondeterministic polynomial time and verify its minimality by checking that none of its subsets is an upward covering set for A . This places the problem in NP^{coNP} and consequently in Σ_2^p . \square

Theorem 4.5. *1. It is Θ_2^p -complete to decide whether a designated alternative is contained in some minimum-size upward covering set for a given dominance graph. That is $\text{MSC}_u\text{-MEMBER}$ is Θ_2^p -complete.*

2. It is Θ_2^p -complete to decide whether a designated alternative is contained in all minimum-size upward covering sets for a given dominance graph. That is $\text{MSC}_u\text{-MEMBER-ALL}$ is Θ_2^p -complete.

Proof. Wagner's Lemma can be used to show Θ_2^p -hardness for both problems. Claim 4.6 shows that in Construction 4.2 the alternative d_1 is contained in some minimal upward covering sets for A if and only if the number of underlying boolean

formulas that are satisfiable is odd. Hence d_1 is also contained in some minimum-size upward covering set if and only if the number of satisfiable boolean formulas is odd. And by the remark made after this claim it even holds that then the alternative d_1 is contained in all minimum-size upward covering sets for A . Hence $\text{MSC}_u\text{-MEMBER}$ and $\text{MSC}_u\text{-MEMBER-ALL}$ are both Θ_2^p -hard.

To see that $\text{MSC}_u\text{-MEMBER}$ is contained in Θ_2^p , let (A, \succ) be a dominance graph and d a designated alternative in A . Obviously, in nondeterministic polynomial time we can decide, given (A, \succ) , $x \in A$, and some positive integer $\ell \leq |A|$, whether there exists some upward covering set B for A such that $|B| \leq \ell$ and $x \in B$. Using this problem as an NP oracle, in Θ_2^p we can decide, given (A, \succ) and $d \in A$, whether there exists a minimum-size upward covering set for A containing d as follows. The oracle is asked whether for each pair (x, ℓ) , where $x \in A$ and $1 \leq \ell \leq |A|$, there exists an upward covering set for A of size bounded by ℓ that contains the alternative x . The number of queries is polynomial (more specifically in $\mathcal{O}(|A|^2)$), and all queries can be asked in parallel. Having all the answers, determine the size k of a minimum-size upward covering set for A , and accept if the oracle answer to (d, k) was yes, otherwise reject.

To show that $\text{MSC}_u\text{-MEMBER-ALL}$ is in Θ_2^p let (A, \succ) be a dominance graph and d a designated alternative in A . We now use as our oracle the set of all (x, ℓ) , where $x \in A$ is an alternative, and $\ell \leq |A|$ a positive integer, such that there exists some upward covering set B for A with $|B| \leq \ell$ and $x \notin B$. Clearly, this problem is also in NP, and the size k of a minimum-size upward covering set for A can again be determined by asking $\mathcal{O}(|A|^2)$ queries in parallel (if all oracle answers are no, it holds that $k = |A|$). Now, the Θ_2^p machine accepts its input $((A, \succ), d)$ if the oracle answer for the pair (d, k) is no, otherwise it rejects. \square

- Theorem 4.6.** *1. (Brandt and Fischer [BF08]) It is coNP-complete to decide whether a designated alternative is contained in all minimal upward covering sets for a given dominance graph. That is, $\text{MC}_u\text{-MEMBER-ALL}$ is coNP-complete.*
- 2. It is coNP-complete to decide whether a given subset of the alternatives is a minimal upward covering set for a given dominance graph. That is, $\text{MC}_u\text{-TEST}$ is coNP-complete.*
- 3. It is coNP-hard and in Σ_2^p to decide whether there is a unique minimal upward covering set for a given dominance graph. That is, $\text{MC}_u\text{-UNIQUE}$ is coNP-hard and in Σ_2^p .*

Proof. It follows from Claim 4.3 that in Construction 4.1 the boolean formula φ is not satisfiable if and only if the entire set of alternatives is a (unique) minimal upward covering set for A . Furthermore, if φ is satisfiable, there exists more than

one minimal upward covering set for A and none of them contains e_1 (provided that φ has more than one satisfying assignment, which can be ensured, if needed, by adding a dummy variable such that the satisfiability of the formula is not affected). This proves coNP-hardness for all three problems. $\text{MC}_u\text{-MEMBER-ALL}$ and $\text{MC}_u\text{-TEST}$ are also *contained* in coNP, as they can be decided in the positive by checking whether there exists an upward covering set that satisfies certain properties related to the problem at hand, so they both are coNP-complete. $\text{MC}_u\text{-UNIQUE}$ can be decided in the positive by checking whether there exists an upward covering set M such that all sets that are not strict supersets of M are *not* upward covering sets for the set of all alternatives. Thus, $\text{MC}_u\text{-UNIQUE}$ is in Σ_2^p . \square

The first statement of Theorem 4.6 was already shown by Brandt and Fischer [BF08]. However, their proof—which uses essentially the reduction from the proof of Theorem 4.2, except that they start from the coNP-complete problem Validity (which asks whether a given formula is valid, i.e., true under every assignment [Pap95])—does not yield any of the other coNP-hardness results in Theorem 4.6.

Theorem 4.7. *It is coNP-complete to decide whether a given subset of the alternatives is a minimum-size upward covering set for a given dominance graph. That is, $\text{MSC}_u\text{-TEST}$ is coNP-complete.*

Proof. This problem is in coNP, since it can be decided in the positive by checking whether the given subset M of alternatives is an upward covering set for the set A of all alternatives (which is easy) and all sets of smaller size than M are not upward covering sets for A (which is a coNP predicate). Now, coNP-hardness follows directly from Claim 4.3, which shows that in Construction 4.1 the boolean formula φ is not satisfiable if and only if there is a unique minimal upward covering set for A and hence also a unique minimum-size upward covering set for A . \square

Theorem 4.8. *Deciding whether there exists a unique minimum-size upward covering set for a given dominance graph is hard for coNP and in Θ_2^p . That is, $\text{MSC}_u\text{-UNIQUE}$ is coNP-hard and in Θ_2^p .*

Proof. It is easy to see that coNP-hardness follows directly from the coNP-hardness of $\text{MC}_u\text{-UNIQUE}$ (see Theorem 4.6). Membership in Θ_2^p can be proven by using the same oracle as in the proof of the first part of Theorem 4.5. We ask for all pairs (x, ℓ) , where $x \in A$ and $1 \leq \ell \leq |A|$, whether there is an upward covering set B for A such that $|B| \leq \ell$ and $x \in B$. Having all the answers, determine the minimum-size k of a minimum-size upward covering set for A . Accept if there are exactly k distinct alternatives x_1, \dots, x_k for which the answer for (x_i, k) , $1 \leq i \leq k$, was yes, otherwise reject. \square

An important consequence of the proofs of Theorem 4.6 and 4.8 (and of Construction 4.1 that underpins these proofs) regards the hardness of the search problems $\text{MC}_u\text{-FIND}$ and $\text{MSC}_u\text{-FIND}$.

Theorem 4.9. *Assuming $P \neq NP$, neither minimal upward covering sets nor minimum-size upward covering sets can be found in polynomial time. That is, neither $\text{MC}_u\text{-FIND}$ nor $\text{MSC}_u\text{-FIND}$ are polynomial-time computable unless $P = NP$.*

Proof. Consider the problem of deciding whether there exists a *nontrivial* minimal/minimum-size upward covering set, i.e., one that does *not* contain all alternatives. By Construction 4.1 that is applied in proving Theorems 4.6 and 4.8, there exists a trivial minimal/minimum-size upward covering set for A (i.e., one containing all alternatives in A) if and only if this set is the only minimal/minimum-size upward covering set for A . Thus, the coNP-hardness proof for the problem of deciding whether there is a unique minimal/minimum-size upward covering set for A (see the proofs of Theorems 4.6 and 4.8) immediately implies that the problem of deciding whether there is a nontrivial minimal/minimum-size upward covering set for A is NP-hard. However, since the latter problem can easily be reduced to the search problem (because the search problem, when used as a function oracle, yields the set of all alternatives if and only if this set is the only minimal/minimum-size upward covering set for A), it follows that the search problem cannot be solved in polynomial time unless $P = NP$. \square

4.4 Minimal and Minimum-Size Downward Covering Sets

4.4.1 Constructions for Minimal and Minimum-Size Downward Covering Sets

Turning now to the constructions used to show complexity results about minimal and minimum-size downward covering sets, we again start by giving a proof sketch of a result due to Brandt and Fischer [BF08], since the following constructions are based on their construction and proof.

Theorem 4.10. *Deciding whether a designated alternative is contained in some minimal downward covering set for a given dominance graph is NP-hard (i.e., $\text{MC}_d\text{-MEMBER}$ is NP-hard), even if a downward covering set is guaranteed to exist.*

Proof Sketch. NP-hardness of $\text{MC}_d\text{-MEMBER}$ is again shown by a reduction from SAT. Given a boolean formula in conjunctive normal form, $\varphi(v_1, v_2, \dots, v_n) =$

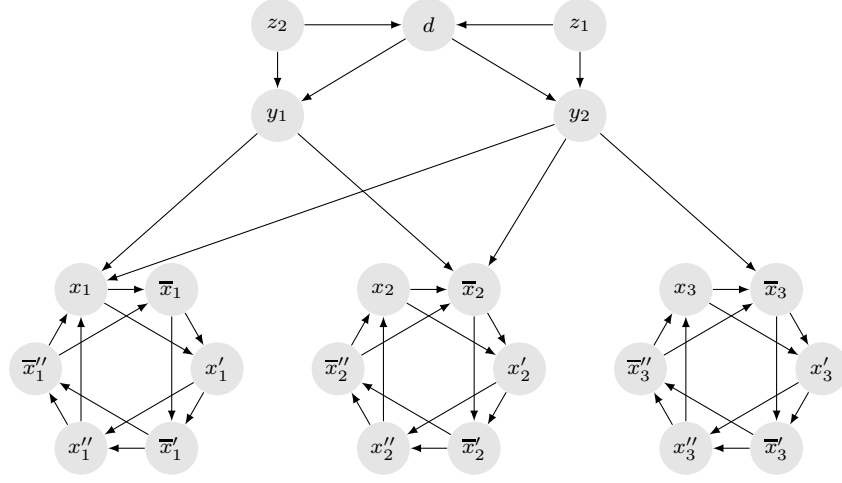


Figure 4.6: Dominance graph (A, \succ) for Theorem 4.10 resulting from the boolean formula $(v_1 \vee \neg v_2) \wedge (v_1 \vee \neg v_2 \vee \neg v_3)$.

$c_1 \wedge c_2 \wedge \dots \wedge c_r$, over the set $V = \{v_1, v_2, \dots, v_n\}$ of variables, we construct an instance of $\text{MC}_d\text{-MEMBER}$ with the dominance graph (A, \succ) and the distinguished alternative d whose membership in some minimal downward covering set for A is to be decided. The set of alternatives is

$$A = \{x_i, \bar{x}_i, x'_i, \bar{x}'_i, x''_i, \bar{x}''_i \mid v_i \in V\} \cup \{y_j, z_j \mid c_j \text{ is a clause in } \varphi\} \cup \{d\}$$

and the dominance relation \succ is defined by:

- For each i , $1 \leq i \leq n$, there is a cycle $x_i \succ \bar{x}_i \succ x'_i \succ \bar{x}'_i \succ x''_i \succ \bar{x}''_i \succ x_i$ with two nested three-cycles, $x_i \succ x'_i \succ x''_i \succ x_i$ and $\bar{x}_i \succ \bar{x}'_i \succ \bar{x}''_i \succ \bar{x}_i$;
- if variable v_i occurs in clause c_j as a positive literal, then $y_j \succ x_i$;
- if variable v_i occurs in clause c_j as a negative literal, then $y_j \succ \bar{x}_i$;
- for each j , $1 \leq j \leq r$, we have $d \succ y_j$ and $z_j \succ d$; and
- for each i and j with $1 \leq i, j \leq r$ and $i \neq j$, we have $z_i \succ y_j$.

Brandt and Fischer [BF08] showed that there is a minimal downward covering set containing d if and only if φ is satisfiable. An example of this reduction is shown in Figure 4.6 for the boolean formula $(v_1 \vee \neg v_2) \wedge (v_1 \vee \neg v_2 \vee \neg v_3)$. This formula can be satisfied by setting v_1, v_2 , and v_3 to false, the corresponding minimal downward covering set is $\{\bar{x}_1, \bar{x}'_1, \bar{x}''_1, \bar{x}_2, \bar{x}'_2, \bar{x}''_2, \bar{x}_3, \bar{x}'_3, \bar{x}''_3, y_1, y_2, z_1, z_2, d\}$. As desired this minimal downward covering set contains alternative d . \square

Regarding their construction sketched above, Brandt and Fischer [BF08] showed that every minimal downward covering set for A must contain exactly three alternatives for every variable v_i (either $x_i, x'_i,$ and x''_i , or $\bar{x}_i, \bar{x}'_i,$ and \bar{x}''_i), and the undominated alternatives z_1, \dots, z_r . Thus each minimal downward covering set for A consists of at least $3n + r$ alternatives and induces a truth assignment α for φ . The number of alternatives contained in any minimal downward covering set for A corresponding to an assignment α is $3n + r + k$, where k is the number of clauses that are satisfied if α is an assignment not satisfying φ , and where $k = r + 1$ if α is a satisfying assignment for φ . As a consequence, minimum-size downward covering sets for A correspond to those assignments for φ that satisfy the least possible number of clauses of φ . Note that this differs from the case of minimum-size upward covering sets for the dominance graph constructed in the proof sketch of Theorem 4.2. Hence the construction in the proof sketch of Theorem 4.10 cannot be used to obtain complexity results for minimum-size downward covering sets in the same way as the construction in the proof sketch of Theorem 4.2 was used to obtain complexity results for minimum-size upward covering sets.

As for upward covering set problems we will now provide a construction that can be used to show coNP-hardness for downward covering set problems, in addition this construction will also be used to show NP-hardness for downward covering set problems. Again a given boolean formula is transformed into a dominance graph and this construction will later be merged with the construction from the proof sketch of Theorem 4.10 so as to apply Lemma 4.1 to downward covering set problems.

Construction 4.3 (for NP- and coNP-hardness of downward covering set problems). *Given a boolean formula in conjunctive normal form, $\varphi(w_1, w_2, \dots, w_k) = f_1 \wedge f_2 \wedge \dots \wedge f_\ell$, over the set $W = \{w_1, w_2, \dots, w_k\}$ of variables, we construct a dominance graph (A, \succ) . The set of alternatives is*

$$A = A_1 \cup A_2 \cup \{\hat{a} \mid a \in A_1 \cup A_2\} \cup \{b, c, d\}$$

with $A_1 = \{x_i, x'_i, x''_i, \bar{x}_i, \bar{x}'_i, \bar{x}''_i, z_i, z'_i, z''_i \mid w_i \in W\}$ and $A_2 = \{y_j \mid f_j \text{ is a clause in } \varphi\}$, and the dominance relation \succ is defined by:

- For each $i, 1 \leq i \leq k$, there is, similarly to the construction in the proof of Theorem 4.10, a cycle $x_i \succ \bar{x}_i \succ x'_i \succ \bar{x}'_i \succ x''_i \succ \bar{x}''_i \succ x_i$ with two nested three-cycles, $x_i \succ x'_i \succ x''_i \succ x_i$ and $\bar{x}_i \succ \bar{x}'_i \succ \bar{x}''_i \succ \bar{x}_i$, and additionally we have $z'_i \succ z_i \succ x_i, z''_i \succ z_i \succ \bar{x}_i, z'_i \succ x_i, z''_i \succ \bar{x}_i$, and $d \succ z_i$;
- if variable w_i occurs in clause f_j as a positive literal, then $x_i \succ y_j$;
- if variable w_i occurs in clause f_j as a negative literal, then $\bar{x}_i \succ y_j$;
- for each $a \in A_1 \cup A_2$, we have $b \succ \hat{a}, a \succ \hat{a}$, and $\hat{a} \succ d$;

- for each j , $1 \leq j \leq \ell$, we have $d \succ y_j$; and
- $c \succ d$.

Figure 4.7 shows the resulting dominance graph (A, \succ) from the boolean formula $(w_1 \vee w_2 \vee w_3) \wedge (\neg w_2 \vee w_3)$, which can be satisfied by setting for example each of w_1 , w_2 , and w_3 to true. A minimal downward covering set for A corresponding to this assignment is $M = \{b, c, \} \cup \{x_i, x'_i, x''_i, z'_i, z''_i \mid 1 \leq i \leq 3\}$. Obviously, the undominated alternatives b , c , z'_i , and z''_i , $1 \leq i \leq 3$, are contained in every minimal downward covering set for the dominance graph constructed. The alternative d , however, is not contained in any minimal downward covering set for A . This can be seen as follows. If d were contained in some minimal downward covering set M' for A then none of the alternatives \hat{a} with $a \in A_1 \cup A_2$ would be downward covered. Hence, all alternatives in $A_1 \cup A_2$ would necessarily be in M' , since they all dominate a different alternative in M' . But then M' is no minimal downward covering set for A , since the minimal downward covering set M for A is a strict subset of M' .

Since we will use this construction to show NP- and coNP-hardness results for minimal and minimum-size downward covering sets we will now show some properties of Construction 4.3 in general. First we show that there always exists a minimal downward covering set for the constructed dominance graph.

Claim 4.7. *Minimal downward covering sets are guaranteed to exist for the dominance graph defined in Construction 4.3.*

Proof. The set A of all alternatives is a downward covering set for itself. Hence, there always exists a minimal downward covering set for the dominance graph defined in Construction 4.3. □

Now we show some key property concerning the alternative d in the constructed dominance graph.

Claim 4.8. *Consider the dominance graph (A, \succ) created by Construction 4.3. For each minimal downward covering set M for A , if M contains the alternative d then all other alternatives are contained in M as well (i.e., $A = M$).*

Proof. If d is contained in some minimal downward covering set M for A , then $\{a, \hat{a}\} \subseteq M$ for every $a \in A_1 \cup A_2$. To see this, observe that for an arbitrary $a \in A_1 \cup A_2$ there is no $a' \in A$ with $a' \succ \hat{a}$ and $a' \succ d$ or with $a' \succ a$ and $a' \succ \hat{a}$. Since the alternatives c and b are undominated, they are also in M , so $M = A$. □

The next two claims consider the minimal downward covering sets subject to the satisfiability of the underlying boolean formula.

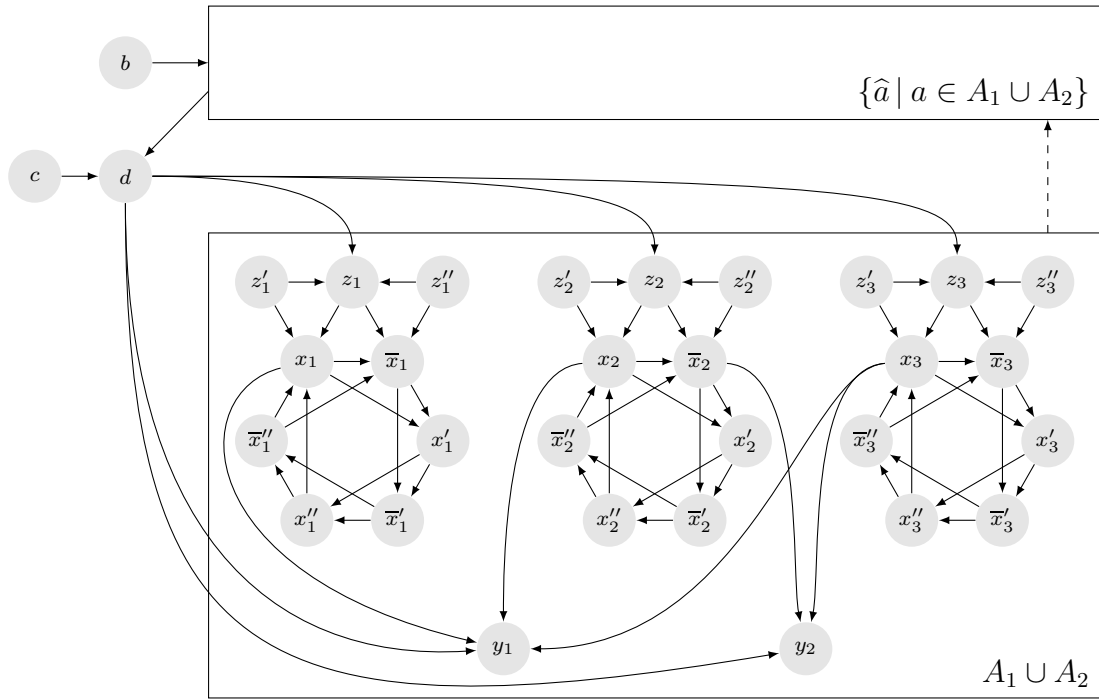


Figure 4.7: Dominance graph (A, \succ) resulting from the formula $(w_1 \vee w_2 \vee w_3) \wedge (\neg w_2 \vee w_3)$ according to Construction 4.3. An edge incident to a set of alternatives represents an edge incident to *each* alternative in the set. The dashed edge indicates that $a \succ \hat{a}$ for each $a \in A_1 \succ A_2$.

Claim 4.9. *Consider Construction 4.3. The boolean formula φ is satisfiable if and only if there is no minimal downward covering set for A that contains d .*

Proof. For the direction from left to right consider a satisfying assignment $\alpha : W \rightarrow \{0, 1\}$ for φ , and define the set

$$B_\alpha = \{b, c\} \cup \{x_i, x'_i, x''_i \mid \alpha(w_i) = 1\} \cup \{\bar{x}_i, \bar{x}'_i, \bar{x}''_i \mid \alpha(w_i) = 0\} \cup \{z'_i, z''_i \mid 1 \leq i \leq k\}.$$

It is not hard to verify that B_α is a minimal downward covering set for A . Thus, there exists a minimal downward covering set for A that does not contain d . If there were a minimal downward covering set M for A that contains d , Claim 4.8 would imply that $M = A$. However, since $B_\alpha \subset A = M$, this contradicts minimality, so no minimal downward covering set for A can contain d .

For the direction from right to left, assume that no minimal downward covering set for A contains d . Since by Claim 4.7 minimal downward covering sets are guaranteed to exist for the dominance graph defined in Construction 4.3, there exists a minimal downward covering set B for A that does not contain d , so $B \neq A$. It holds that $\{z_i \mid w_i \text{ is a variable in } \varphi\} \cap B = \emptyset$ and $\{y_j \mid f_j \text{ is a clause in } \varphi\} \cap B = \emptyset$, for otherwise a contradiction would follow by observing that there is no $a \in A$ with $a \succ d$ and $a \succ z_i$, $1 \leq i \leq k$, or with $a \succ d$ and $a \succ y_j$, $1 \leq j \leq \ell$. Furthermore, we have $x_i \notin B$ or $\bar{x}_i \notin B$, for each variable $w_i \in W$. By external stability, for each clause f_j there must exist an alternative $a \in B$ with $a \succ y_j$. By construction and since $d \notin B$, we must have either $a = x_i$ for some variable w_i that occurs in f_j as a positive literal, or $a = \bar{x}_i$ for some variable w_i that occurs in f_j as a negative literal. Now, define $\alpha : W \rightarrow \{0, 1\}$ such that $\alpha(w_i) = 1$ if $x_i \in B$, and $\alpha(w_i) = 0$ otherwise. It is readily appreciated that α is a satisfying assignment for φ . \square

Claim 4.10. *Consider Construction 4.3. The boolean formula φ is not satisfiable if and only if there is a unique minimal downward covering set for A .*

Proof. We again assume that if φ is satisfiable, it has at least two satisfying assignments. If φ is not satisfiable, there must be a minimal downward covering set for A that contains d by Claim 4.9, and by Claim 4.8 there must be a minimal downward covering set for A containing all alternatives. Hence, there is a unique minimal downward covering set for A . Conversely, if there is a unique minimal downward covering set for A , φ cannot be satisfiable, since otherwise there would be at least two distinct minimal downward covering sets for A , corresponding to the distinct truth assignments for φ , which would yield a contradiction. \square

In the dominance graph created by Construction 4.3, the minimal downward covering sets for A coincide with the minimum-size downward coverings sets for A . If φ is not satisfiable, there is only one minimal downward covering set for A , so this is also the only minimum-size downward covering set for A , and if φ is

satisfiable, the minimal downward covering sets for A correspond to the satisfying assignments of φ . As we have seen in the proof of Claim 4.9, these minimal downward covering sets for A always consist of $5k + 2$ alternatives. Thus, they each are also minimum-size downward covering sets for A .

Merging the construction from the proof sketch of Theorem 4.10 with Construction 4.3, we again provide a construction applying Lemma 4.1, this time to downward covering set problems.

Construction 4.4 (for applying Lemma 4.1 to downward covering set problems). *We again apply Wagner's Lemma with the NP-complete problem $S = \text{SAT}$ and construct a dominance graph. Fix an arbitrary $m \geq 1$ and let $\varphi_1, \varphi_2, \dots, \varphi_{2m}$ be $2m$ boolean formulas in conjunctive normal form such that the satisfiability of φ_j implies the satisfiability of φ_{j-1} , for each $j \in \{2, \dots, 2m\}$. Without loss of generality, we assume that for each j , $1 \leq j \leq 2m$, φ_j has at least two satisfying assignments, if φ_j is satisfiable.*

We now define a polynomial-time computable function f , which maps the given $2m$ boolean formulas to a dominance graph (A, \succ) that has useful properties for our downward covering set problems. The set of alternatives is

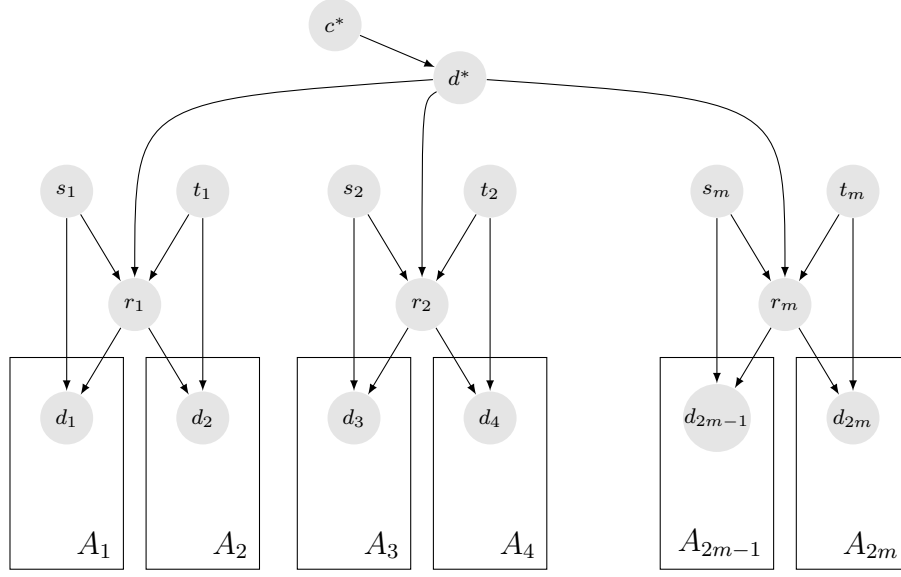
$$A = \left(\bigcup_{i=1}^{2m} A_i \right) \cup \left(\bigcup_{i=1}^m \{r_i, s_i, t_i\} \right) \cup \{c^*, d^*\},$$

and the dominance relation \succ on A is defined by

$$\left(\bigcup_{i=1}^{2m} \succ_i \right) \cup \left(\bigcup_{i=1}^m \{(r_i, d_{2i-1}), (r_i, d_{2i}), (s_i, r_i), (s_i, d_{2i-1}), (t_i, r_i), (t_i, d_{2i})\} \right) \cup \left(\bigcup_{i=1}^k \{(d^*, r_i)\} \right) \cup \{(c^*, d^*)\},$$

where we use the following notation:

- For each i , $1 \leq i \leq m$, let (A_{2i-1}, \succ_{2i-1}) be the dominance graph that results from the formula φ_{2i-1} according to Brandt and Fischer's construction given in the proof sketch of Theorem 4.10. We again use the same names for the alternatives in A_{2i-1} as in that proof sketch, except that we attach the subscript $2i - 1$.
- For each i , $1 \leq i \leq m$, let (A_{2i}, \succ_{2i}) be the dominance graph that results from the formula φ_{2i} according to Construction 4.3. We again use the same names for the alternatives in A_{2i} as in that construction, except that we attach the subscript $2i$.


 Figure 4.8: Dominance graph (A, \succ) from Construction 4.4

- For each i , $1 \leq i \leq m$, the dominance graphs (A_{2i-1}, \succ_{2i-1}) and (A_{2i}, \succ_{2i}) are connected by the alternatives s_i , t_i , and r_i (which play a similar role as the alternatives z_i , z'_i , and z''_i for each variable in Construction 4.3). The resulting dominance graph is denoted by (B_i, \succ_i^B) .
- Connect the m dominance graphs (B_i, \succ_i^B) , $1 \leq i \leq m$ (again similarly as in Construction 4.3). The alternative c^* dominates d^* , and d^* dominates the m alternatives r_i , $1 \leq i \leq m$.

This construction is illustrated in Figure 4.8. Clearly (A, \succ) is computable in polynomial time.

Claim 4.11. Consider Construction 4.4. For each i , $1 \leq i \leq 2m$, let M_i be a minimal downward covering set for (A_i, \succ_i) . Then each of the sets M_i must be contained in every minimal downward covering set for (A, \succ) .

Proof. For each i , $1 \leq i \leq 2m$, the only alternative in A_i dominated from outside A_i is d_i . Since d_i is also dominated by the undominated alternative $z_{1,i} \in A_i$ for odd i , and by the undominated alternative $c_i \in A_i$ for even i , it is readily appreciated that internal and external stability with respect to the elements of A_i only depend on the restriction of the dominance graph to A_i . \square

Claim 4.12. Consider Construction 4.4. It holds that

$$\begin{aligned} & |\{i \mid \varphi_i \in \text{SAT}\}| \text{ is odd} \\ & \iff d^* \text{ is contained in some minimal downward covering set } M \text{ for } A. \end{aligned} \quad (4.3)$$

Proof. For the direction from left to right in (4.3), assume that $|\{i \mid \varphi_i \in \text{SAT}\}|$ is odd. Thus, there is some $j \in \{1, \dots, m\}$ such that $\varphi_1, \varphi_2, \dots, \varphi_{2j-1}$ are each satisfiable and $\varphi_{2j}, \varphi_{2j+1}, \dots, \varphi_{2m}$ are each not. Define

$$M = \left(\bigcup_{i=1}^{2m} M_i \right) \cup \left(\bigcup_{i=1}^m \{s_i, t_i\} \right) \cup \{r_j, c^*, d^*\},$$

where for each i , $1 \leq i \leq 2m$, M_i is some minimal downward covering set of the restriction of the dominance graph to A_i , satisfying that $d_i \in M_i$ if and only if

1. i is odd and φ_i is satisfiable, or
2. i is even and φ_i is not satisfiable.

Such sets M_i exist by the proof sketch of Theorem 4.10 and by Claim 4.9. In particular, φ_{2j-1} is satisfiable and φ_{2j} is not, so $\{d_{2j-1}, d_{2j}\} \subseteq M$. There is no alternative that dominates d_{2j-1} , d_{2j} , and r_j . Thus, r_j must be in M . The other alternatives r_i , $1 \leq i \leq m$ and $i \neq j$, are downward covered by either s_i if $d_{2i-1} \notin M$, or t_i if $d_{2i} \notin M$. Finally, d^* cannot be downward covered, because $d^* \succ r_j$ and no alternative dominates both d^* and r_j . Internal and external stability with respect to the elements of M_i , as well as minimality of $\bigcup_{i=1}^{2k} M_i$, follow from the proofs of Theorem 4.10 and Claim 4.9. All other elements of M are undominated and thus contained in every downward covering set. We conclude that M is a minimal downward covering set for A that contains d^* .

For the direction from right to left in (4.3), assume that there exists a minimal downward covering set M for A with $d^* \in M$. By internal stability, there must exist some j , $1 \leq j \leq k$, such that $r_j \in M$. Thus, d_{2j-1} and d_{2j} must be in M , too. It then follows from the proof sketch of Theorem 4.10 and Claim 4.9 that φ_{2j-1} is satisfiable and φ_{2j} is not. Hence $|\{i \mid \varphi_i \in \text{SAT}\}|$ is odd. \square

By the remark made after Theorem 4.10, Construction 4.4 cannot be used straightforwardly to obtain complexity results for minimum-size downward covering sets.

4.4.2 Proofs for Minimal and Minimum-Size Downward Covering Sets

The results for problems related to minimal and minimum-size downward covering sets presented in Theorem 4.1 and Table 4.2 will be proved in this section by making use of the constructions presented in Section 4.4.1.

Theorem 4.11. *It is NP-complete to decide, given a dominance graph (A, \succ) and a positive integer k , whether there exists a minimal/minimum-size downward covering set for A of size at most k . That is, $\text{MC}_d\text{-SIZE}$ and $\text{MSC}_d\text{-SIZE}$ are both NP-complete.*

Proof. Membership in NP is obvious, since we can nondeterministically guess a subset $M \subseteq A$ of the alternatives with $|M| \leq k$ and can then check in polynomial time whether M is a downward covering set for A . NP-hardness of $\text{MC}_d\text{-SIZE}$ and $\text{MSC}_d\text{-SIZE}$ follows from Construction 4.3, the proof of Claim 4.9, and the comments made after Claim 4.10: If φ is a given formula with n variables, then there exists a minimal/minimum-size downward covering set of size $5n + 2$ if and only if φ is satisfiable. \square

Theorem 4.12. $\text{MSC}_d\text{-MEMBER}$, $\text{MSC}_d\text{-MEMBER-ALL}$, and $\text{MSC}_d\text{-UNIQUE}$ are coNP-hard and in Θ_2^p .

Proof. It follows from Claim 4.10 that in Construction 4.3 the boolean formula φ is not satisfiable if and only if the entire set A of alternatives is the unique minimum-size downward covering set for itself. Moreover, assuming that φ has at least two satisfying assignments, if φ is satisfiable, there are at least two distinct minimum-size downward covering sets for A . This shows that each of $\text{MSC}_d\text{-MEMBER}$, $\text{MSC}_d\text{-MEMBER-ALL}$, and $\text{MSC}_d\text{-UNIQUE}$ is coNP-hard. For all three problems, membership in Θ_2^p is shown similarly to the proofs of the corresponding minimum-size upward covering set problems. However, since downward covering sets may fail to exist, the proofs must be slightly adapted. For $\text{MSC}_d\text{-MEMBER}$ and $\text{MSC}_d\text{-UNIQUE}$, the machine rejects the input if the size k of a minimum-size downward covering set cannot be computed (simply because there doesn't exist any such set). For $\text{MSC}_d\text{-MEMBER-ALL}$, if all oracle answers are no, it must be checked whether the set of all alternatives is a downward covering set for itself. If so, the machine accepts the input, otherwise it rejects. \square

Theorem 4.13. *It is coNP-complete to decide whether a given subset is a minimum-size downward covering set for a given dominance graph. That is, $\text{MSC}_d\text{-TEST}$ is coNP-complete.*

Proof. This problem is in coNP, since its complement (i.e., the problem of deciding whether a given subset of the set A of alternatives is not a minimum-size downward covering set for A) can be decided in nondeterministic polynomial time. Hardness for coNP follows directly from Claim 4.10, which shows that in Construction 4.3 the boolean formula φ is not satisfiable if and only if there is a unique minimal downward covering set for A and hence also a unique minimum-size downward covering set for A . \square

Theorem 4.14. *Deciding whether a designated alternative is contained in some minimal downward covering set for a given dominance graph is hard for Θ_2^p and in Σ_2^p . That is, $\text{MC}_d\text{-MEMBER}$ is hard for Θ_2^p and in Σ_2^p .*

Proof. Membership in Σ_2^p can be shown analogously to the proof of Theorem 4.4, and Θ_2^p -hardness follows directly from Claim 4.12, which applies Wagner’s Lemma to downward covering sets. Specifically, this claim shows that in Construction 4.4 the alternative d^* is contained in some minimal downward covering set for A if and only if the number of underlying boolean formulas is odd. \square

Theorem 4.15. 1. (Brandt and Fischer [BF08]) *It is coNP-complete to decide whether a designated alternative is contained in all minimal downward covering sets for a given dominance graph. That is, $\text{MC}_d\text{-MEMBER-ALL}$ is coNP-complete.*

2. *It is coNP-complete to decide whether a given subset of the alternatives is a minimal downward covering set for a given dominance graph. That is, $\text{MC}_d\text{-TEST}$ is coNP-complete.*

3. *It is coNP-hard and in Σ_2^p to decide whether there is a unique minimal downward covering set for a given dominance graph. That is, $\text{MC}_d\text{-UNIQUE}$ is coNP-hard and in Σ_2^p .*

Proof. It follows from Claim 4.10 that φ is not satisfiable if and only if the entire set of alternatives A is a unique minimal downward covering set for A . Furthermore, if φ is satisfiable, there exists more than one minimal downward covering set for A and none of them contains d (provided that φ has more than one satisfying assignment, which can be ensured, if needed, by adding a dummy variable such that the satisfiability of the formula is not affected). This proves coNP-hardness for all three problems. $\text{MC}_d\text{-MEMBER-ALL}$ and $\text{MC}_d\text{-TEST}$ are also contained in coNP, because they can be decided in the positive by checking whether there does not exist a downward covering set that satisfies certain properties related to the problem at hand. Thus, they are both coNP-complete. $\text{MC}_d\text{-UNIQUE}$ can be decided in the positive by checking whether there exists a downward covering set M such that all sets that are not strict supersets of M are *not* downward covering sets for the set of all alternatives. This shows that $\text{MC}_d\text{-UNIQUE}$ is in Σ_2^p . \square

The first statement of Theorem 4.15 was already shown by Brandt and Fischer [BF08]. However their proof—which uses essentially the reduction from the proof of Theorem 4.10, except that they start from the coNP-complete problem VALIDITY —does not yield any of the other coNP-hardness results in Theorem 4.15.

An important consequence of the proofs of Theorem 4.12 and 4.15 regards the hardness of the search problems $\text{MC}_d\text{-FIND}$ and $\text{MSC}_d\text{-FIND}$. (Note that the hardness of $\text{MC}_d\text{-FIND}$ also follows from a result by Brandt and Fischer [BF08, Thm. 9], see the discussion in Section 4.2).

Theorem 4.16. *Assuming $P \neq NP$, neither minimal downward covering sets, nor minimum-size downward covering sets can be found in polynomial time (i.e., neither MC_d -FIND nor MSC_d -FIND are polynomial time computable unless $P = NP$), even when the existence of a downward covering set is guaranteed.*

Proof. Consider the problem of deciding whether there exists a *nontrivial* minimal/minimum-size downward covering set, i.e., one that does *not* contain all alternatives. By Construction 4.3 that is applied in proving Theorems 4.12 and 4.15, there exists a trivial minimal/minimum-size downward covering set for A (i.e., one containing all alternatives in A) if and only if this set is the only minimal/minimum-size downward covering set for A . Thus, the coNP-hardness proof for the problem of deciding whether there is a unique minimal/minimum-size downward covering set for A (see the proofs of Theorems 4.12 and 4.15) immediately implies that the problem of deciding whether there is a nontrivial minimal/minimum-size downward covering set for A is NP-hard. However, since the latter problem can easily be reduced to the search problem (because the search problem, when used as a function oracle, yields the set of all alternatives if and only if this set is the only minimal/minimum-size downward covering set for A), it follows that the search problem cannot be solved in polynomial time unless $P = NP$. \square

4.5 Summary

We raised the existing NP-lower bounds for the problems MC_u -MEMBER and MC_d -MEMBER to Θ_2^P , and provided a Σ_2^P upper bound. We defined and studied the complexity of various problems related to minimal upward and downward covering sets. In addition we studied all these problems also in their minimum-size variants. We presented one construction for upward covering set problems and one for downward covering set problems, and then showed how to merge them with the constructions given by Brandt and Fischer [BF08] to apply Wagner's Lemma [Wag87]. By using these constructions we showed hardness for NP, coNP, and Θ_2^P . An important consequence of our results is that neither minimal/minimum-size upward nor minimal/minimum-size downward covering sets can be found in polynomial time unless $P = NP$.

5 Interference in Judgment Aggregation

The aim of a judgment aggregation process is to aggregate individual judgment sets of judges over possibly interconnected propositions to reach a collective outcome. The complexity-theoretic study of problems associated to judgment aggregation was initiated by Endriss et al. [EGP10a, EGP10b], by the analysis of the winner determination and manipulation problem in judgment aggregation. In this chapter we pursue their direction and study the complexity of manipulation for premise-based quota rules, a whole class of judgment aggregation procedures. A judgment aggregation scenario is said to be manipulable if one judge has an incentive to report an untruthful judgment set as this yields a more favorable outcome for him, where the distance between the preferences of the manipulator can be measured by the Hamming distance. We show not only NP-hardness but also study the parameterized complexity (W[2]-hardness) of this problem. Furthermore we show strategyproofness for certain restrictions on the agenda. Besides the manipulation problem, we also investigate bribery in judgment aggregation. This work is inspired by different bribery problems in voting theory. In addition to classical complexity (NP-hardness) results we also obtain W[2]-hardness for bribery with respect to natural parameters, and membership in P for restricted problem instances for bribery problems. Furthermore, we introduce three different types of control specific to judgment aggregation, again inspired by the corresponding problems from voting theory. NP-hardness is shown for all three types of control considered here. The results presented in this chapter have already been published in [BER11, BEER12b, BEER12a].

5.1 Preliminaries

Different ways of influencing the outcome of elections have been studied in social choice. The complexity of problems related to manipulation, bribery, and control has also been studied intensely in computational social choice, see, e.g., the early work of Bartholdi et al. [BTT89, BO91, BTT92] and the recent surveys and book chapters by Faliszewski et al. [FP10, FHH10], Brandt et al. [BCE12], and Baumeister et al. [BEH⁺10]. These problems are not only relevant to voting, but to decision-making processes in general. Studying the susceptibility to different ways

of influencing the outcome is particularly important for judgment aggregation, since the aggregation of different *yes/no* opinions about possibly interconnected propositions is often applied in practice. To avoid these forms of interference, a common approach in computational social choice is to apply methods from theoretical computer science to show that undesirable strategic behavior is blocked, or at least hindered, by the corresponding task being a computationally intractable problem. In this chapter we will introduce various manipulation, bribery, and control problems for judgment aggregation and study their computational complexity.

We adopt the formal definition of the judgment aggregation framework from Endriss et al. [EGP10b]. Let PS be the set of all propositional variables and \mathcal{L}_{PS} the set of propositional formulas built from PS . As connectives we allow disjunction (\vee), conjunction (\wedge), and equivalence (\leftrightarrow) in their usual meaning, as shown in Table 5.1, where the boolean constants 1 and 0 represent “true” and “false”, respectively.

Table 5.1: Overview of connectives in propositional formulas

a	b	$a \vee b$	$a \wedge b$	$a \leftrightarrow b$
0	0	0	0	1
0	1	1	0	0
1	0	1	0	0
1	1	1	1	1

For notational convenience, we want to avoid double negations and let $\sim\alpha$ denote the complement of α . This means that $\sim\alpha = \neg\alpha$ if α is not negated, and $\sim\alpha = \beta$ if $\alpha = \neg\beta$. The set of formulas to be judged by the judges is called the *agenda* and will be denoted by Φ . Formally, the agenda is a nonempty subset of \mathcal{L}_{PS} , does not contain doubly negated formulas, and is closed under complementation. Hence it holds that $\sim\alpha \in \Phi$ for all $\alpha \in \Phi$. The judgment provided by a single judge is called his or her *individual judgment set* and corresponds to the propositions in the agenda accepted by this judge. The *collective judgment set* is the set of propositions obtained by aggregating the individual judgment sets with some judgment aggregation rule. Formally, an individual or collective judgment set J on an agenda Φ is a subset $J \subseteq \Phi$. For both, individual and collective judgment sets we consider the three basic properties: completeness, complement-freeness, and consistency; that are defined as follows.

- A judgment set J is *complete* if it contains α or $\sim\alpha$ for each $\alpha \in \Phi$.
- A judgment set J is *complement-free* if there is no $\alpha \in J$ with $\sim\alpha \in J$.
- A judgment set J is *consistent* if there is an assignment that satisfies all formulas in J .

Note that the consistency of a judgment set directly implies its complement-freeness.

The set of judges that take part in a judgment aggregation scenario will be denoted by $N = \{1, \dots, n\}$, and we will also assume that $n \geq 2$, hence there are at least 2 judges. By J_i , $i \in N$, we denote the individual judgment set of judge i , and the profile of all n individual judgment sets is denoted be $\mathbf{J} = (J_1, \dots, J_n)$.

A judgment aggregation procedure F is needed to obtain a collective judgment set from a given profile $\mathbf{J} \in \mathcal{J}(\Phi)^n$. Formally, this is a function $F : \mathcal{J}(\Phi)^n \rightarrow 2^\Phi$, mapping a profile of n complete and consistent judgment sets to a subset of the agenda Φ , the collective judgment set. We consider the same three basic properties for judgment aggregation procedures as for judgment sets. A judgment aggregation procedure F is said to be *complete/complement-free/consistent* if $F(\mathbf{J})$ is complete/complement-free/consistent for all profiles $\mathbf{J} \in \mathcal{J}(\Phi)^n$. A very natural judgment aggregation procedure is the majority rule, where a proposition is contained in the collective outcome, if it is contained in a majority of the individual outcomes. A common approach to avoid an inconsistent collective outcome is to apply the judgment aggregation rule only to the premises of the agenda, see also the illustration of the doctrinal paradox in Section 2.3. Such a procedure is formalized by the *premise-based procedure*, see Endriss et al. [EGP10b].

Definition 5.1 (Premise-based Procedure [EGP10b]). *Let the agenda Φ be divided into two disjoint sets, $\Phi = \Phi_p \uplus \Phi_c$, where Φ_p is the set of premises and Φ_c is the set of conclusions, and both Φ_p and Φ_c are closed under complementation. The premise-based procedure is a function $PBP : \mathcal{J}(\Phi)^n \rightarrow 2^\Phi$ mapping, for $\Phi = \Phi_p \uplus \Phi_c$, each profile $\mathbf{J} = (J_1, \dots, J_n)$ to the following judgment set:*

$$PBP(\mathbf{J}) = \Delta \cup \{\varphi \in \Phi_c \mid \Delta \models \varphi\}$$

with $\Delta = \{\varphi \in \Phi_p \mid |\{i \mid \varphi \in J_i\}| > n/2\}$, where $|S|$ denotes the cardinality of set S and \models denotes the satisfaction relation.

This definition applies the majority procedure only to the premises of the agenda, and the collective outcome for the conclusions is derived from the collective outcome of the premises. However, this is not sufficient to always obtain a complete and consistent procedure. Assume that the agenda contains the variable α and $\sim\alpha$, the propositional formula $\alpha \vee \beta$ and its negation. Applying the premise-based procedure to this agenda, identifying the set of literals as the premises, results in an incomplete outcome if the collective judgment set contains $\sim\alpha$, since the collective outcome for the formula $\alpha \vee \beta$ cannot be derived. To obtain a complete and consistent procedure it is furthermore required that the agenda is closed under propositional variables (i.e., every variable that occurs in a formula of Φ is contained in Φ), that the set of premises is the set of all literals in the agenda, and that the number of judges is odd. Endriss et al. [EGP10b] argue that this definition

is appropriate, since the problem of determining whether an agenda guarantees a complete and consistent outcome for the majority procedure is an intractable problem.

We extend this approach to the class of *uniform quota rules* as defined by Dietrich and List [DL07b]. In contrast to the definition of the premise-based procedure we allow an arbitrary quota and do not restrict our scenarios to an odd number of judges.

Definition 5.2 (Premise-based Quota Rule). *Let the agenda Φ be divided into two disjoint sets, $\Phi = \Phi_p \uplus \Phi_c$, where Φ_p is the set of premises and Φ_c is the set of conclusions, and both Φ_p and Φ_c are closed under complementation. Divide the set of premises Φ_p into two disjoint subsets, Φ_1 and Φ_2 , such that for each $\varphi \in \Phi_p$, either $\varphi \in \Phi_1$ and $\sim\varphi \in \Phi_2$ or $\varphi \in \Phi_2$ and $\sim\varphi \in \Phi_1$. Define a quota $q_\varphi \in \mathbb{Q}$ with $0 \leq q_\varphi < 1$ for every $\varphi \in \Phi_1$. The quota for every $\varphi \in \Phi_2$ is then defined as $q'_\varphi = 1 - q_\varphi$. The premise-based quota rule is a function $PQR : \mathcal{J}(\Phi)^n \rightarrow 2^\Phi$ mapping for $\Phi = \Phi_p \uplus \Phi_c$, each profile $\mathbf{J} = (J_1, \dots, J_n)$ to the following judgment set:*

$$PQR(\mathbf{J}) = \Delta_q \cup \{\varphi \in \Phi_c \mid \Delta_q \models \varphi\},$$

where

$$\Delta_q = \{\varphi \in \Phi_1 \mid |\{i \mid \varphi \in J_i\}| > n \cdot q_\varphi\} \cup \{\varphi \in \Phi_2 \mid |\{i \mid \varphi \in J_i\}| > \lceil n \cdot q'_\varphi - 1 \rceil\}.$$

We again require that the agenda Φ is closed under propositional variables and that Φ_p consists of all literals, to obtain complete and consistent collective judgment sets. Note that in the case of $q_\varphi = q'_\varphi = 1/2$ and an even number of judges the number of affirmations needed to be in the collective judgment set differs for the variables in Φ_1 and Φ_2 . For $\varphi \in \Phi_1$, at least $\lfloor n \cdot q_\varphi + 1 \rfloor$ affirmations from the judges are needed, and for $\varphi \in \Phi_2$, $\lceil n \cdot q'_\varphi \rceil$ affirmations are needed. Clearly, since $\lfloor n \cdot q_\varphi + 1 \rfloor + \lceil n \cdot q'_\varphi \rceil = n + 1$, it is ensured that for every $\varphi \in \Phi$, either $\varphi \in PQR(\mathbf{J})$ or $\sim\varphi \in PQR(\mathbf{J})$. Observe that the quota $q_\varphi = 1$ for a literal $\varphi \in \Phi_1$ is not considered here, since then $n + 1$ affirmations were needed for $\varphi \in \Phi_1$ to be in the collective judgment set, which is not possible. Hence, the outcome does not depend on the individual judgment sets. By contrast, considering $q_\varphi = 0$ leads to the case that $\varphi \in \Phi_1$ needs at least one affirmation, and $\sim\varphi \in \Phi_2$ needs n affirmations, which may be a reasonable choice.

A special case of the premise-based quota rules are *uniform premise-based quota rules*. Here the quota q_φ is identical for all literals in Φ_1 and hence also the quota q'_φ for all literals in Φ_2 . We denote the quotas by q for all $\varphi \in \Phi_1$ and q' for all $\varphi \in \Phi_2$. In this chapter we will focus on these uniform premise-based quota rules and denote it by $UPQR_q$. For the case of $q = 1/2$ and an odd number of judges, we obtain exactly the premise-based procedure defined by Endriss et al. [EGP10b] (see Definition 5.1).

Furthermore, we will consider yet another variant of premise-based procedure, which was introduced by Dietrich and List [DL07b] and is called *constant premise-based quota rule*. Formally, it is defined by

$$CPQR(\mathbf{J}) = \Delta'_q \cup \{\varphi \in \Phi_c \mid \Delta'_q \models \varphi\}.$$

In contrast to the premise-based quota rule, here the number of affirmations needed to be in the set Δ'_q does not depend on the number of judges, but is a fixed constant. Thus $q_\varphi \in \mathbb{N}$, $0 \leq q_\varphi < n$, and $\Delta'_q = \{\varphi \in \Phi_1 \mid |\{i \mid \varphi \in J_i\}| > q_\varphi\} \cup \{\varphi \in \Phi_2 \mid |\{i \mid \varphi \in J_i\}| > q'_\varphi\}$. To ensure that for every $\varphi \in \Phi$, either $\varphi \in CPQR(\mathbf{J})$ or $\sim \varphi \in CPQR(\mathbf{J})$, we again require that $q_\varphi + q'_\varphi = n - 1$ for all $\varphi \in \Phi_1$. The uniform variant, $UCPQR_q$ is defined analogously.

Obviously both classes represent the same judgment aggregation procedures if the number of judges taking part in the process is fixed. However, we will study control problems where the number of judges can vary. In this case, the quota n in the constant premise-based quota rule can be seen as an upper bound on the highest number of judges possibly participating in the process. This definition is closely related to (a simplified version of) a referendum. Suppose that there is a fixed number of possible participants who are allowed to go to the polls (e.g., all citizens of a town), and there is a fixed number of affirmations needed for a certain decision, independent of the number of people who are actually participating. Of course, this number may depend on the number of possible participants, for example 20% of them.

The problems of manipulation, bribery, and control were first defined and studied for preference aggregation, especially for voting scenarios. Now we argue, that it makes sense to investigate these problems also in the context of judgment aggregation. Recall the example from Section 2.3, where the premise-based procedure for the majority rule is applied to aggregate the individual judgments from three referees for a penalty decision. Bovens and Rabinowicz [BR06] (see also List [Lis06]) provide a similar example where a committee has to decide whether a candidate deserves tenure on the basis of his teaching and research capabilities. To get tenure, the candidate has to be good enough in research and he has to be good enough in teaching. Table 5.2 illustrates the individual judgments of three judges and the collective judgment set obtained by applying the premise-based procedure for the majority rule.

On the basis of such examples, List [Lis06] concludes that in a premise-based procedure the judges might have an incentive to report insincere judgments. In the above example the tenure decision derived by the majority rule used as a premise-based procedure is “yes”. Suppose that the judges are absolutely sure about their decision and hence want the aggregated outcome of the conclusion to be identical to their own conclusions. In this case, judge 2 has an incentive to insincerely change his judgment for teaching from “yes” to “no”. Then the collective outcome is “no”

Table 5.2: Example illustrating the premise-based procedure for the majority rule [BR06, Lis06]

	Teaching	Research		Tenure
Judge 1	yes	yes		yes
Judge 2	yes	no		no
Judge 3	no	yes		no
Majority	yes	yes	\Rightarrow	yes

for teaching and hence tenure is denied for the candidate, as desired by judge 2. For the same reason judge 3 has an incentive to report “no” instead of “yes” when judging the candidates research capabilities. This is a classical *manipulation scenario* which we have already described for voting in Chapter 3.1. In the context of voting manipulation has been studied in depth, see, e.g., [Con10, FHH10, FP10] and the references cited therein. In the field of judgment aggregation, strategic judging (i.e., changing one’s individual judgments for the purpose of manipulating the collective outcome) was previously considered by List [Lis06] and by Dietrich and List [DL07c]. Recently the computational aspect of strategic judging has been studied by Endriss et al. [EGP10b] for the first time.

In addition to the above described strategic judging we will also study bribery in judgment aggregation. Along with the possible winner problems in Chapter 3 we already introduced some variants of bribery in the context of voting, see Faliszewski et al. [FHH09, FHHR09a] and Elkind et al. [EFS09]. For bribery in judgment aggregation return to the above example. Suppose that the judgments of judge 2 and judge 3 were “no” for both premises. The candidate wants to get tenure by any means necessary and might try to make some deals with some of the judges in order to reach a positive evaluation. For example he might offer to take off some of the teaching load of judge 2’s shoulder, or offer to apply for a joint research grant with judge 3, or simply bribe the judges with money not exceeding his budget. Beside the context of voting, bribery has also been studied in the context of optimal lobbying (first by Christian et al. [CFRS07], see also [EFG⁺09] and Chapter 5.2 for more details).

The last type of interference we consider is control for judgment aggregation. For voting systems control by adding, deleting, or partitioning candidates or voters has been studied extensively (see, e.g., [BTT92, HHR07, BEH⁺10, BEH⁺10]). We will study three different types of control for judgment aggregation, namely adding, deleting, and replacing judges. The problems where judges are added or deleted correspond to the problems of adding and deleting voters in voting, but control by replacing judges is newly introduced here. This is especially important in the context of judgment aggregation, since in the case of prejudice it is usual

to replace a judge. Detailed motivation for all three types of control by examples from the American jury trial system and international arbitration can be found in [BEER12b].

5.2 Problem Definitions

In voting theory bribery problems were introduced by Faliszewski et al. [FHH09] (see also, e.g., [EFS09, FHHR09a]). Here an external actor tries to bribe the voters, without exceeding his budget, such that a desired candidate becomes a winner of the election. In judgment aggregation it is not the case that there is a single winner, so the briber may bribe the judges to obtain a specific collective outcome, or he might be interested in only some formulas of the collective outcome. Hence the briber has a, maybe incomplete, judgment set as desired outcome. This exact bribery problem is then defined as follows for a given aggregation procedure F .

F -EXACT BRIBERY

- Given:** An agenda Φ , a profile $\mathbf{T} \in \mathcal{J}(\Phi)^n$, a consistent judgment set J (not necessarily complete) desired by the briber, and a positive integer k .
- Question:** Is it possible to change up to k individual judgment sets in \mathbf{T} such that for the resulting new profile \mathbf{T}' it holds that $J \subseteq F(\mathbf{T}')$?
-

Note that in the case of a complete desired judgment set J the question is whether $J = F(\mathbf{T}')$.

The manipulation problem in voting asks, if a voter can make his desired candidate win the election by reporting an untruthful preference (see [BTT89, BO91]). Since in the case of judgment aggregation there is no winner, Endriss et al. [EGP10b] used the Hamming distance to measure the distance between two judgments sets in the definition of the manipulation problem in judgment aggregation. In their definition, an outcome (i.e., a collective judgment set) is more desirable for the manipulator if its Hamming distance to the manipulator's desired judgment is smaller, where for an agenda Φ the Hamming distance $H(J, J')$ between two complete and consistent judgment sets $J, J' \in \mathcal{J}(\Phi)$ is defined as the number of positive formulas in Φ on which J and J' differ. The manipulation problem in judgment aggregation is then defined as follows for a given aggregation procedure F .

F -MANIPULATION

- Given:** An agenda Φ , a profile $\mathbf{T} \in \mathcal{J}(\Phi)^{n-1}$, and a consistent and complete judgment set J desired by the manipulator.
- Question:** Does there exist a judgment set $J' \in \mathcal{J}(\Phi)$ such that $H(J, F(\mathbf{T}, J')) < H(J, F(\mathbf{T}, J))$?
-

If for a given judgment aggregation procedure a judge can never benefit from reporting an insincere individual judgment set, the procedure will be called strategy-proof.

We will adopt the Hamming distance approach, and also define bribery problems where not an exact outcome is sought, but an outcome with a small Hamming distance to the desired judgment set. However we extend the approach also to incomplete (albeit consistent) desired judgment sets. This reflects the scenario where the briber may be interested only in some part of the agenda. The definition of the Hamming distance is extended accordingly as follows. Let Φ be an agenda, $J \in \mathcal{J}(\Phi)$ be a complete and consistent judgment set, and $J' \subseteq \Phi$ be a consistent judgment set. The *Hamming distance* $H(J, J')$ between J and J' is defined as the number of formulas from J' on which J does not agree:

$$H(J, J') = |\{\varphi \mid \varphi \in J' \wedge \varphi \notin J\}|.$$

Note that if J' is also complete, this extended notion of Hamming distance coincides with the notion Endriss et al [EGP10b] use.

Now we can give the formal definition of bribery in judgment aggregation, where the briber seeks to obtain a collective judgment set having a smaller Hamming distance to the desired judgment set, then the original outcome has.

<i>F</i> -BRIBERY	
Given:	An agenda Φ , a profile $\mathbf{T} \in \mathcal{J}(\Phi)^n$, a consistent judgment set J (not necessarily complete) desired by the briber, and a positive integer k .
Question:	Is it possible to change up to k individual judgment sets in \mathbf{T} such that for the resulting new profile \mathbf{T}' it holds that $H(F(\mathbf{T}'), J) < H(F(\mathbf{T}), J)$?

The notion of microbribery for irrational voters was introduced by Faliszewski et al. [FHHR09a]. Instead of paying to change a whole vote, here the briber has to pay separately for each change in a preference table. We transfer the notion of microbribery to judgment aggregation, by allowing a briber with budget k to change up to k premise entries in the given profile instead of k entire judgment sets. Note that since we focus on premise-based procedures, the entries for the conclusions in the individual judgment sets are changed automatically according to the premises if necessary. The formal definition of microbribery for a premise-based judgment aggregation procedure F is as follows.

F-MICROBRIBERY

- Given:** An agenda Φ , a profile $\mathbf{T} \in \mathcal{J}(\Phi)^n$, a consistent judgment set J (not necessarily complete) desired by the briber, and a positive integer k .
- Question:** Is it possible to change up to k entries among the premises in the individual judgment sets in \mathbf{T} such that for the resulting new profile \mathbf{T}' it holds that $H(F(\mathbf{T}'), J) < H(F(\mathbf{T}), J)$?
-

We also consider the problem *F*-EXACT MICROBRIBERY which is defined analogously to the corresponding bribery problem with the difference that the briber is allowed to change only up to k premise-entries in \mathbf{T} rather than to change k complete individual judgment sets.

Now we turn to the formal definitions of the control problems. For a given judgment aggregation procedure F , the problem of control by adding judges is defined as follows.

F-CONTROL BY ADDING JUDGES

- Given:** An agenda Φ , profiles $\mathbf{T} \in \mathcal{J}(\Phi)^n$ and $\mathbf{S} \in \mathcal{J}(\Phi)^{|\mathbf{S}|}$, a positive integer k , and a consistent judgment set J (not necessarily complete).
- Question:** Is there a subset $\mathbf{S}' \subset \mathbf{S}$, $|\mathbf{S}'| \leq k$, such that $H(J, F(\mathbf{T} \cup \mathbf{S}')) < H(J, F(\mathbf{T}))$?
-

If we consider the variant *F*-EXACT CONTROL BY ADDING JUDGES, we ask if there is a subset $\mathbf{S}' \subset \mathbf{S}$, $|\mathbf{S}'| \leq k$, such that $J \subseteq F(\mathbf{T} \cup \mathbf{S}')$.

Control by deleting judges is defined as follows for a given judgment aggregation procedure F .

F-CONTROL BY DELETING JUDGES

- Given:** An agenda Φ , a profile $\mathbf{T} \in \mathcal{J}(\Phi)^n$, a positive integer k , and a consistent judgment set J (not necessarily complete).
- Question:** Is there a subset $\mathbf{T}' \subset \mathbf{T}$ with $|\mathbf{T}'| \leq k$ such that $H(J, F(\mathbf{T} \setminus \mathbf{T}')) < H(J, F(\mathbf{T}))$?
-

The exact variant *F*-EXACT CONTROL BY DELETING JUDGES is defined analogously to the case of adding judges.

Control by replacing judges is the new control problem we introduce here and it considers the case where some judges may be replaced. For a specific judgment aggregation procedure F it is defined as follows.

F-CONTROL BY REPLACING JUDGES

- Given:** An agenda Φ , profiles $\mathbf{T} \in \mathcal{J}(\Phi)^n$ and $\mathbf{S} \in \mathcal{J}(\Phi)^{|\mathbf{S}|}$, a positive integer k , and a consistent judgment set J (not necessarily complete).
- Question:** Are there subsets $\mathbf{T}' \subset \mathbf{T}$ and $\mathbf{S}' \subset \mathbf{S}$, with $|\mathbf{T}'| = |\mathbf{S}'| \leq k$, such that $H(J, F((\mathbf{T} \setminus \mathbf{T}') \cup \mathbf{S}')) < H(J, F(\mathbf{T}))$?
-

Define F -EXACT CONTROL BY REPLACING JUDGES analogously to the exact variants of the adding and deleting judges problems.

To study the complexity of adding, deleting, and replacing judges, we adopt the terminology introduced in [BTT92] for control problems in voting and adapt it to judgment aggregation.

Let F be an aggregation procedure and let \mathcal{C} be a given control type.

1. F is said to be *immune* to control by \mathcal{C} if it is never possible for an external person to successfully control the judgment aggregation procedure via \mathcal{C} -control.
2. F is said to be *susceptible* to control by \mathcal{C} if it is not immune.
3. F is said to be *resistant* to control by \mathcal{C} if it is susceptible and the corresponding decision problem is NP-hard.
4. F is said to be *vulnerable* to control by \mathcal{C} if it is susceptible and the corresponding decision problem is in P.

In one of our proofs we will use a reduction from the OPTIMAL LOBBYING problem which is closely related to judgment aggregation:

OPTIMAL LOBBYING	
Given:	An $m \times n$ matrix L (whose rows represent the voters, whose columns represent the referenda, and whose 0-1 entries represent No/Yes votes), a positive integer $k \leq m$, and a target vector $x \in \{0, 1\}^n$.
Question:	Is there a choice of k rows in L such that by changing the entries of these rows the resulting matrix has the property that, for each j , $1 \leq j \leq n$, the j th column has a strict majority of ones (respectively, zeros) if and only if the j th entry of the target vector x of The Lobby is one (respectively, zero)?

This problem has been introduced by Christian et al. [CFRS07]. They showed W[2]-completeness when parameterized by the number k of rows The Lobby can change. A more general framework of the lobbying problem and more W[2]-hardness results can be found in [EFG⁺09].

The multiple referenda as in OPTIMAL LOBBYING can be seen as a special case of judgment aggregation, where the agenda is closed under complementation and propositional variables and contains only premises and where the majority rule is used for aggregation. The potential of The Lobby corresponds to our exact bribery problem. More precisely, a PBP -EXACT BRIBERY instance with only premises in

the agenda and with a complete desired judgment set J is nothing other than an OPTIMAL LOBBYING instance, where J corresponds to The Lobby’s target vector.¹

Such multiple referenda are another example why it is worth studying bribery and manipulation in judgment aggregation. Suppose the citizens of a town have to decide by a referendum whether two projects, A and B (e.g., a new street and a new gymnasium), are to be realized. The citizens are asked to give their opinion only for the premises A and B and not for the conclusion (whether both projects are to be realized), hence the doctrinal paradox is avoided. Now consider the building contractor who, of course, is interested in being awarded a contract for both projects. He might try to bribe some of the citizens to influence the outcome of the referenda. And again the citizens might also vote strategically in these referenda, since both projects will cost money, and it is clear that if both projects are realized, the amount available for each must be reduced. So some citizens may wish to support A , but they are not satisfied if the amount for A would be reduced when both projects are realized. Thus it is natural that they consider the possibility of reporting insincere votes (provided they know how the others will vote); this may turn out to be more advantageous for them, as then they possibly can prevent that both projects are realized.

5.3 Results

5.3.1 Manipulation in Judgment Aggregation

Endriss et al. [EGP10b] showed that *PBP*-MANIPULATION is NP-complete, we will extend the study of the manipulation problem to uniform premise-based quota rules and establish $W[2]$ -hardness results with respect to a natural parameter. The proof will be by a reduction from a classical problem in graph theory called DOMINATING SET. In a given graph $G = (V, E)$, with the set of vertices V and the set of edges E , a *dominating set* is a subset $V' \subseteq V$ of the vertices such that for each $v \in V \setminus V'$ there is an edge $\{v, v'\}$ in E with $v' \in V'$. The size of a dominating set V' is the number $|V'|$ of its vertices. The formal definition of the unparameterized NP-complete (see [GJ79]) version is as follows.

DOMINATING SET	
Given:	A graph $G = (V, E)$, with the set V of vertices and the set E of edges, and a positive integer $k \leq V $.
Question:	Does G have a dominating set of size at most k ?

¹Although exact bribery in judgment aggregation thus generalizes lobbying in the sense of Christian et al. [CFRS07] (which is different from bribery in voting, as defined by Faliszewski et al. [FHH09]), we will use the term “bribery” rather than “lobbying” in the context of judgment aggregation.

This problem is $W[2]$ -complete when parameterized by the upper bound k on the size of the dominating set (see [DF99]). To be explicit we denote this parameterized version by k -DOMINATING SET.

Theorem 5.1. *For each rational quota q , $0 \leq q < 1$ and for any fixed number $n \geq 3$ of judges, $UPQR_q$ -MANIPULATION is $W[2]$ -hard when parameterized by the maximum number of changes in the premises needed in the manipulator's judgment set.*

Proof. We start by giving the details for $q = 1/2$ and three judges, and later explain how this proof can be extended to capture any other rational quota values q , $0 \leq q < 1$, and any fixed number of judges greater than three.

The proof will be by a reduction from the $W[2]$ -complete problem k -DOMINATING SET. Given a graph $G = (V, E)$ with the set of vertices $V = \{v_1, \dots, v_n\}$, define $N(v_i)$ as the closed neighborhood of vertex v_i , i.e., the union of the set of vertices adjacent to v_i and the vertex v_i itself. Then, V' is a dominating set for G if and only if $N(v_i) \cap V' \neq \emptyset$ for each i , $1 \leq i \leq n$. We will now describe how to construct a manipulation instance for judgment aggregation. Let the agenda Φ contain the variables² v_1, \dots, v_n, y and their negations, the formula $\varphi_i = v_i^1 \vee \dots \vee v_i^j \vee y$ and its negation, where $\{v_i^1, \dots, v_i^j\} = N(v_i)$ for each i , $1 \leq i \leq n$, and $n - 1$ syntactic variations of each of these formulas and its negation. This can be seen as giving each formula φ_i a weight of n . A syntactic variation of a formula can, for example, be obtained by an additional conjunction with the constant 1. Furthermore, Φ contains the formula $v_1 \vee \dots \vee v_n$, its negation, and $n^2 - k - 2$ syntactic variations of this formula and its negation; this can be seen as giving this formula a weight of $n^2 - k - 1$. The set of judges is $N = \{1, 2, 3\}$, with the individual judgment sets J_1 , J_2 , and J_3 (where J_3 is the judgment set of the manipulative judge), and the collective judgment set as shown in Table 5.3. Note that the Hamming distance between J_3 and the collective judgment set is $1 + n^2$.

Table 5.3: Construction for the proof of Theorem 5.1

Judgment Set	v_1	\dots	v_n	y	φ_1	\dots	φ_n	$v_1 \vee \dots \vee v_n$
J_1	1	\dots	1	0	1	\dots	1	1
J_2	0	\dots	0	0	0	\dots	0	0
J_3	0	\dots	0	1	1	\dots	1	0
$UPQR_{1/2}$	0	\dots	0	0	\Rightarrow 0	\dots	0	0

We claim that there is an alternative judgment set for J_3 that yields a smaller Hamming distance to the collective outcome if and only if there is a dominating

²We use the same identifiers v_1, \dots, v_n for the vertices of G and the variables in Φ , specifying the intended meaning only if it is not clear from the context.

set of size at most k for G .

(\Leftarrow) Assume that there is a dominating set V' of G with $|V'| = k$. (If $|V'| < k$, we simply add any $k - |V'|$ vertices to obtain a dominating set of size exactly k .) Regarding the premises, the judgment set of the manipulator contains the variables $v_i \in V'$ and also the literal y . Then the collective outcome also contains the variables $v_i \in V'$, and since V' is a dominating set, each φ_i , $1 \leq i \leq n$, evaluates to true and the formula $v_1 \vee \dots \vee v_n$ is also evaluated to true. The Hamming distance to the original judgment set of the manipulator is then $k + 1 + (n^2 - k - 1) = n^2$. Hence the manipulation was successful, and the number of entries changed in the judgment set of the manipulator is exactly k .

(\Rightarrow) Now assume that there is a successful manipulation with judgment set J' . The manipulator can change only the premises in the agenda to achieve a better outcome for him or her. A change for the literal y changes nothing in the collective outcome, hence the changes must be within the set $\{v_1, \dots, v_n\}$. Including j of the v_i into J' has the effect that these v_i are included in the collective judgment set, and that all variations of the formula $v_1 \vee \dots \vee v_n$ and of those φ_i that are evaluated to true are also included in the collective judgment set. If ℓ formulas φ_i are evaluated to true in the collective judgment set, the Hamming distance to J_3 is $j + 1 + (n^2 - n\ell) + (n^2 - k - 1)$. Since the manipulation was successful, the Hamming distance can be at most n^2 . If $\ell < n$, it must hold that $j \leq k - n$, which is not possible given that $k \leq n$ and $j > 0$. Hence, $\ell = n$ and $j = k$. Then exactly k literals v_i are set to true, and since this satisfies all φ_i , they must correspond to a dominating set of size k , concluding the proof for the quota $q = 1/2$ and three judges.

This proof can be adapted to work for any fixed number $m \geq 3$ of judgment sets S_1, \dots, S_m and for any rational value of q , with $1 \leq mq < m$. The agenda remains the same, but $S_1, \dots, S_{\lfloor mq \rfloor}$ are each equal to the judgment set J_1 and $S_{\lfloor mq \rfloor + 1}, \dots, S_{m-1}$ are each equal to the judgment set J_2 . The judgment set S_m of the manipulative judge equals the judgment set J_3 , and the quota is q for every positive variable and $1 - q$ for every negative variable. The number of affirmations every positive formula needs to be in the collective judgment set is then $\lfloor mq \rfloor + 1$. Then the same argumentation as above holds.

For the remaining case, where $0 \leq mq < 1$, the construction must be slightly modified. The formulas $\varphi_1, \dots, \varphi_n$ are replaced by $\varphi'_i = (v_i^1 \wedge \dots \wedge v_i^j) \vee \neg y$, where $\{v_i^1, \dots, v_i^j\} = N(v_i)$ for each i , $1 \leq i \leq n$, and the individual judgment sets J_1, \dots, J_m are shown as in Table 5.4, where J_m is the judgment set of the manipulative judge. Then by similar arguments as above there is a successful manipulation if and only if the given graph has a dominating set of size at most k .

Since the number of premises changed by the manipulator depends only on the size k of the dominating set, W[2]-hardness for $UPQR_q$ -MANIPULATION holds for this parameter. \square

Table 5.4: Construction for the second part of the proof of Theorem 5.1

Judgment Set	v_1	\cdots	v_n	y	φ'_1	\cdots	φ'_n	$v_1 \vee \cdots \vee v_n$
J_1, \dots, J_{m-1}	0	\cdots	0	1	0	\cdots	0	0
J_m	0	\cdots	0	0	1	\cdots	1	0
$UPQR_{1/2}$	0	\cdots	0	1	\Rightarrow	0	\cdots	0

Since the unparameterized problem DOMINATING SET is NP-complete, the proof of Theorem 5.1 implies NP-hardness of $UPQR_q$ -MANIPULATION for any fixed number $n \geq 3$ of judges. Note that a generalization of the proof of Theorem 2 in [EGP10b], which shows NP-hardness for $UPQR_{1/2}$ -MANIPULATION by a reduction from the boolean satisfiability problem, also shows the NP-hardness of $UPQR_q$ -MANIPULATION, but this reduction would not be appropriate to establish W[2]-hardness, since the corresponding parameterized version of SAT is not known to be W[2]-hard.

Studying a fixed number of judges and the parameter maximum number of changes in the premises needed in the manipulator's judgment set, is very natural. We already argued that a fixed total number of judges is important to study since the number of participating judges may be small in many situations. Considering the parameter maximum number of changes in the premises needed in the manipulator's judgment set is also very natural, since the manipulator may wish to report a judgment set as close as possible to his or her sincere judgment set in order to remain undiscovered.

In contrast to the hardness results established in Theorem 5.1, the following proposition shows that under certain assumptions $UPQR_q$, $0 \leq q < 1$, is strategy-proof.

Proposition 5.1. *If the agenda contains only premises then $UPQR_q$, $0 \leq q < 1$, is strategy-proof.*

Proof. Assume that the agenda Φ contains only premises. Then every variable is considered independently. Let n be the number of judges. If φ is contained in the judgment set J of the manipulator, and φ does not have $\lfloor n \cdot q + 1 \rfloor$ (respectively, $\lceil n(1-q) \rceil$) affirmations without considering J , it cannot reach the required number of affirmations if the manipulator switches from φ to $\neg\varphi$ in his judgment set. \square

Dietrich and List [DL07c] showed the far more general result, that any independent and monotonic Judgment aggregation procedures is strategy-proof. But since we use a slightly different formal framework, we gave the short proof for Proposition 5.1 instead of translating their result into our framework.

Since $UPQR_q$ -MANIPULATION is NP-complete with a fixed number of judges, there is little hope to find a polynomial-time algorithm for the general problem,

even when the number of judges participating is fixed. However, if the agenda is simple and contains no conclusions, by Proposition 5.1, $UPQR_q$ is strategy-proof.

5.3.2 Bribery in Judgment Aggregation

After studying the complexity of manipulation in judgment aggregation for the class of uniform premise-based quota rules, we now turn to bribery in judgment aggregation. In this section we will study the complexity of different bribery problems in judgment aggregation for the premise-based procedure PBP , which is equal to $UPQR_{1/2}$ for an odd number of judges. We will again establish NP-completeness, W[2]-hardness for a natural parameter, and show polynomial-time solvability under certain assumptions. The first problem we consider is PBP -BRIBERY

Theorem 5.2. *PBP -BRIBERY is NP-complete, even when the total number of judges ($n \geq 3$ odd) or the number of judges that can be bribed is a fixed constant.*

Proof. We will show NP-hardness by a slightly modified construction from the proof of Theorem 5.1 (see Table 5.3). Membership in NP is obvious. We start by considering the case, where the briber is allowed to bribe exactly one judge. The notation and the agenda from that proof remain unchanged, but the individual judgment sets are slightly different. The first two judges remain unchanged, but the third judge has the same judgment set as the second one, and the desired judgment set J is equal to J_3 from the proof of Theorem 5.1. Table 5.5 summarizes these individual judgment sets and the evaluation according to PBP .

Table 5.5: Construction for the proof of Theorem 5.2

Judgment Set	v_1	\dots	v_n	y		φ_1	\dots	φ_n	$v_1 \vee \dots \vee v_n$
J_1	1	\dots	1	0		1	\dots	1	1
J_2, J_3	0	\dots	0	0		0	\dots	0	0
PBP	0	\dots	0	0	\Rightarrow	0	\dots	0	0
J	0	\dots	0	1		1	\dots	1	0

Since the quota is $1/2$, two affirmations are needed to be in the collective judgment set. Again the briber cannot benefit from bribing one judge to switch from $\neg y$ to y in his or her individual judgment set. Hence the change must be in the set of variables $\{v_1, \dots, v_n\}$ from the second or the third judge. By a similar argument as in the proof of Theorem 5.1, there is a successful bribery action if and only if there is a dominating set of size at most k for the given graph.

Now we consider the case that the briber is allowed to bribe more than one judge. If the briber is allowed to bribe c judges, we construct an instance with $2c+1$ judges,

where one judgment set is equal to J_1 and the remaining $2c$ individual judgment sets are equal to J_2 from the proof of Theorem 5.1. It is again not possible for the briber to change the entry for y , and the briber must change the entry for any v_i in the judgment sets from c judges to obtain a different collective outcome. This construction works by similar arguments as above. \square

Now we turn to the case where the bribery can change up to a fixed number of entries in the premises of the individual judgment sets, instead of completely changing a fixed number of individual judgment sets. We consider the cases where the number of judges or the number of microbribes allowed is a fixed constant, where a microbribe denotes the change of one premise entry in an individual judgment set.

Theorem 5.3. *PBP-MICROBRIBERY is NP-complete, even when the total number of judges ($n \geq 3$) or the number of microbribes allowed is a fixed constant.*

Proof. The proof that PBP-MICROBRIBERY is NP-hard is similar to the proof of Theorem 5.2. The agenda Φ is defined as in the proof of Theorem 5.1. Let $c \in \mathbb{N}$ be a fixed constant. The number of judges is $2c + 1$, where the individual judgment sets of c judges are of type J_1 and the remaining $c + 1$ individual judgment sets are of type J_2 . The desired outcome of the briber is the judgment set J_3 from the proof of Theorem 5.1. The number of affirmations needed to be in the collective judgment set is at least $c + 1$, and the number of entries the briber is allowed to change is at most k . Since none of the judges have y in their individual judgment sets, the briber cannot change the collective outcome for y to 1. Hence all entries that can be changed are for the variables v_1, \dots, v_n . Obviously, setting the value for one v_i in one of the judges of type J_2 to 1 causes v_i to be in the collective judgment set and all other changes have no effect on the collective judgment set. By similar arguments as in the proof of Theorem 5.1, there is a successful microbribery action if and only if the given graph has a dominating set of size at most k . Since membership in NP is obvious this completes the proof. \square

In the case of PBP-EXACT BRIBERY we now show W[2]-hardness with respect to the number of judges that can be bribed by making use of the W[2]-hardness of an OPTIMAL LOBBYING problem.

Theorem 5.4. *PBP-EXACT BRIBERY is W[2]-hard when parameterized by the number of judges that can be bribed.*

Proof. Observe that an exact bribery instance with only premises in the agenda and with a complete desired judgment set J is exactly the OPTIMAL LOBBYING problem. Since this is W[2]-complete for the parameter number of rows that can be changed, PBP-EXACT BRIBERY inherits the W[2]-hardness lower bound, where the parameter is the number of judges that can be bribed. \square

Note that $W[2]$ -hardness with respect to any parameter directly implies NP-hardness for the corresponding unparameterized problem, so PBP -EXACT BRIBERY is also NP-complete (all unparameterized problems considered here are easily seen to be in NP).

For PBP -EXACT MICROBRIBERY we again show NP-completeness.

Theorem 5.5. *PBP -EXACT MICROBRIBERY is NP-complete, even when the total number of judges ($n \geq 3$ odd) or the number of microbribes allowed is a fixed constant.*

Proof. Consider the construction in the proof of Theorem 5.3, and change the agenda such that there are only $n^2 - 2$ (instead of $n^2 - k - 2$) syntactic variations of the formula $v_1 \vee \dots \vee v_n$ (i.e., this can be seen as giving a weight of $n^2 - 1$ to this formula), and the desired judgment set J is incomplete and contains all conclusions. Note that this is possible due to the additional variable y . By similar arguments as above, a successful microbribery of k entries is possible if and only if there is a dominating set for G of size at most k . \square

In Theorems 5.2, 5.3, and 5.5 we studied different bribery problems with a fixed number of judges, as for the manipulation problem in Theorem 5.1. It turned out that even in this case BRIBERY, MICROBRIBERY, and EXACT-MICROBRIBERY, are all NP-complete for the premise-based procedure for the majority rule. Furthermore we considered the case of a fixed number of judges allowed to bribe for PBP -BRIBERY, the corresponding parameter for its exact variant, and the case of a fixed number of microbribes allowed for PBP -MICROBRIBERY and its exact variant. Both parameters concern the budget of the briber. Since the briber aims at spending as little money as possible, it is also natural to consider this parameter. But again, NP-completeness was shown even when the budget is a fixed constant and in one case $W[2]$ -hardness for this parameter, so bounding the budget does not help to solve the problem easily.

Although the exact microbribery problem is computationally hard in general for the aggregation procedure PBP , there are some interesting naturally restricted instances where it is computationally easy.

Theorem 5.6. *If the desired judgment set is complete or if the desired judgment set is incomplete but contains all of the premises or only premises, then PBP -EXACT MICROBRIBERY is in P.*

Proof. We give only an informal description of the algorithm that computes a successful microbribery

Input: Our algorithm takes as input a complete profile \mathbf{T} , a consistent judgment set J , and a positive integer k .

Step 1: For each premise present in J , compute the minimum number of entries that have to be flipped in order to make the collective judgment on that premise equal to the desired judgment set's entry on that premise. Note that this can be done in linear time, since it is a simple counting. Let d_i denote the number of entries needed to flip for premise i .

Step 2: Check if $\sum_i d_i \leq k$.

Output: If $\sum_i d_i \leq k$, output the entries which have to be flipped and halt. Otherwise, output “bribery impossible” and halt.

Clearly this algorithm works in polynomial time. The output is correct since if we need no more than k flips in the premises, the premises are evaluated exactly as they are in J , and the conclusions follow automatically, since we are using a premise-based procedure. \square

5.3.3 Control in Judgment Aggregation

For the above studied manipulation and bribery problems the number of judges participating is constant and hence uniform premise-based quota rules and uniform constant premise-based quota rules describe the same judgment aggregation procedures. However, in control by adding or deleting judges the number of participating judges is not fixed. The number of affirmations varies with the number of participating judges for the uniform premise-based quota rule, whereas the number of affirmations needed in the uniform constant premise-based quota rule remains the same regardless of the number of judges participating. Since in control by adding or deleting judges the number of judges participating varies, we study these problems with respect to both judgment aggregation procedures. In contrast, the number of judges participating is constant in control by replacing judges. We will first consider the uniform premise-based quota rule and show hardness for $UCPQR_q$ for control by adding judges in the Hamming distance based and in the exact variant. The following proofs will again be by a reduction from the NP-complete problem DOMINATING SET.

Theorem 5.7. *For each admissible value of q , $UCPQR_q$ is resistant to CONTROL BY ADDING JUDGES and to EXACT CONTROL BY ADDING JUDGES.*

Proof. Membership in NP is obvious for both problems. We prove resistance to EXACT CONTROL BY ADDING JUDGES and to CONTROL BY ADDING JUDGES at the same time by indicating the slight differences required in the construction.

The reduction is similar to that in the proof of Theorem 5.1. Let (G, k) be a given DOMINATING SET instance and let $N(v_i)$ denote the closed neighborhood of vertex $v_i \in V$. For the judgment aggregation instance, let the agenda Φ contain the variables v_1, \dots, v_n, y and their negations, for each i , $1 \leq i \leq n$, the formula

$\varphi_i = v_i^1 \vee \dots \vee v_i^m \vee y$, where $\{v_i^1, \dots, v_i^m\} = N(v_i)$, and its negation, the formula $\psi = v_1 \vee \dots \vee v_n$, its negation, and $n - 2$ syntactic variations of this formula and its negation. This can be seen as giving the weight $n - 1$ to the formula ψ . The quota for every positive literal is q , hence $q + 1$ affirmations are needed to be in the collective judgment set. The set \mathbf{T} of judges who initially take part contains q judgment sets that contain the literals v_1, \dots, v_n , and hence all formulas φ_i , $1 \leq i \leq n$, and the formula ψ and its syntactic variations, and the negation of all formulas in Φ not mentioned here. Furthermore, there is one judgment set \mathbf{T} that contains all negated formulas from Φ .

In the case of EXACT CONTROL BY ADDING JUDGES the desired judgment set J is incomplete and contains the formulas φ_i , $1 \leq i \leq n$. And in the case of CONTROL BY ADDING JUDGES the desired judgment set J additionally contains the negation of the formula ψ (and its syntactic variations). Observe that in both cases J is consistent, since setting y to true and all v_i , $1 \leq i \leq n$, to false results in the desired evaluation. The profile \mathbf{S} of judges who may be added contains n judges, with the individual judgment sets J_i , $1 \leq i \leq n$, where J_i contains the variable v_i , the negation of all v_j , $1 \leq j \leq n$, $j \neq i$, the negation of y , and the corresponding conclusions.

For the case of EXACT CONTROL BY ADDING JUDGES we claim that there is a dominating set of size at most k for G if and only if we can ensure that the outcome contains all formulas from J by adding at most k judges from \mathbf{S} . From left to right, if there is a dominating set V' , we can ensure that the formulas from J are part of the collective judgment set by adding those judges J_i with $v_i \in V'$. Thus, all formulas φ_i , $1 \leq i \leq n$, evaluate to true.

Conversely, assume that all formulas φ_i , $1 \leq i \leq n$, evaluate to true. It is not possible to achieve this by having y in the collective outcome, since there are no individual judgment sets containing y . Hence, the collective outcome for v_i , $1 \leq i \leq n$, makes all φ_i true. The maximum number of judges that can be added is k , and exactly one literal v_i is contained in the collective judgment set for each judge from \mathbf{S} that is added. Hence, the vertices v_i corresponding to the judges J_i from \mathbf{S} that have been added must form a dominating set for the graph G .

Now we consider the case of CONTROL BY ADDING JUDGES. Note that the Hamming distance between the collective outcome from the judges in \mathbf{T} and J is n . Assume that there is a dominating set V' , as in the previous case add those judges J_i with $v_i \in V'$. Then, the Hamming distance of the collective outcome to J is $n - 1$, since the formula ψ (and its $n - 2$ syntactic variations) are evaluated to true, and J contains the negations of these $n - 1$ formulas.

If there are at least k judges from \mathbf{S} that are added such that the Hamming distance of the collective outcome to J is smaller than n , all formulas φ_i , $1 \leq i \leq n$ must be evaluated to true, since adding judges from \mathbf{S} always causes that the formula ψ (and its $n - 2$ syntactic variations) are contained in the collective outcome. Hence, it must again be the case that those vertices v_i where the judge

J_i from \mathbf{S} was added must form a dominating set for the graph G . \square

Now we show NP-hardness for control by deleting judges in the Hamming distance and in the exact variant.

Theorem 5.8. *For each admissible value of q , $UCPQR_q$ is resistant to CONTROL BY DELETING JUDGES and to EXACT CONTROL BY DELETING JUDGES.*

Proof. Both problems are easily seen to be in NP, and we will show NP-hardness by a similar construction as in the proof of Theorem 5.7. Again, we show hardness in both cases with the same construction. We first consider the case of EXACT CONTROL BY DELETING JUDGES. For a given DOMINATING SET instance (G, k) , we construct the following judgment aggregation scenario. Let the agenda Φ be the same as in the proof of Theorem 5.7 plus an additional variable z , its negation, and $n - 1$ syntactic variations of this variable and its negation. This can again be seen as if z has a weight of n . The quota is q for $\neg v_i$, $1 \leq i \leq n$, and all remaining positive literals. The profile \mathbf{T} contains q individual judgment sets that each contain z and the negation of all remaining formulas in Φ , one judgment set containing v_1, \dots, v_n, z and the negation of all remaining formulas, and for each i one judgment set J_i that contains all v_i , $i \neq j$, and the negation of all remaining formulas. The judgment set desired by the chair is incomplete and contains z (and its syntactic variations) and the formulas φ_i , $1 \leq i \leq n$.

We claim that there is a dominating set of size at most k for G if and only if there is a successful control action. If there is a dominating set V' for G , then the desired judgment set J is obtained by deleting those J_i with $v_i \in V'$. Conversely, assume that it is possible that the desired formulas are in the collective judgment set by deleting at most k judges. Since z is in the collective outcome, only judges of the form J_i , $1 \leq i \leq n$, may be deleted. The deletion of a judge J_i has the effect that v_i is in the collective outcome, hence at most k different v_i are contained in the collective outcome, and since they evaluate all formulas φ_i , $1 \leq i \leq n$, to true, those v_i must form a dominating set of size at most k for G .

For the problem CONTROL BY DELETING JUDGES the desired outcome J also contains the formula ψ . To see that the reduction holds, observe that the Hamming distance from J to the collective judgment set of the initial instance is n . If there is a dominating set V' for G , deleting those judges J_i with $v_i \in V'$, results in an outcome with Hamming distance $n - 1$ to J , since ψ is contained in the collective outcome.

Now, assume that by deleting k judges an outcome with a smaller Hamming distance to J than between the original outcome and J is obtained. Removing z from the collective judgment set always results in a Hamming distance to J that is at least n , hence z is still in the collective judgment set. This implies that only judges of the form J_i have been deleted. Deleting one judge J_i in turn causes ψ to be in the collective judgment set. Hence the Hamming distance to J is $n - 1$, and

all formulas φ_i , $1 \leq i \leq n$, must also be contained in the collective outcome. Thus those v_i where the judge J_i has been deleted form a dominating set of size at most k for G . \square

In the following we consider the uniform premise-based quota rule in the case of adding and deleting judges. Here we focus on $UPQR_{1/2}$, which equals the premise-based procedure PBP in the case of an odd number of judges. We start by showing NP-hardness for control by adding judges in both problem variants. We will show NP-hardness by a reduction from the NP-complete problem EXACT COVER BY 3-SETS (X3C for short), which is formally defined in Section 3.5. Recall that this problem asks, if for a given set $X = \{x_1, \dots, x_{3m}\}$ and a collection $C = \{C_1, \dots, C_n\}$ with $|C_i| = 3$ and $C_i \subseteq X$, $1 \leq i \leq n$, there is a subcollection $C' \subseteq C$ that is an exact cover for X .

Theorem 5.9. *$UPQR_{1/2}$ is resistant to EXACT CONTROL BY ADDING JUDGES and to CONTROL BY ADDING JUDGES.*

Proof. Membership in NP is obvious for both problems. Again, we show NP-hardness for $UPQR_{1/2}$ -EXACT CONTROL BY ADDING JUDGES and $UPQR_{1/2}$ -CONTROL BY ADDING JUDGES at the same time, by a reduction from the NP-complete problem X3C. Given an X3C instance (X, C) with $X = \{x_1, \dots, x_{3m}\}$ and $C = \{C_1, \dots, C_n\}$, define the following judgment aggregation scenario. The agenda Φ contains $\{\alpha_0, \alpha_1, \dots, \alpha_{3m}\}$ and their negations. The quota $q = 1/2$ holds for every positive literal. The profile of the individual judgment sets initially taking part in the process is $\mathbf{T} = (T_1, \dots, T_{m+1})$ with $T_1 = \{\alpha_0, \alpha_1, \dots, \alpha_{3m}\}$, $T_i = \{\neg\alpha_0, \alpha_1, \dots, \alpha_{3m}\}$, $2 \leq i \leq m$, and $T_{m+1} = \{\neg\alpha_0, \neg\alpha_1, \dots, \neg\alpha_{3m}\}$. The profile of the judges who can be added is $\mathbf{S} = (S_1, \dots, S_n)$ with $S_i = \{\alpha_0, \alpha_j, \neg\alpha_\ell \mid x_j \in C_i, x_\ell \notin C_i, 1 \leq j, \ell \leq 3m\}$. The maximum number of judges from \mathbf{S} who can be added is m . The desired outcome of the external person is $J = \{\alpha_0, \alpha_1, \dots, \alpha_{3m}\}$. Then it holds, that there is a profile $\mathbf{S}' \subseteq \mathbf{S}$, $|\mathbf{S}'| \leq m$, such that $H(J, F(\mathbf{T} \cup \mathbf{S}')) < H(J, F(\mathbf{T}))$ if and only if there is an exact cover for the given X3C instance. The collective judgment set for $UPQR_{1/2}(\mathbf{T})$ is $\{\neg\alpha_0, \alpha_1, \dots, \alpha_{3m}\}$. Observe that $H(J, F(\mathbf{T})) = 1$, since the only difference lies in α_0 . Hence, $F(\mathbf{T} \cup \mathbf{S}')$ must be exactly J , and the reduction will hold for both problems at hand.

(\Leftarrow) Assume that there is an exact cover $C' \subseteq C$ for the given X3C instance (X, C) . Then the profile \mathbf{S}' contains those judges S_i with $C_i \in C'$. The total number of judges is then $2m + 1$. The number of affirmations needed to be in the collective judgment set is strictly greater than $m + (1/2)$, so $m + 1$ affirmations are needed. Note that α_0 gets one affirmation from the judges in \mathbf{T} and m affirmations from the judges in \mathbf{S}' . Every α_i , $1 \leq i \leq 3m$, gets m affirmations from the judges in \mathbf{T} and one affirmation from a judge in \mathbf{S}' . Hence, the collective judgment set is J , as desired.

(\Rightarrow) Assume that there is a profile \mathbf{S}' with $|\mathbf{S}'| \leq m$ such that $UPQR_{1/2}(\mathbf{T} \cup \mathbf{S}') = J$. Since α_0 is contained in the collective judgment set it must receive enough affirmations of the judges in \mathbf{S}' . Adding less than m new affirmations for α_0 is not enough, since $m - 1 \leq (2m)(1/2)$, but since $(2m + 1)(1/2) < m + 1$, m new affirmations are enough. As above, if there is a total number of $2m + 1$ judges then the number of affirmations needed for a positive formula to be in the collective judgment set is $m + 1$. Since the α_i , $1 \leq i \leq 3m$, receive only m affirmations from \mathbf{T} , they must all get one additional affirmation from \mathbf{S}' . Since $|\mathbf{S}'| \leq m$ and every judge affirms of exactly four formulas, including α_0 , the sets C_i corresponding to the judges in \mathbf{S}' must form an exact cover for the given X3C instance. \square

One important point regarding the proof of Theorem 5.9 is that the agenda contains only premises. In contrast to the NP-hardness result obtained here, we showed in Proposition 5.1 that $UPQR_q$ is strategy-proof for each rational quota q , $0 \leq q < 1$, if the agenda contains only premises. Similarly Theorem 5.6 shows that *PBP-EXACT MICROBRIBERY* is also in P if the desired judgment set contains only premises. We now show that this is not the case for *UPQR_{1/2}-EXACT CONTROL BY DELETING JUDGES*.

Theorem 5.10. *UPQR_{1/2} is resistant to EXACT CONTROL BY DELETING JUDGES and to CONTROL BY DELETING JUDGES.*

Proof. Membership in NP is obvious for both problems. We first prove NP-hardness for *UPQR_{1/2}-EXACT CONTROL BY DELETING JUDGES* and show afterwards how the construction must be modified to also hold for the case of *UPQR_{1/2}-CONTROL BY DELETING JUDGES*. The proof of NP-hardness for the exact problem is by a reduction from the NP-complete X3C problem. Given an X3C instance (X, C) with $X = \{x_1, \dots, x_{3m}\}$ and $C = \{C_1, \dots, C_n\}$, we assume that every element from X occurs in at least one set from C . If this is not the case, it is a no-instance for X3C (and we then map to an easily constructed no-instance of our exact control problem in judgment aggregation). Now we construct the following judgment aggregation scenario. The agenda Φ contains $\beta, \alpha_0, \dots, \alpha_{3m}$ and their negations, and the quota is $1/2$ for every positive literal. The complete profile is $\mathbf{T} = \mathbf{T}_1 \cup \mathbf{T}_2$, where $\mathbf{T}_1 = \{J_1, \dots, J_{n+m}\}$ and $\mathbf{T}_2 = \{L_1, \dots, L_n\}$, so $|\mathbf{T}| = 2n + m$. The individual judgment sets J_i from \mathbf{T}_1 consist of the set $\{\alpha_j, \neg\alpha_\ell \mid m + d_j \geq i, m + d_\ell < i, 1 \leq j, \ell \leq 3m\}$, for $1 \leq i \leq n + m$, where d_i is the number of sets C_j in which x_i occurs. Note that α_0 is contained in J_i if $i \leq n + 1$, and $\neg\alpha_0$ otherwise; and that β is contained in J_i if $i \leq m$, and $\neg\beta$ otherwise. The individual judgment sets from \mathbf{T}_2 are $L_i = \{\beta, \neg\alpha_0, \alpha_j, \neg\alpha_\ell \mid x_j \notin C_i, x_\ell \in C_i, 1 \leq j, \ell \leq 3m\}$, $1 \leq i \leq n$. The desired outcome is $J = \{\neg\beta, \alpha_0, \dots, \alpha_{3m}\}$, and at most m judges can be deleted. The outcome of $UPQR_{1/2}(\mathbf{T})$ is $\{\beta, \neg\alpha_0, \alpha_1, \dots, \alpha_{3m}\}$, since α_0 receives $n + 1$ affirmations, and β and all α_i , $1 \leq i \leq 3m$, receive $n + m$ affirmations each. The only difference

between the actual outcome and the desired judgment set is that α_0 and $\neg\beta$ are not contained in the collective judgment set.

Assume that there is an exact cover $C' \subseteq C$ with $|C'| = m$. By deleting the judges L_i corresponding to this exact cover, we have that α_0 still receives $n + 1$ affirmations, and each α_i , $1 \leq i \leq 3m$, loses exactly $m - 1$ affirmations and has $n + 1$ affirmations. Also, β loses m affirmations and has only n affirmations left. The number of participating judges is now $2n$, so $n + 1$ affirmations are needed to be in the collective judgment set. As desired, the collective outcome is $\{\neg\beta, \alpha_0, \alpha_1, \dots, \alpha_m\}$.

Conversely, assume that by deleting at most m judges, the collective outcome is $\{\neg\beta, \alpha_0, \alpha_1, \dots, \alpha_m\}$. Observe that exactly m judges must be deleted, since α_0 has only $n + 1$ affirmations and hence can be in the collective judgment set only if at least m voters that do not affirm α_0 are deleted. Furthermore, β should not be contained in the collective outcome. The number of initial affirmations for β is $n + m$. If the number of judges participating is reduced by m , $n + 1$ affirmations suffice to be in the collective judgment set. This implies that only judges having β in their individual judgment sets can be deleted. In total, this means that all m judges must be deleted from the set \mathbf{T}_2 . If there is one x_i which is not contained in any of sets C_j corresponding to the deleted judges, α_i loses m affirmations, and would no longer be contained in the collective judgment set. But if every α_i is deleted exactly $m - 1$ times, they all have $n + 1$ affirmations and are all contained in the collective judgment set. Hence the sets C_i corresponding to the deleted judges must form an exact cover for the given X3C instance.

With a slight adaption of the above proof we can also show that *PBP-CONTROL BY DELETING JUDGES* is NP-hard. The agenda is extended by the formula $\alpha_0 \wedge \neg\beta$ and its negation, and the desired judgment set J is incomplete and contains α_i , $1 \leq i \leq 3m$, and the formula $\alpha_0 \wedge \neg\beta$. The Hamming distance from J to the original outcome is 1, since $\alpha_0 \wedge \neg\beta$ is evaluated to false for the collective judgment set. By the same reasoning as above, there is an exact cover for the given X3C instance if and only if it is possible to control the judgment aggregation scenario successfully by deleting at most m judges. \square

In the proof of NP-hardness for *PBP-EXACT CONTROL BY DELETING JUDGES* it again is the case that the agenda contains only premises, but for *PBP-CONTROL BY DELETING JUDGES* it remains open whether the problem is still NP-complete if the agenda contains only premises.

Furthermore, there are no natural restrictions that turn one of our control problems to be solvable in polynomial time, as was the case for manipulation and bribery.

Finally, we consider the newly introduced control type *CONTROL BY REPLACING JUDGES* specific to judgment aggregation. In contrast to the problems of control by adding and deleting judges, here the number of judges participating is

constant, as it was the case for the manipulation and bribery problems. Thus, there is again no difference between the uniform premise-based quota rule and the uniform constant premise-based quota rule. In the following theorem we show NP-completeness for both classes of rules in the Hamming distance based and the exact variant of control by replacing judges.

Theorem 5.11. *For each rational quota q , $0 \leq q < 1$, $UPQR_q$ is resistant to EXACT CONTROL BY REPLACING JUDGES and CONTROL BY REPLACING JUDGES.*

Proof. It can easily be seen that both problems belong to NP. Hardness for NP will be shown by a slight modification of the construction from the proof of Theorem 5.7. To make the reduction from this proof work for the case of replacing judges, we add k judges to the profile \mathbf{T} that potentially will be replaced by those in \mathbf{S} . These k new judgment sets from \mathbf{T} contain all negated formulas from the agenda. To ensure that only those judges can be replaced, we introduce one new formula, z , its negation, and n syntactic variations of it into the agenda. This formula is contained in all $q + 1$ individual judgment sets from the initial set \mathbf{T} . The judges from \mathbf{S} all contain $\neg z$. Furthermore, the desired judgment set J also contains z . For the exact variant, the same arguments as in the proof of Theorem 5.7 hold. For CONTROL BY REPLACING JUDGES, note that the Hamming distance of the original outcome to J is still n , and that replacing a judge that has z in its individual judgment set always results in a Hamming distance that is greater than $n + 1$. Again, only those judges that contain all negated formulas from Φ can be replaced, and the same arguments as in the proof of Theorem 5.7 apply. \square

5.4 Conclusions and Future Work

Following up a line of research initiated by Endriss et al. [EGP10a, EGP10b], we have studied the computational complexity of problems related to manipulation, bribery, and control in judgment aggregation. Our results for manipulation and bribery are summarized in Table 5.6, where the results for $UPQR_q$ -MANIPULATION hold for every rational q , $0 \leq q < 1$. “# of judges” stands for a fixed number of judges, “max # of changes” stands for the parameter “maximum number of changes in the premises needed in the manipulators judgment set”, in case of NP-completeness “# of bribes” means that only a fixed number of judges can be bribed and denotes the corresponding parameter in case of W[2]-hardness, and “# of microbribes” indicates that the number of microbribes is a fixed constant. The entry \mathbf{X} implies that the combination of parameter and problem is not applicable.

The unparameterized results for control by adding, deleting, and replacing judges are summarized in Table 5.7. Resistance for $UCPQR_q$ holds for all admissible values for q , and in the case of $UPQR_q$ resistance holds for all values of q , $0 \leq q < 1$.

Table 5.6: Overview of results for manipulation and bribery problems

	$UPQR_q$ -MANIPULATION	PBP -BRIBERY	PBP -EXACT BRIBERY	PBP -MICROBRIBERY	PBP -EXACT MICROBRIBERY
# of judges	NP-comp.	NP-comp.		NP-comp.	NP-comp.
max # of changes	W[2]-hard	x	x	x	x
# of bribes	x	NP-comp.	W[2]-hard	x	x
# of microbribes	x	x	x	NP-comp.	NP-comp.
general problem	NP-comp.	NP-comp.	NP-comp.	NP-comp.	NP-comp.

In particular, the complexity of bribery and control—though deeply investigated in the context of voting [FHH09, EFS09, FHHR09a, BTT92, HHR07, BEH⁺10, BEH⁺10]—has not been studied before in the context of judgment aggregation. For the four natural scenarios modeling different ways of bribery, we have shown that the corresponding decision problems are NP-complete even when some natural parameters are a fixed constant and one problem is shown to be W[2]-hard for a natural parametrization. In addition, extending the results of Endriss et al. [EGP10b] on the (classical) complexity of manipulation in judgment aggregation, we have obtained W[2]-hardness for the class of uniform premise-based quota rules, for each reasonable quota. It remains open, however, whether one can also obtain matching upper bounds in terms of parameterized complexity. We suspect that all W[2]-hardness results from this chapter in fact can be strengthened to W[2]-completeness results. For the three very natural types of control introduced for judgment aggregation, we obtained NP-hardness for all studied variants. But the complexity for control and exact control by adding or deleting candidates remains open for the uniform premise based quota rules for all values of q except $1/2$.

Faliszewski et al. [FHH09] introduced and studied also the “priced” and “weighted” versions of bribery in voting. These notions can be reasonably applied to bribery in judgment aggregation: The “priced” variant means that judges may request different amounts of money to be willing to change their judgments ac-

Table 5.7: Overview of results for control problems

Problem Type	$UCPQR_q$	$UPQR_{1/2}$	$UPQR_q$
CONTROL BY ADDING JUDGES	resistant	resistant	
EXACT CONTROL BY ADDING JUDGES	resistant	resistant	
CONTROL BY DELETING JUDGES	resistant	resistant	
EXACT CONTROL BY DELETING JUDGES	resistant	resistant	
CONTROL BY REPLACING JUDGES	resistant	resistant	resistant
EXACT CONTROL BY REPLACING JUDGES	resistant	resistant	resistant

ording to the briber’s will, and the “weighted” variant means that the judgments of some judges may be heavier than those of others. For example it is reasonable that the judgments of some experts are heavier than those of the remaining judges. Although we have not defined this in a formal setting here, note that our hardness results carry over to more general problem variants as well.

We have introduced three types of control for judgment aggregation. Two were derived from voting and one especially introduced for the context of judgment aggregation. Here, the question arises if there are any other control problems that can be defined in the context of judgment aggregation. In contrast to the manipulation and bribery problems, the parameterized complexity of the control problems has not been studied yet. Furthermore, there are other natural parameters or other natural judgment aggregation procedures that are worth investigating. Until now we only investigated the exact variant and the Hamming-distance based approach. One could also define manipulation, bribery, and control problems with respect to other natural distance functions between judgment sets.

A more interesting task for future research would be to try to complement our parameterized worst-case hardness results by studying typical-case behavior for these problems, as is currently done intensely in the context of voting (see, e.g., [RS12]).

In summary, the computational analysis of problems related to judgment aggregation is an interesting field of research and still leaves possibilities for further investigations.

Bibliography

- [Alo06] N. Alon. Ranking tournaments. *SIAM Journal on Discrete Mathematics*, 20(1):137–142, 2006.
- [AMO93] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, 1993.
- [Arr63] K. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 1951 (revised edition 1963).
- [BBCN12] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting - A parameterized complexity perspective. In H. Bodlaender, R. Downey, F. Fomin, and D. Marx, editors, *The Multivariate Algorithmic Revolution and Beyond: Essays Dedicated to Michael R. Fellows on the Occasion of His 60th Birthday*, pages 318–368. Springer-Verlag *Lecture Notes in Computer Science #7370*, 2012.
- [BBF⁺10] D. Baumeister, F. Brandt, F. Fischer, J. Hoffmann, and J. Rothe. The complexity of computing minimal unidirectional covering sets. In *Proceedings of the 7th International Conference on Algorithms and Complexity*, pages 299–310. Springer-Verlag *Lecture Notes in Computer Science #6078*, 2010.
- [BBF⁺12] D. Baumeister, F. Brandt, F. Fischer, J. Hoffmann, and J. Rothe. The complexity of computing minimal unidirectional covering sets. *Theory of Computing Systems*, 2012. Accepted subject to minor revision.
- [BCE12] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiß, editor, *Multiagent Systems*. MIT Press, 2012. 2nd edition, to appear.
- [BD10] N. Betzler and B. Dorn. Towards a dichotomy of finding possible winners in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.
- [BEER12a] D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. Bribery and control in judgment aggregation. In *Proceedings of the 4th International Workshop on Computational Social Choice*, 2012. To appear.

- [BEER12b] D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. Control in judgment aggregation. In *Proceedings of the 6th European Starting AI Researcher Symposium*, 2012. To appear.
- [BEH⁺10] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editor, *Handbook of Approval Voting*, pages 199–251. Springer, 2010.
- [BER11] D. Baumeister, G. Erdélyi, and J. Rothe. How hard is it to bribe the judges? A study of the complexity of bribery in judgment aggregation. In *Proceedings of the 2nd International Conference on Algorithmic Decision Theory*, pages 1–15. Springer-Verlag *Lecture Notes in Artificial Intelligence #6992*, 2011.
- [BF78] S. Brams and P. Fishburn. Approval voting. *American Political Science Review*, 72(3):831–847, 1978.
- [BF02] S. Brams and P. Fishburn. Voting procedures. In K. Arrow, A. Sen, and K. Suzumura, editors, *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. North-Holland, 2002.
- [BF08] F. Brandt and F. Fischer. Computing the minimal covering set. *Mathematical Social Sciences*, 56(2):254–268, 2008.
- [BFH09] F. Brandt, F. Fischer, and P. Harrenstein. The computational complexity of choice sets. *Mathematical Logic Quarterly*, 55(4):444–459, 2009.
- [BFHM10] F. Brandt, F. Fischer, P. Harrenstein, and M. Mair. A computational analysis of the tournament equilibrium set. *Social Choice and Welfare*, 34(4):597–609, 2010.
- [BFLR12] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 577–584. IFAAMAS, 2012.
- [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*, pages 53–58. IJCAI, 2009.
- [Bla58] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

-
- [BNW11] N. Betzler, R. Niedermeier, and G. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 55–60. IJCAI, 2011.
- [BO91] J. Bartholdi III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
- [Bor81] J.-C. de Borda. Mémoire sur les élections au scrutin. *Histoire de L’Académie Royale des Sciences, Paris*, 1781. English translation appears in [Gra53].
- [Bor83] G. Bordes. On the possibility of reasonable consistent majoritarian choice: Some positive results. *Journal of Economic Theory*, 31:122–132, 1983.
- [BR06] L. Bovens and W. Rabinowicz. Democratic answers to complex questions – an epistemic perspective. *Synthese*, 150(1):131–153, 2006.
- [BR10] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. In *Proceedings of the 19th European Conference on Artificial Intelligence*, pages 1019–1020. IOS Press, 2010. Short paper.
- [BR12] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, 2012.
- [BRR11] D. Baumeister, M. Roos, and J. Rothe. Computational complexity of two variants of the possible winner problem. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 853–860. IFAAMAS, 2011.
- [BRR⁺12] D. Baumeister, M. Roos, J. Rothe, L. Schend, and L. Xia. The possible winner problem with uncertain weights. In *Proceedings of the 20th European Conference on Artificial Intelligence*. IOS Press, 2012. To appear.
- [BS06] S. Brams and R. Sanver. Critical strategies under approval voting: Who gets ruled in and ruled out. *Electoral Studies*, 25(2):287–305, 2006.
- [BS09] S. Brams and R. Sanver. Voting systems that combine approval and preference. In S. Brams, W. Gehrlein, and F. Roberts, editors, *The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn*, pages 215–237. Springer, 2009.

- [BTT89] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.
- [BTT92] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical Comput. Modelling*, 16(8/9):27–40, 1992.
- [CELM07] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *Proceedings of the 33rd Conference on Current Trends in Theory and Practice of Computer Science*, pages 51–69. Springer-Verlag *Lecture Notes in Computer Science* #4362, 2007.
- [CFRS07] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, 2007.
- [CGH⁺88] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy I: Structural properties. *SIAM Journal on Computing*, 17(6):1232–1252, 1988.
- [CGH⁺89] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. The boolean hierarchy II: Applications. *SIAM Journal on Computing*, 18(1):95–111, 1989.
- [CLM⁺12] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 2012. To appear.
- [CLMM10] Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible winners when new candidates are added: The case of scoring rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 762–767. AAAI Press, 2010.
- [Con85] J.-A.-N. de Caritat, Marquis de Condorcet. Essai sur l’application de l’analyse à la probabilité des décisions rendues à la pluralité des voix. 1785. Facsimile reprint of original published in Paris, 1972, by the Imprimerie Royale. English translation appears in I. McLean and A. Urken, *Classics of Social Choice*, University of Michigan Press, 1995, pages 91–112.
- [Con06] V. Conitzer. Computing Slater rankings using similarities among candidates. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 613–619. AAAI Press, 2006.

-
- [Con10] V. Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.
- [Cop51] A. Copeland. A “reasonable” social welfare function. Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan, 1951.
- [CS02] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 314–319. AAAI Press, 2002.
- [CS05] V. Conitzer and T. Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence*, pages 145–152. AUAI Press, 2005.
- [CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.
- [Deb87] B. Debord. *Axiomatisation de procédures d’agrégation de préférences*. PhD thesis, Université Scientifique, Technologique et Médicale de Grenoble, 1987.
- [DF99] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.
- [DKNS01] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 613–622. ACM Press, 2001.
- [DKNW11] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for Borda manipulation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 657–662. AAAI Press, 2011.
- [DL99] B. Dutta and J.-F. Laslier. Comparison functions and choice correspondences. *Social Choice and Welfare*, 16(4):513–532, 1999.
- [DL07a] F. Dietrich and C. List. Arrow’s theorem in judgment aggregation. *Social Choice and Welfare*, 29(1):19–33, 2007.
- [DL07b] F. Dietrich and C. List. Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics*, 19(4):391–424, 2007.

- [DL07c] F. Dietrich and C. List. Strategy-proof judgment aggregation. *Economics and Philosophy*, 23(3):269–300, 2007.
- [Dod76] C. Dodgson. A method of taking votes on more than two issues. Pamphlet printed by the Clarendon Press, Oxford, and headed “not yet published” (see the discussions in [MU95, Bla58], both of which reprint this paper), 1876.
- [Dun95] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77:321–357, 1995.
- [Dut88] B. Dutta. Covering sets and a new Condorcet choice correspondence. *Journal of Economic Theory*, 44:63–80, 1988.
- [EE12] E. Elkind and G. Erdélyi. Manipulation under voting rule uncertainty. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 627–634. IFAA-MAS, 2012.
- [EFG⁺09] G. Erdélyi, H. Fernau, J. Goldsmith, N. Mattei, D. Raible, and J. Rothe. The complexity of probabilistic lobbying. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory*, pages 86–97. Springer-Verlag *Lecture Notes in Artificial Intelligence* #5783, 2009.
- [EFS09] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 299–310. Springer-Verlag *Lecture Notes in Computer Science* #5814, 2009.
- [EFS10] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 768–773. AAAI Press, 2010.
- [EGP10a] U. Endriss, U. Grandi, and D. Porello. Complexity of judgment aggregation: Safety of the agenda. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 359–366. IFAAMAS, 2010.
- [EGP10b] U. Endriss, U. Grandi, and D. Porello. Complexity of winner determination and strategic manipulation in judgment aggregation. In V. Conitzer and J. Rothe, editors, *Proceedings of the 3rd International Workshop on Computational Social Choice*, pages 139–150. Universität Düsseldorf, 2010.

-
- [Eme] P. Emerson. The original Borda count and partial voting. *Social Choice and Welfare*. To appear.
- [End11] U. Endriss. Computational social choice: Prospects and challenges. *Procedia Computer Science*, 7:68–72, 2011. Proceedings of the 2nd European Future Technologies Conference and Exhibition (FET’11).
- [ER91] E. Ephrati and J. Rosenschein. The Clarke Tax as a consensus mechanism among automated agents. In *Proceedings of the 9th National Conference on Artificial Intelligence*, pages 173–178. AAAI Press, 1991.
- [ER93] E. Ephrati and J. Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pages 423–429. Morgan Kaufmann, 1993.
- [Fal08] P. Faliszewski. Nonuniform bribery. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1569–1572. ACM Press, 2008.
- [FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2006.
- [FHH09] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.
- [FHH10] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.
- [FHHR09a] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.
- [FHHR09b] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, chapter 14, pages 375–406. Springer, 2009.

- [FHS08] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 983–990. IFAAMAS, 2008.
- [FHS10] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Manipulation of Copeland elections. In *Proceedings of the 9th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 367–374. IFAAMAS, 2010.
- [Fis77] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.
- [FKS03] R. Fagin, R. Kumar, and D. Sivakumar. Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data*, pages 301–312. ACM Press, 2003.
- [FP10] P. Faliszewski and A. Procaccia. AI’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.
- [Gib73] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.
- [GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.
- [Gra53] A. de Grazia. Mathematical deviation of an election system. *Isis*, 44:41–51, 1953.
- [Hem87] L. Hemachandra. The strong exponential hierarchy collapses. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 110–122. ACM Press, 1987.
- [HH07] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.
- [HHM12] E. Hemaspaandra, L. Hemaspaandra, and C. Menton. Search versus decision for election manipulation problems. Technical Report arXiv:cs/1202.6641v2 [cs.GT], ACM Computing Research Repository (CoRR), 2012.
- [HHR97a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll’s 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

-
- [HHR97b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Raising NP lower bounds to parallel NP lower bounds. *SIGACT News*, 28(2):2–13, 1997.
- [HHR07] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.
- [HHR08] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. Technical Report arXiv:cs/0608057v2 [cs.GT], ACM Computing Research Repository (CoRR), 2008. Conference version appeared in *Proc. IJCAI’07*. Journal version appeared in *Mathematical Logic Quarterly*, 55(4):397–424, 2009.
- [HP01] G. Hägele and F. Pukelsheim. The electoral writings of Ramon Llull. *Studia Lulliana*, 41(97):3–38, 2001.
- [HR98] E. Hemaspaandra and J. Rothe. Recognizing when greed can approximate maximum independent sets is complete for parallel access to NP. *Information Processing Letters*, 65(3):151–156, 1998.
- [HRS06] E. Hemaspaandra, J. Rothe, and H. Spakowski. Recognizing when heuristics can approximate minimum vertex covers is complete for parallel access to NP. *R.A.I.R.O. Theoretical Informatics and Applications*, 40(1):75–91, 2006.
- [HSV05] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.
- [HW02] E. Hemaspaandra and G. Wechsung. The minimization problem for boolean formulas. *SIAM Journal on Computing*, 31(6):1948–1958, 2002.
- [Kem59] J. Kemeny. Mathematics without numbers. *Dædalus*, 88:577–591, 1959.
- [KL05] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, 2005.
- [KS86] L. A. Kornhauser and L. G. Sager. Unpacking the court. *Yale Law Journal*, 96(1):82–117, 1986.
- [KSW87] J. Köbler, U. Schöning, and K. Wagner. The difference and truth-table hierarchies for NP. *R.A.I.R.O. Informatique théorique et Applications*, 21:419–435, 1987.

- [Las97] J. Laslier. *Tournament Solutions and Majority Voting*. Springer-Verlag, 1997.
- [Lis] C. List. The theory of judgment aggregation: An introductory review. *Synthese*. To appear.
- [Lis06] C. List. The discursive dilemma and public reason. *Ethics*, 116(2):362–402, 2006.
- [LLS75] R. Ladner, N. Lynch, and A. Selman. A comparison of polynomial time reducibilities. *Theoretical Computer Science*, 1(2):103–124, 1975.
- [LP02] C. List and P. Pettit. Aggregating sets of judgments: An impossibility result. *Economics and Philosophy*, 18(1):89–110, 2002.
- [LP04] C. List and P. Pettit. Aggregating sets of judgments: Two impossibility results compared. *Synthese*, 140(1–2):207–235, 2004.
- [LP09] C. List and C. Puppe. Judgment aggregation: A survey. In P. Anand, P. Pattanaik, and C. Puppe, editors, *The Handbook of Rational and Social Choice*, chapter 19. Oxford University Press, 2009.
- [LPR⁺12] J. Lang, M. Pini, F. Rossi, D. Salvagnin, K. Venable, and T. Walsh. Winner determination in voting trees with incomplete preferences and weighted votes. *Journal of Autonomous Agents and Multi-Agent Systems*, 25(1):130–157, 2012.
- [McG53] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.
- [Mil80] N. Miller. A new solution set for tournaments and majority voting: Further graph-theoretical approaches to the theory of voting. *American Journal of Political Science*, 24(1):68–96, 1980.
- [MPS08] J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson’s rule. *Social Choice and Welfare*, 31(2):311–330, 2008.
- [MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.
- [MU95] I. McLean and A. Urken. *Classics of Social Choice*. University of Michigan Press, Ann Arbor, Michigan, 1995.

-
- [NWX11] N. Narodytska, T. Walsh, and L. Xia. Manipulation of nanson’s and baldwin’s rules. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pages 713–718. AAAI Press, 2011.
- [Pap95] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 2nd edition, 1995. Reprinted with corrections.
- [Pet01] P. Pettit. Deliberative democracy and the discursive dilemma. *Philosophical Issues*, 11:268–299, 2001.
- [PZ83] C. Papadimitriou and S. Zachos. Two remarks on the power of counting. In *Proceedings of the 6th GI Conference on Theoretical Computer Science*, pages 269–276. Springer-Verlag *Lecture Notes in Computer Science #145*, 1983.
- [RBLR11] J. Rothe, D. Baumeister, C. Lindner, and I. Rothe. *Einführung in Computational Social Choice: Individuelle Strategien und kollektive Entscheidungen beim Spielen, Wählen und Teilen*. Spektrum Akademischer Verlag, 2011. In german.
- [Rot05] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer, 2005.
- [RS12] J. Rothe and L. Schend. Typical-case challenges to complexity shields that are supposed to protect elections against manipulation and control: A survey. In *Website Proceedings of the Special Session on Computational Social Choice at the 12th International Symposium on Artificial Intelligence and Mathematics*, 2012.
- [RSV03] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, 36(4):375–386, 2003.
- [Sat75] M. Satterthwaite. Strategy-proofness and Arrow’s conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.
- [Sch78] T. Schaefer. The complexity of satisfiability problems. In *Proceedings of the 10th ACM Symposium on Theory of Computing*, pages 216–226. ACM Press, 1978.
- [Tay95] A. Taylor. *Mathematics and Politics*. Springer-Verlag, 1995.
- [Tay05] A. Taylor. *Social Choice and the Mathematics of Manipulation*. Cambridge University Press, 2005.

- [Tid06] N. Tideman. *Collective Decisions and Voting: The Potential for Public Choice*. Ashgate Publishing, 2006.
- [Wag87] K. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.
- [Woe03] G. Woeginger. Banks winners in tournaments are difficult to recognize. *Social Choice and Welfare*, 20:523–528, 2003.
- [XC11] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.
- [XCP10] L. Xia, V. Conitzer, and A. Procaccia. A scheduling approach to coalitional manipulation. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 275–284. ACM Press, 2010.
- [XLM11] L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: New results coming up! In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 82–836. IFAAMAS, 2011.
- [XZP⁺09] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 348–353. IJCAI, 2009.
- [You77] H. Young. Extending Condorcet’s rule. *Journal of Economic Theory*, 16(2):335–353, 1977.
- [ZLR11] M. Zuckerman, O. Lev, and J. Rosenschein. An algorithm for the coalitional manipulation problem under maximin. In *Proceedings of the 10th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 845–852. IFAAMAS, 2011.
- [ZPR09] M. Zuckerman, A. Procaccia, and J. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, 173(2):392–412, 2009.