The Control Complexity of Sincere-Strategy Preference-Based Approval Voting and of Fallback Voting, and a Study of Optimal Lobbying and Junta Distributions for SAT

Inaugural-Dissertation

zur

Erlangung des Doktorgrades der

Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von

Gábor Erdélyi

aus Baja

Düsseldorf, im Januar 2009

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

Tag der mündlichen Prüfung: 16.03.2009

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Dissertation selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Teile dieser Arbeit wurden bereits in den folgenden Schriften veröffentlicht bzw. zur Publikation angenommen: [EHRS07a, EHRS07b, ENR08a, ENR08b, BEH$^+$]. Einige Ergebnisse des Forschungsberichts [EHRS07b] wurden bei der Zeitschrift *Theoretical Computer Science* unter dem Titel "Generalized Juntas an NP-Hard Sets" und andere Ergebnisse von [EHRS07b] wurden bei der Zeitschrift *Information Processing Letters* unter dem Titel "Frequency of Correctness versus Average Polynomial Time" eingereicht. Einige Ergebnisse des Forschungsberichts [ENR08b] wurden bei der Zeitschrift *Mathematical Logic Quarterly* für die Sonderausgabe *Logic and Complexity within Computational Social Choice* unter dem Titel "Sincere-Strategy Prefernce-Based Approval Voting Fully Resists Constructive Control and Broadly Resists Destructive Control" eingereicht.

Düsseldorf, 16. Januar 2009 Gábor Erdélyi

Acknowledgments

First of all, I am deeply grateful to my advisor Jörg Rothe for giving me the chance to join his research team. Without his support, help, and inspiring discussions I would never have had the chance to gain ground in the scientific community.

Moreover, I would like to thank Egon Wanke for serving as a referee for this thesis.

I further thank all my coauthors, including Dorothea Baumeister, Dagmar Bruß, Edith Hemaspaandra, Lane A. Hemaspaandra, Markus Nowak, Tim Meyer, Tobias Riege, Jörg Rothe, and Holger Spakowski.

My research was supported in parts by the German Science Foundation (DFG) under the grant RO 1202/11-1.

I am also grateful to all my friends for proofreading my thesis, including Dorothea Baumeister, Shane Foster, Frank Gurski, Henriett Győry, Ágnes and Christian Klapka, Andreas Krause, Magnus Roos, and Tina Siepmann. Thank you guys, you were a great help!

Furthermore, I thank all my colleagues at the Department of Computer Science of the Heinrich-Heine-University Düsseldorf, Dorothea Baumeister, Claudia Forstinger, Frank Gurski, Christian Knieling, Guido Königstein, Claudia Lindner, Tobias Riege, Magnus Roos, Jörg Rothe, and Holger Spakowski.

I thank the Department of Computer Science at the University of Trier, especially Henning Fernau, for hosting my visit in Trier.

Above all, I am deeply grateful to my family, especially to my parents for making everything possible, to my brother for always cheering me up, and, primarily, to my wife, Olivia, for her love, encouragement, and endless support.

Zusammenfassung

Wahlverfahren werden nicht nur in der Politik, sondern auch in vielen Gebieten der Informatik eingesetzt (z.B. bei Multi-Agenten-Systemen in der Künstlichen Intelligenz). Brams und Sanver [BS06] führten zwei Wahlsysteme ein, die das bekannte *Approval Voting* modifizieren: *Sincere-strategy Preference-based Approval Voting* (SP-AV) und *Fallback Voting* (FV). Diese beiden Wahlsysteme werden in der vorliegenden Arbeit im Hinblick auf Wahlkontrolle untersucht. In solchen Kontrollszenarien versucht ein externer Agent, das Wahlergebnis durch Hinzufügen/Entfernen/Partitionieren von entweder Kandidaten oder Wählern zu beeinflussen.

Es wird gezeigt, dass SP-AV gegen 19 der üblichen 22 Typen von Wahlkontrolle resistent ist, d.h., die entsprechenden Kontrollprobleme sind NP-hart). Unter allen natürlichen Wahlsystemen, deren Sieger in Polynomialzeit bestimmt werden können, besitzt SP-AV somit die meisten Resistenzen gegen Wahlkontrolle. Insbesondere ist SP-AV (nach *Copeland Voting*, siehe Faliszewski et al. [FHHR08a]) das zweite solche Wahlsystem, das gegen alle Typen der konstruktiven Wahlkontrolle resistent ist. Anders als *Copeland Voting* ist SP-AV jedoch auch weitgehend resistent gegen die destruktiven Kontrolltypen. Außerdem wird gezeigt, dass FV – ebenso wie SP-AV – vollständig resistent gegen alle Typen von Kandidatenkontrolle ist.

Christian et al. [CFRS06] zeigten, dass das Problem *Optimal Lobbying* im Sinne der parametrisierten Komplexität schwer zu lösen ist. In der vorliegenden Arbeit wird ein effizienter Algorithmus entworfen und analysiert, der sogar die verallgemeinerte Variante *Optimal Weighted Lobbying* dieses Problems in einem logarithmischen Faktor approximiert, und es wird gezeigt, dass dieser Approximationsfaktor für diesen Algorithmus nicht verbessert werden kann.

Weiterhin wird das Gewinnerproblem für Dodgson-Wahlen untersucht. Hemaspaandra, Hemaspaandra und Rothe [HHR97] zeigten, dass dieses Problem vollständig für parallelen Zugriff auf NP ist (d.h. vollständig für die Θ_2^p-Stufe der Polynomialzeit-Hierarchie). Homan und Hemaspaandra [HH06] stellten eine effiziente Heuristik vor, die unter geeigneten Voraussetzungen Dodgson-Gewinner mit einer garantierten Häufigkeit findet, d.h., diese Heuristik ist ein so genannter “frequently self-knowingly correct algorithm”. In der vorliegenden Arbeit wird dieser Algorithmus-Typ in Bezug zur Klasse *Average-Case Polynomial Time* (AvgP) gesetzt. Es wird gezeigt, dass *jedes* Verteilungsproblem in AvgP bezüglich der Gleichverteilung einen solchen *frequently self-knowingly correct algorithm* hat, der in Polynomialzeit läuft. Außerdem werden einige Eigenschaften des verwandten Begriffs *probability weight of correctness* hinsichtlich der von Procaccia and Rosenschein [PR07] eingeführten so genannten *Junta*-Verteilungen untersucht.

Abstract

While voting systems were originally used in political science, they are now also of central importance in various areas of computer science, such as artificial intelligence (in particular within multiagent systems). Brams and Sanver [BS06] introduced sincere-strategy preference-based approval voting (SP-AV) and fallback voting (FV), two election systems which combine the preference rankings of voters with their approvals of candidates. We study these two systems with respect to procedural control—settings in which an agent seeks to influence the outcome of elections via control actions such as adding/deleting/partitioning either candidates or voters.

We prove that SP-AV is computationally resistant (i.e., the corresponding control problems are NP-hard) to 19 out of 22 types of constructive and destructive control. Thus, for the 22 control types studied here, SP-AV has more resistances to control, by three, than is currently known for any other natural voting system with a polynomial-time winner problem. In particular, SP-AV is (after Copeland voting, see Faliszewski et al. [FHHR08a]) the second natural voting system with an easy winner-determination procedure that is known to have full resistance to constructive control, and unlike Copeland voting it in addition displays broad resistance to destructive control.

We show that FV has full resistance to candidate control.

We also investigate two hard problems related to voting, the optimal weighted lobbying problem and the winner problem for Dodgson elections. Regarding the former problem, Christian et al. [CFRS06] showed that optimal lobbying is intractable in the sense of parameterized complexity. We propose an efficient greedy algorithm that nonetheless approximates a generalized variant of this problem, optimal weighted lobbying, and thus the original optimal lobbying problem as well. We also show that the approximation ratio of this algorithm is tight.

The problem of determining Dodgson winners is known to be complete for parallel access to NP [HHR97]. Homan and Hemaspaandra [HH06] proposed an efficient greedy heuristic for finding Dodgson winners with a guaranteed frequency of success, and their heuristic is indeed a "frequently self-knowingly correct algorithm." We prove that every distributional problem solvable in polynomial time on the average with respect to the uniform distribution has a frequently self-knowingly correct polynomial-time algorithm. Furthermore, we study some features of probability weight of correctness with respect to Procaccia and Rosenschein's junta distributions [PR07].

Contents

List of Figures

List of Tables

Chapter 1

Introduction

> *"Elections in which great care is taken to prevent any explicit or hidden structural bias towards any one candidate, aside from those beneficial biases that naturally result from an electorate that is equally well informed about various assets and liabilities of each candidate."*
>
> *(Democracy Watch on fair elections)*

Computational social choice is a new field emerging at the interface of social choice theory and computer science. This new field has two main advantages. First, it applies techniques developed in computer science to problems from social choice theory, for example to study the complexity of problems related to voting (see, e.g., the survey by Faliszewski et al. [FHHR09]) or fair division (see, e.g., [BL05]). Second, it transfers concepts from social choice theory into computer science, such as preference aggregation.

The study of voting procedures is a central task within computational social choice. Voting provides a particularly useful method for preference aggregation and collective decision-making. While voting systems were originally used in political science, economics, and operations research, they are now also of central importance in various areas of computer science, such as artificial intelligence (in particular, within multiagent systems). In automated, large-scale computer settings, voting systems have been applied, e.g., for planning [ER93] and similarity search [FKS03], and have also been used in the design of recommender systems [GMHS99] and ranking algorithms [DKNS01], where they help to lessen the spam in meta-search web-page rankings. A meta-search engine can be viewed as a machine that treats conventional search engines as voters who rank web-pages, as candidates, resulting from a search query. A broad overview in computational social choice is presented by Chevaleyre et al. [CELM07].

For such applications, it is crucial to explore the computational properties of voting systems and, in particular, to study the complexity of problems related to voting (see, e.g., the survey by Faliszewski et al. [FHHR09]).

Now, what exactly are voting systems and elections? An election $E = (C,V)$ is specified by a finite set C of candidates and a finite collection V of voters who express their preferences over the candidates in C, where distinct voters may of course have the same preferences. A voting system is a set of rules to aggregate the voters' individual preferences in order to determine the winners of a given election.

Elections have a long history going back to Athenian democracy of the ancient Greece. The greek used a procedure called ostracism to expel prominent citizens from the city-state for ten years. Every citizen could scratch the name of another citizen they wished to expel on potshards (papyrus was at that time very expensive and had to be imported from Egypt, potshards, on the other hand, were easily available and affordable), and deposited them in urns. The citizen who was named most of all was expelled for ten years. This voting system, where each voter gives one point to his or her most desired candidate and zero to any other candidate, and where the winner is each candidate with the maximum number of points, is called plurality. Of course, the ancient greek have used elections not only for expelling citizens, but also to elect their government. The Greek philosopher Aristotle, a student of Plato and teacher of Alexander the Great, was one of the first who tried to compare better and worse forms of governments and democracies.

In the following centuries, nothing really mentionable happened in the matter of voting systems until the year 1785, when Marie-Jean-Antoine-Nicolas de Caritat, the Marquis de Condorcet, and Jean-Charles de Borda, two french mathematicians and political scientists, argued about whose voting system was better. Condorcet suggested a system based on strict preference rankings, where the candidate who defeats every other candidate in a head-to-head contest is the winner [Con85].

Many voting systems require voters to specify their preference rankings either over the whole set or a subset of candidates. We say that a voter $v \in V$ has a preference *weak order* $\succcurlyeq$ on C, if $\succcurlyeq$ is *transitive* (i.e., $x \succcurlyeq y$ and $y \succcurlyeq z$ imply $x \succcurlyeq z$) and *complete* (i.e., for any two distinct candidates $x, y \in C$, either $x \succcurlyeq y$ or $y \succcurlyeq x$). $x \succcurlyeq y$ means that voter v likes x at least as much as y. If ties are excluded in the voters' preference rankings, this leads to a *linear order* or *strict ranking*, denoted by $\succ$. A strict ranking is always antisymmetric (i.e., for any two distinct candidates $x, y \in C$ either $x \succ y$ or $y \succ x$ holds, but not both at the same time) and irreflexive (i.e., for each $x \in C$ the following does not hold: $x \succ x$). Condorcet's system clearly requires a strict ranking from each voter.

Borda's system (called Borda count) was also based on preference rankings, but he distributed points among the candidates in the following way. Let say we have m candidates. Each voter gives $m-1$ points to his or her most preferred candidate, $m-2$ points to his or her second favourite one and so on until the least preferred candidate, who then gets zero points. A winner is each candidate with the highest score.

Condorcet pointed out that Borda's system is very susceptible to strategic voting (i.e., if a voter v wants his or her top candidate, say a, to win, v could rank a's most serious opponent on last place, even if this candidate is not v's most despised candidate). Such an influence is also called *manipulation*. On the other hand, Condorcet's system itself had a major problem, in fact, Condorcet winners do not always exist.

Example 1.1. *Let $E = (C,V)$ be an election, where $C = \{a,b,c\}$ is the set of candidates and $V = \{v_1, v_2, v_3\}$ is the set of voters [1] with the following votes:*

$$\begin{array}{ccccccc} v_1 & : & a & \succ & b & \succ & c, \\ v_2 & : & b & \succ & c & \succ & a, \\ v_3 & : & c & \succ & a & \succ & b, \end{array}$$

where $a \succ b \succ c$ means that a is this voter's favourite candidate, b is his or her second favourite, and c is the voter's most despised candidate. It is easy to see that there is no Condorcet winner, since a beats b, b beats c, and c beats a in a head-to-head contest. This yields the strict cycle a, b, c, a. This problem is known as the Condorcet paradox.

Nearly a century later, in 1876, the mathematician Charles Dodgson (a.k.a. Lewis Carroll, the author of "Alice's Adventures in Wonderland") introduced a new voting system based on a combinatorial optimization problem, somehow related to Condorcet's system [Dod76]. Dodgson was most likely not aware of Condorcet's work, see Black [Bla58]. His idea was very simple and elegant. If there is a Condorcet winner, that candidate is undoubtedly also the Dodgson winner. In the absence of a Condorcet winner, then those candidates who are "closest" to being a Condorcet winner are the Dodgson winners. Dodgson defined the "closeness" of a candidate c to a Condorcet winner as the minimum number of sequential swaps between adjacent candidates in the voters' preference rankings that are needed to make c the Condorcet winner. Another century later, Hemaspaandra, Hemaspaandra, and Rothe proved that determining Dodgson winners is computationally hard [HHR97].

Despite of the fact that systems like Condorcet's system and Borda count require from each voter strict preference rankings over all candidates, the resulting societal preference allows ties. We will see later systems, where the voters don't necessarily have to specify a strict preference ranking over all candidates.

As the discussions of Condorcet and Borda have already shown, different voting systems could yield different winners. As an example, consider the following election.

Example 1.2. *Let $E = (C,V)$ be an election, where $C = \{a,b,c,d\}$ is the set of candidates and $V = \{v_1, v_2, v_3\}$ is the set of voters with the following votes:*

[1] In this thesis we will sometimes slightly abuse notation by writing "set of voters" instead of "multiset of voters."

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

If we use plurality rule, then candidate a is clearly the winner, since a has two first places, b one first place, and both c and d have no first places at all. On the other hand, if we apply the rules of Borda count, we get $score(a) = 6$, $score(b) = 7$, $score(c) = 4$, *and* $score(d) = 1$, *thus candidate b is the winner.*

Which one of the candidates would really deserve it to win in the previous example? Candidate *a* who had the most first places, or candidate *b* who was constantly in front positions. Well, the answer is, it depends.

In the early 1950's, the economist Kenneth Arrow deliberated over the question if it was possible to find a "fair" voting system, where fair is meant in the way that the system should satisfy some reasonably stated conditions. During his research he reached some insuperable barriers, which made him draw the shocking conclusion that, under certain assumptions, there can't possibly exist a "fair" voting system [Arr63]. Let's take a closer look at this devastating theorem, starting with Arrow's notion of a voting system. A *voting system* maps the voters' individual preference rankings into a single preference ranking. Arrow first stated five fairness criteria:

Universality There should be no restrictions on how voters can rank the candidates (except of transitivity[2]).

Nondictatorship The voting system should not be dependent on only one voter, that is, there sould never be a voter whose preference ranking is soever the societal preference ranking, regardless of the other votes.

Independence of Irrelevant Alternatives The voting system should determine the same ranking among a subset of candidates as it would for the whole set of candidates. If a voter changes his or her preference ranking outside this subset (thus, he changes the preference ranking for irrelevant alternatives), then this should not have any effect whatsoever on the societal preference ranking for the subset.

Citizen Sovereignty If a candidate *a* is ranked higher than candidate *b* in the societal preference ranking, then there has to be at least one voter who ranks *a* higher than *b*.

Monotonicity If a voter modifies his or her preference ranking by ranking a candidate higher in his or her profile, then this candidate can't be ranked lower in the societal preference as before the change.

[2]Transitivity seems to be a reasonable and fair restriction. It is natural to assume that if a candidate *a* beats *b*, and *b* beats *c*, then also *a* beats *c*.

Arrow's impossibility theorem says, that it is not possible to design a voting system with at least two voters and more than two candidates satisfying the five conditions stated above. For the proof of the theorem, see [Arr63]. For further discussions and interesting examples we draw the attention of the reader to [HK05]. A related line of research has shown that, in principle, all natural voting systems can be manipulated by strategic voters. Most notable among such results is the classical work of Gibbard [Gib73] and Satterthwaite [Sat75]. The study of strategy-proofness is still an extremely active and interesting area in social choice theory (see, e.g., Duggan and Schwartz [DS00]) and in artificial intelligence (see, e.g., Everaere et al. [EKM07]).

After Arrow's result, several social choice theorists and mathematicians tried to find a way to circumvent this paradox. They all agreed that the only solution is to weaken the criteria. One suggestion was a voting system called approval voting. In *approval voting*, each voter has to vote "Yes" or "No" for each candidate. The winners of the election are the candidates with the maximum number of "Yes" votes. At first glance, approval voting does not even satisfy the definition of a voting system in Arrow's sense, since the voters don't have to specify a preference ranking over all candidates. However, the ballots in approval voting can also be seen as a kind of preference ranking. Let us redefine approval voting in the following way, with two more conditions according to Hodge and Klima [HK05]: (i) Ballots must be preference weak orders, where some candidates (where also the empty set as well as the whole set of candidates are allowed) are tied for first place and all the other candidates are tied for last place. (ii) The societal preference order is determined by the number of first places that the candidates receive, and the candidate with the maximum number of first places is ranked highest and so on until the candidate with the minimum number of first places, who is then on the last place. It is immediately clear from this alternative definition that approval voting violates Arrow's universality condition.

Approval voting was introduced by Brams and Fishburn ([BF78, BF83], see also [BF02]), axiomatized by Fishburn [Fis78] and Sertel [Ser88], and analyzed by Brams and Fishburn [BF78, BF83] and Merrill [Mer88]. Of all single-ballot nonranked systems, Brams and Fishburn appealed for the use of approval voting, emphasizing that it is the only voting system allowing the voters to approve of an unrestricted number of candidates. In their enthusiasm about approval voting they even called it "the electoral reform of the twentieth century". In fact, approval voting is in use in many companies, states and institutions to elect officers, for example in the Institute for Operations Research and Management Science, in the American Mathematical Society, in the IEEE, or in the United Nations to elect the Secretary-General. To read more about approval voting we point the reader to the textbooks by Arrow [BF02] and Hodge and Klima [HK05].

In Chapter 2, we outline all definitions and problems relevant for this work, especially, in Section 2.1 a detailled discussion about the computational model we will use. In

Section 2.3 follows a short introduction of basic principles of parameterized complexity theory, and Section 2.4 provides two useful parameterized graph problems for this thesis.

Chapter 3 gives a review over elections and voting systems. In section 3.1, we will introduce all voting systems considered in this work, amongst others, sincere-strategy preference-based approval voting (SP-AV, for short), and fallback voting (FV, for short). Section 3.2 and Section 3.3 comprehend detailled discussions about the three main possibilities to affect the outcome of an election namely, bribery, manipulation, and control. In contrast to manipulation, where, as shown earlier, a group of manipulators change their preferences to make their favourite candidate win, in *bribery* an external agent seeks to influence the outcome of the election via changing some voters' preference lists (see Section 3.2 for the formal definitions of manipulation and bribery). In *electoral control*, an external actor—traditionally called *the chair*—seeks to influence the outcome of an election via procedural changes to the election's structure, such as adding/deleting/partitioning either candidates or voters (see Section 3.3 for the formal definitions of our control problems). We consider both *constructive* control (introduced by Bartholdi, Tovey, and Trick [BTT92]), where the chair's goal is to make a given candidate the unique winner, and *destructive* control (introduced by Hemaspaandra, Hemaspaandra, and Rothe [HHR07a]), where the chair's goal is to prevent a given candidate from being a unique winner.

We investigate those twenty types of constructive and destructive control that were studied for approval voting [HHR07a] along with two additional control types introduced by Faliszewski et al. [FHHR07a] for a voting system that was proposed by Brams and Sanver [BS06] as a combination of preference-based and approval voting.

The study of voting systems from a complexity-theoretic perspective was initiated by Bartholdi, Tovey, and Trick's series of seminal papers about the complexity of winner determination [BTT89b], manipulation [BTT89a], and procedural control [BTT92] in elections.

One of the simplest preference-based voting systems is plurality. The purpose of Chapter 4 is to show that Brams and Sanver's combined system (adapted here so as to keep its useful features even in the presence of control actions) combines the strengths, in terms of computational resistance to control, of plurality and approval voting.

Some voting systems are *immune* to certain types of control in the sense that it is never possible for the chair to reach his or her goal via the corresponding control action. Of course, immunity to any type of control is most desirable, as it unconditionally shields the voting system against this particular control type. Unfortunately, like most voting systems, approval voting is *susceptible* (i.e., not immune) to many types of control, and plurality voting is susceptible to all types of control. However, and this was Bartholdi, Tovey, and Trick's brilliant insight [BTT92], even for systems susceptible to control, the chair's task of controlling a given election may be too hard computationally (namely, NP-hard, see Definition 2.3) for him or her to succeed. The voting system is then said to be *resistant* to this control type. On the other hand, if a voting system is susceptible to some type of control, but the chair's task can be solved in polynomial time, the system is said to be *vulnerable* to this control type.

The quest for a natural voting system with an easy winner-determination procedure that is universally resistant to control has lasted for more than 15 years now. Among the voting systems that have been studied with respect to control are plurality, Condorcet, approval, cumulative, Llull, and (variants of) Copeland voting [BTT92, HHR07a, HHR07b, PRZ07, FHHR07a, FHHR08a, BU08]. Among these systems, plurality and Copeland voting (denoted Copeland$^{0.5}$ in [FHHR08a]) display the broadest resistance to control, yet even they are not universally control-resistant. The only system currently known to be fully resistant—to the 20 types of constructive and destructive control studied in [HHR07a, HHR07b]—is a highly artificial system constructed via hybridization [HHR07b]. (It should be mentioned that this system was not designed for direct, real-world use as a "natural" system but was rather intended to rule out the existence of an Arrow-like impossibility theorem [HHR07b].)

As mentioned above, in Chapter 4 we study a voting system that combines the voters' preference rankings with their approvals/disapprovals of the candidates in a natural way. While approval voting nicely distinguishes between each voter's acceptable and inacceptable candidates, it ignores the preference rankings the voters may have about their approved (or disapproved) candidates. This shortcoming motivated Brams and Sanver [BS06] to introduce a voting system that combines approval and preference-based voting, and they defined the related notions of sincere and admissible approval strategies, which are quite natural requirements. We adapt their sincere-strategy preference-based approval voting system in a natural way such that, for elections with at least two candidates, admissibility of approval strategies (see Definition 4.1) can be ensured even in the presence of control actions such as deleting candidates and partitioning candidates or voters. The purpose of Chapter 4 is to study if, and to what extent, this hybrid system (where "hybrid" is not meant in the sense of [HHR07b] but refers to combining preference-based with approval voting in the sense of Brams and Sanver [BS06]) inherits the control resistances of plurality (which is perhaps the simplest preference-based system) and approval voting. We show that SP-AV, in fact, does combine all the resistances of plurality and approval voting. In addition, we show that SP-AV is even resistant to a control type (namely, "destructive control by partition of voters in model TE," see Section 4.2 and Table 4.1) to which both plurality and approval are vulnerable.

More specifically, we prove that sincere-strategy preference-based approval voting is resistant to 19 and vulnerable to only three of the 22 types of control considered here. In comparison, Condorcet voting is resistant to three and immune to four control types leaving seven vulnerabilities, approval voting is resistant to four and immune to nine control

Number of	Condorcet	Approval	Copeland	Plurality	SP-AV	FV
resistances	3	4	15	16	19	≥ 14
immunities	4	9	0	0	0	0
vulnerabilities	7	9	7	6	3	≤ 8
References	[BTT92, HHR07a]	[BTT92, HHR07a]	[FHHR07a, FHHR08a]	[BTT92, HHR07a, FHHR07a]	[HHR07a, ENR08b], this thesis	this thesis

Table 1.1: Comparison of voting systems with respect to control.

types leaving enine vulnerabilities, Copeland voting is resistant to 15 control types leaving seven vulnerabilities, and plurality is resistant to 16 control types leaving six vulnerabilities. Table 1.1 shows the number of resistances, immunities, and vulnerabilities to our 22 control types that are known for each of Condorcet, approval, plurality, and Copeland voting (see [BTT92, HHR07a, FHHR08a]), SP-AV (see Theorem 4.1 and Table 4.1 in Section 4.2.1), and for fallback voting (see Theorem 5.1).

Note that Table 1.1 lists only 14 instead of 22 control types for Condorcet voting. This is due to the fact that, on one hand, Condorcet winners must be unique if they exist at all (so it doesn't make sense to distinguish between the two tie-handling rules TP ("ties promote") and TE ("ties eliminate") in the two types of control by partition of candidates (with or without run-off) and in control by partition of voters) and, on the other hand, that the two additional control types in Section 3.3 (constructive and destructive control by adding a limited number of candidates [FHHR07b]) haven't been considered for Condorcet voting [BTT92, HHR07a].

We also study approval voting and SP-AV with respect to destructive bribery. Faliszewski et al. [FHH06a] proposed bribery as another way of influencing the outcome of elections and showed in particular that approval voting is resistant to constructive bribery. In contrast, we prove that approval voting is vulnerable to destructive bribery, even when weights and prices are assigned to the voters.

In Chapter 5, we study the second voting system, called fallback voting (FV for short), introduced by Brams and Sanver [BS06], which combines in a natural way voters' preference rankings with their approvals/disapprovals of the candidates. The main difference in the presentation of the ballots between SP-AV and FV is that in FV only the candidates a voter approves of are ranked, candidates the voters disapprove of are not.

The name "fallback" derives from the fact that during the winner determination, fallback voting successively falls back on lower-ranked approved candidates until a candidate is found, who is approved of by a strict majority (i.e., more than 50%) of the voters. Fallback voting was first mentioned by Brams and Kilgour [BK98] in the context of bargaining, not in voting.

We prove that fallback voting is, like SP-AV and majority, fully resistant to candidate control, and is fully susceptible to voter control.

The topic of Chapter 6 is motivated by a problem related to voting, namely the optimal weighted lobbying problem. Regarding the former problem, Christian et al. [CFRS06] defined its unweighted variant as follows: Given a 0-1 matrix that represents the No/Yes votes for multiple referenda in the context of direct democracy, a positive integer k, and a target vector (of the outcome of the referenda) of an external actor ("The Lobby"), is it possible for The Lobby to reach its target by changing the votes of at most k voters? They proved the optimal lobbying problem complete for the complexity class $\mathrm{W}[2]$, thus providing strong evidence that it is intractable even for small values of the parameter k. However, The Lobby might still try to find an approximate solution efficiently. We propose an efficient greedy algorithm that establishes the first approximation result for the weighted version of this problem in which each voter has a price for changing his or her 0-1 vector to The Lobby's specification. Our approximation result applies to Christian et al.'s original optimal lobbying problem (in which each voter has unit price), and also provides the first approximation result for that problem. In particular, we achieve logarithmic approximation ratios for both these problems. In a different context, this result has been independently achieved by Sandholm et al. [SSGL02]. For the sake of completeness, we will present their approach in Section 6.3.

Chapter 7's work, while not directly about elections, is motivated by models and notions from two papers that are from the study of elections, namely, the work of Homan and Hemaspaandra on greedy algorithms for Dodgson elections [HHa] and the work of Procaccia and Rosenschein on the relationship between junta distributions and manipulation of elections [PR07].

The Dodgson winner problem was shown NP-hard by Bartholdi, Tovey, and Trick [BTT89b]. Hemaspaandra, Hemaspaandra, and Rothe [HHR97] optimally improved this result by showing that the Dodgson winner problem is complete for $\mathrm{P}_{\|}^{\mathrm{NP}}$, the class of problems solvable via parallel access to NP. Since these hardness results are in the worst-case complexity model, it is natural to wonder if one at least can find a heuristic algorithm solving the problem efficiently for "most of the inputs occurring in practice." Homan and Hemaspaandra ([HHa], see also the closely related work of McCabe-Dansted, Pritchard, and Slinko [MPS]; [HHa] discusses in detail the similarities and contrasts between the two papers' work) proposed a heuristic, called Greedy-Winner, for finding Dodgson winners. They proved that if the number of voters greatly exceeds the number of candidates (which in many real-world cases is a very plausible assumption), then their heuristic is a *frequently self-knowingly correct algorithm*, a notion they introduced to formally capture a strong notion of the property of "guaranteed success frequency" [HHa]. We study this notion in relation with average-case complexity.

We also investigate Procaccia and Rosenschein's notion of deterministic heuristic polynomial time for their so-called junta distributions, a notion they introduced in their study of the "average-case complexity of manipulating elections" [PR07]. We show that under the junta definition, when stripped to its basic three properties, every NP-hard set is $\leq_m^p$-reducible to a set in deterministic heuristic polynomial time relative to some junta distribution and we also show a very broad class of sets (including many NP-complete sets) to be in deterministic heuristic polynomial time relative to some junta distribution.

Chapter 2

Preliminaries

In this chapter we will give a brief overview of computational complexity theory. For the formal definitions and specifications of the following notions, see any textbook about complexity theory (e.g., [Rot05, Pap95, HO02]).

We first establish some basic notation that is commonly used in mathematics. Let $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ the set of integers, $\mathbb{N} = \{0, 1, 2, \ldots\}$ the set of nonnegative integers, and $\mathbb{N}^+ = \{1, 2, 3, \ldots\}$ the set of positive integers. Let $\mathbb{Q}$ denote the set of rational numbers defined as the quotient of two integers, $\mathbb{Q}_0^+$ the set of nonnegative rational numbers, and $\mathbb{Q}^+$ the set of positive rational numbers. For any set S, let $\|S\|$ denote the *cardinality* of S, i.e., the number of elements in S.

2.1 The Computational Model

Fix the alphabet $\Sigma = \{0, 1\}$. Σ^* is the set of strings with finite length over Σ. For any $w \in \Sigma^*$, $|w|$ denotes the *length* of w. Let Σ^n denote the set of all length n strings in Σ^*. Any subset $L \subseteq \Sigma^*$ is called a *language* or a *problem*, the complement of L is defined by $\overline{L} = \Sigma^* - L$. $\|L\|$ denotes analogous to common sets the cardinality of L, that is, the number of strings in language L. The *characteristic function* χ_L of L tells us whether or not a string over the alphabet Σ is in the language L, i.e., $\chi_L(w) = 0$ if $w \notin L$, and $\chi_L(w) = 1$ if $w \in L$. For any $x, y \in \Sigma^*$, $x < y$ means that x *precedes* y in lexicographic order, and $x - 1$ denotes the *lexicographic predecessor* of x.

One main objective of complexity theory is to classify problems according to their computational complexity, and to determine their hardness, that is, given a language L, how hard is it for an algorithm to decide whether or not a given string $w \in \Sigma^*$ belongs to L? Before going into this, we have to clarify what an algorithm is. An *algorithm* for computing a function f is a well-defined computational procedure with a finite set of rules that provides an output $f(w)$ from an input string $w \in \Sigma^*$. An algorithm either terminates after a finite number of steps computing the desired function f on input w or rejects the input string, or never halts at all. We call an algorithm that decides whether $w \in L$ a *decision algorithm* for L. For formal definitions and many algorithmic problems see [CLRS01].

Our goal in this work is to classify the underlying problems in terms of their complexity. Complexity classes are characterized by several parameters.

First, by the underlying computational model. We will use the Turing machine as our algorithmic device. A *Turing machine* is a mathematical model with an input tape and a fixed number of work tapes which models the computation of an all-purpose computer. A Turing machine can be either used as an *acceptor*, which accepts a language L, or as a *transducer*, which computes a function. We will use in this thesis Turing machines as acceptors. For the formal definition of Turing machines, see [Rot05].

Second, after choosing the algorithmic device (in our case the Turing machine), we have to specify the way our machine accepts its input. In this work, we will distinguish between deterministic Turing machines (DTM, for short) and nondeterministic Turing machines (NTM, for short). Of course, there are many other types, such as probabilistic or alternating Turing machines, but these are not relevant to our work. The main difference between DTMs and NTMs lies in their way of computation. While an NTM can have more than one (even infinitely many) computation paths, a so-called *computation tree*, a DTM has only one computation path on a given input string, whereas every configuration other than the initial configuration is uniquely determined by its predecessor configuration.

Since there are many different aspects of how to evaluate the computation of an algorithm, such as the time or the space needed for the computation etc., the third and last parameter we have to choose for the characterization of a complexity class is the resource used. In this work we will focus on the resource time, i.e., the number of computation steps used in the algorithm. A DTM M is a *deterministic polynomial-time bound Turing machine* (DPTM for short), if there exists a fixed polynomial p, such that for each input string w, the DTM M reaches its accepting or rejecting final configuration in at most $p(|w|)$ steps. An NTM N is a *nondeterministic polynomial-time bound Turing machine* (NPTM for short), if there exists a fixed polynomial p, such that for each input string w, every computational path of N has length at most $p(|w|)$.

For a Turing machine (deterministic or nondeterministic) M, let $L(M)$ denote the *languange* accepted by M. There exists a tool which can make Turing machines more powerful, namely oracles. An *oracle Turing machine M* with oracle A (written as M^A), where $A \subseteq \Sigma^*$, is a conventional Turing machine that makes use of the information provided by the *oracle A*. M^A has an additional work tape called the *query tape* and three additional states $z_?$, z_{yes}, and z_{no}. An oracle Turing machine M^A works analogously to a conventional Turing machine until it changes to state $z_?$. At this point, M^A interrupts its computation, and if a string q is on the query tape then M^A receives from the oracle the answer "Yes" if $q \in A$, or "No" if $q \notin A$, within one step. On a "Yes" answer, M jumps into the state z_{yes} and continues its computation, and on a "No" answer, M^A changes into the state z_{no} and continues its computation. Note that the oracle gives the answer to the question "Is $q \in A$?" in only one step no matter how hard it is to decide A. In this sense, M^A on input w is the computation of M on input w, *relative to A*. Is the running time of an oracle Turing machine bounded by some polynomial, we write DPOTM in the deterministic case, and NPOTM in the nondeterministic case.

Unless stated otherwise, we will always consider *worst-case complexity* if we talk about running times, that is, we consider the maximum number of steps an algorithm makes on any length n input w. Worst-case complexity is based on the $\mathcal{O}$-notation.

Definition 2.1. *Let $g : \mathbb{N} \to \mathbb{N}$ be a function. Define the family of functions $\mathcal{O}(g)$ as*

$$\mathcal{O}(g) = \{f : \mathbb{N} \to \mathbb{N} | (\exists c > 0)(\exists n_0 \in \mathbb{N})(\forall n \geq n_0)[f(n) \leq c \cdot g(n)]\}.$$

We say that the functions in $\mathcal{O}$ grow asymptotically no faster than g.

We usually use the terms $\mathcal{O}(n)$ or $\mathcal{O}(n^3)$ instead of $\mathcal{O}(g)$ where $g(n) = n$ or where $g(n) = n^3$, respectively. Note that the $\mathcal{O}$-notation neglects constant factors and finitely many exceptions. Having some of the most basic terms and tools we need, in the next section we will introduce complexity classes and techniques relevant to this thesis.

2.2 Complexity Classes, Problems and Algorithms

Before classifying our election problems, we introduce the complexity classes that will come up later in this thesis. The two most important complexity classes are P and NP. We say that a languange L belongs to P if there exists a polynomial-time algorithm that on each input $w \in \Sigma^*$ decides whether $w \in L$ (i.e., there is a deterministic polynomial-time Turing machine that accepts L). A language L is in NP if there is a nondeterministic Turing machine that accepts L. Many natural problems (natural in the sense that these problems have already been encountered in practice) belong to NP when formalized as a *decision problem*, i.e., as a problem whose solution is either "Yes" or "No". Such a natural problem is, for example, to partition a given set of integer numbers (which sum up to an even number) into two subsets in a way, that the sum of the integers in the two subsets is the same. The formal description of this problem is:

Name: Partition.
Given: A sequence of nonnegative integers $s_1, s_2, \ldots, s_n$ such that $\sum_{i=1}^{n} s_i$ is an even number.
Question: Does there exist a subset $A \subseteq \{1, 2, \ldots, n\}$ such that $\sum_{i \in A} s_i = \sum_{i \in \{1,2,\ldots,n\}-A} s_i$?

While the problems in P are said to be easy to solve, the so-called NP-complete problems (see Definition 2.3) are considered to be "intractable", i.e., to be computationally hard unless P = NP.

Here we already face one of the most significant questions of complexity theory, namely, whether or not P equals NP? Clearly, P $\subseteq$ NP, since a DTM is a special NTM. Unfortunately, it is not known if P is a proper subset of NP or not.

To exactly classify a problem in terms of its complexity, it is not enough to give an algorithm that decides it. This gives only an upper bound. If we could compare our problem with problems whose complexity is known, for example, to show that our given problem is at least as hard as the known problem, that would help us to precisely capture the complexity of the problem. Fortunately, complexity theory has a powerful tool for such comparisons, namely, so-called reductions.

Definition 2.2. *Let A and B be two languages over the alphabet Σ. We say that A is* polynomial-time many-one reducible to *B ($A \leq_{\mathrm{m}}^{\mathrm{p}} B$) if and only if there is a polynomial-time computable function $f : \Sigma^* \to \Sigma^*$ such that*

$$w \in A \Longleftrightarrow f(w) \in B$$

holds for all $w \in \Sigma^*$.

Of course, there are also other reducibilities beside polynomial-time many-one reducibility, such as polynomial-time truth-table reducibilities, polynomial-time Turing reducibilities, strong nondeterministic reducibilities, and many others. A number of different reducibilities can be found, e.g., in the textbook of Rothe [Rot05]. Since we will use only the polynomial-time many-one reducibilities in this thesis, we will simply use the term reducibility, unless stated otherwise. Based on reducibility, we can define the notions of hardness and completeness.

Definition 2.3. *Let $\mathscr{C}$ be any complexity class, and let B be a language over Σ^*. We say that B is* $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for *$\mathscr{C}$ if and only if every language in $\mathscr{C}$ reduces to it (i.e., $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ for all $A \in \mathscr{C}$). B is called* $\leq_{\mathrm{m}}^{\mathrm{p}}$-complete *for $\mathscr{C}$ if and only if $B \in \mathscr{C}$ and B is $\leq_{\mathrm{m}}^{\mathrm{p}}$-hard for $\mathscr{C}$.*

When clear from context, we will use the terms "$\mathscr{C}$-hard" and "$\mathscr{C}$-complete" instead of "$\leq^{\mathrm{p}}_{\mathrm{m}}$-hard for $\mathscr{C}$" and "$\leq^{\mathrm{p}}_{\mathrm{m}}$-complete for $\mathscr{C}$". The first problem that was proved to be NP-complete is the so called Satisfiability problem. Before giving the formal definition of this problem, we have to make a digression to Boolean logic.

Definition 2.4. • *The two basic boolean constants are* true *and* false*, denoted by* 1 *and* 0*, respectively. Let* $x_1, x_2, \ldots$ *be* boolean variables*, i.e.,* $x_i \in \{0,1\}$*. Variables and their negations are called* literals*.*

- Boolean formulas *are inductively defined as follows:*

  1. *The boolean constants and every boolean variable is a boolean formula.*
  2. *Let F and G be two boolean formulas, then the following terms are also boolean formulas:*
     - $\neg F$ *(the* negation *of F),*
     - $F \wedge G$ *(the* conjunction *of F and G),*
     - $F \vee G$ *(the* disjunction *of F and G),*
     - $F \implies G$ *(F* implies *G, i.e.,* $F \implies G = \neg F \vee G$*), and*
     - $F \iff G$ *(F and G are* equivalent*, i.e.,* $F \iff G = (F \wedge G) \vee (\neg F \wedge \neg G)$*).*
  3. *Nothing else is a boolean formula.*

- *A* truth assignment *for a boolean formula F, with variables $x_1, x_2, \ldots, x_n$, assigns "true" or "false" to each variable $x_i \in F$ for all $1 \leq i \leq n$. A truth assignment* satisfies *F if the aggregated value of F is true.*

- *A boolean formula H is in* conjunctive normal form*, if*

$$H = \bigwedge_i \big(\bigvee_j L_{i,j}\big),$$

where $L_{i,j}$ are literals.

Boolean operations such as negation or conjunction are defined by their truth table as illustrated by Table 2.1.

The satisfiability problem is then defined as follows:

Name: Satisfiability (SAT).
Instance: A boolean formula F in conjunctive normal form.
Question: Is there a satisfying truth assignment for F?

x_1	x_2	$\neg x_1$	$x_1 \vee x_2$	$x_1 \wedge x_2$	$x_1 \Longrightarrow x_2$	$x_1 \Longleftrightarrow x_2$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Table 2.1: Truth table for the boolean operations in Definition 2.4.

The NP-completeness of SAT was shown independently by Cook [Coo71] and Levin [Lev73].

In Chapters 4 and 5, we show that special control problems are "computationally resistant to certain types of attacks". We do so by proving that these problems are NP-hard. To this end, we provide reductions from the NP-complete problems Hitting Set and Exact Cover by Three-Sets (X3C, for short). To learn more about these two problems, we refer to the textbook of Garey and Johnson [GJ79], where many standard problems are described and discussed. X3C is defined as follows.

Name: Exact Cover by Three-Sets (X3C).
Instance: A set $B = \{b_1, b_2, \ldots, b_{3m}\}$, $m \geq 1$, and a collection $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$ with $\|S_i\| = 3$ for each i.
Question: Is there a subcollection $\mathscr{S}' \subseteq \mathscr{S}$ such that every element of B occurs in exactly one set in $\mathscr{S}'$?

In Theorems 4.5 and 4.7 we will use a slightly modified version of X3C namely, with the restriction that $m > 1$. Note that this modified problem is still NP-complete.

The formal definition of our second NP-complete problem, Hitting Set is as follows.

Name: Hitting Set.
Instance: A set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$, and a positive integer $k \leq m$.
Question: Does $\mathscr{S}$ have a hitting set of size at most k, i.e., is there a set $B' \subseteq B$ with $\|B'\| \leq k$ such that for each i, $S_i \cap B' \neq \emptyset$?

Again, in Theorem 5.2 we will use a slightly modified version of Hitting Set. This time, we require that $n > 1$. The resulting problem is still NP-complete.

Most of our NP-hardness proofs in Chapter 4 are via reductions from the above defined Hitting Set problem. However, in the proof of Theorem 4.6 we will use a version of

Hitting Set with the restriction that $n(k+1)+1 \leq m-k$ is required in addition.

Name: Restricted Hitting Set.
Instance: A set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$, and a positive integer $k \leq m$ such that $n(k+1)+1 \leq m-k$.
Question: Does $\mathscr{S}$ have a hitting set of size at most k, i.e., is there a set $B' \subseteq B$ with $\|B'\| \leq k$ such that for each i, $S_i \cap B' \neq \emptyset$?

Restricted Hitting Set is also NP-complete [HHR07a].

Another NP-complete problem we will use in Chapter 6 is Set Cover which is defined as follows.

Name: Set Cover.
Instance: A set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$, and a positive integer $k \leq m$.
Question: Does $\mathscr{S}$ contain a cover for B of size k or less, i.e., is there a set $\mathscr{S}' \subseteq \mathscr{S}$ with $\|\mathscr{S}'\| \leq k$ such that every element of B belongs to at least one member of $\mathscr{S}'$?

In addition to decision problems, there are also other types of problems, such as optimization problems. In optimization problems, we want to find the ”best“ solution out of all feasible solutions. An optimization problem is either a minimization problem or a maximization problem.

Definition 2.5. *An* NP minimization problem Π *is a 3-tuple* $(I, sol(x), f)$*, where*

- *I is the set of input instances,*
- *$sol(x)$ is the set of all feasible solutions for any input $x \in I$, and*
- *f is a function that assigns a positive integer $f(x,s)$ to each solution $s \in sol(x)$. We say that $f(x,s)$ is the* quality *of solution s for instance x.*

An optimal solution *for an input instance $x \in I$ is the smallest function value of $f(x,s)$, denoted by $OPT(x)$.*

NP maximization problems can be defined analogously with the difference that the optimal solution is the maximum value of $f(x,s)$.

Set Cover in the form defined above is a decision problem, but it can be redefined as a minimization problem as well [Vaz03]:

Name: Find Minimum Set Cover.
Instance: A set $B = \{b_1, b_2, \ldots, b_m\}$, a collection $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i \subseteq B$, and a cost function $c : \mathscr{S} \to \mathbb{Q}^+$.
Question: Find a minimum cost subcollection of $\mathscr{S}$ that contains a cover for B.

Unlike decision problems, where the only two possible answers are ”Yes“ and ”No“, NP-optimization problems sometimes have an algorithm that gives a near-optimal solution in polynomial time, even if there is no fast exact solution. Such algorithms are called approximation algorithms.

Definition 2.6 ([Vaz03])**.** *Let* Π *be a minimization problem as defined in Definition 2.5 and let* $\delta : \mathbb{Z}^+ \to \mathbb{Q}^+$ *be a function with* $\delta \geq 1$. *An algorithm A is a* factor δ approximation algorithm for Π *if, for any instance* $x \in I$, *the algorithm outputs in polynomial time a feasible solution* $s \in sol(x)$ *such that* $f(x,s) \leq \delta(|x|) \cdot OPT(x)$.

For details, techniques, and further discussions about optimization problems and approximation algorithms, see the textbooks by Vazirani [Vaz03], and by Garey and Johnson [GJ76].

For the following complexity classes, we first have to define the class of complements of the sets in a complexity class $\mathscr{C}$ as $co\mathscr{C} = \{\overline{L} \mid L \in \mathscr{C}\}$. To capture the complexity of problems beyond NP, we will generalize the classes P, NP, and coNP by defining the polynomial hierarchy built upon NP. Before starting with that, let us recall the definition of an oracle Turing machine.

Definition 2.7. *Let* P^A *be the complexity class of all sets L such that there exists a DPOTM M with access to an oracle* $A \in \Sigma^*$ *with* $L = L(M^A)$. *Analogously, let* NP^A *be the complexity class of all sets L such that there exists an NPOTM N with access to an oracle* $A \in \Sigma^*$ *with* $L = L(N^A)$. *This definition can be extended to the notion of computation relative to a complexity class* $\mathscr{C}$*:*

$$\mathrm{P}^{\mathscr{C}} = \bigcup_{A \in \mathscr{C}} \mathrm{P}^A \textit{ and } \mathrm{NP}^{\mathscr{C}} = \bigcup_{A \in \mathscr{C}} \mathrm{NP}^A.$$

Now we can define the polynomial hierarchy.

Definition 2.8 ([MS72])**.** *The* polynomial hierarchy *is inductively defined as follows:*

- $\Delta_0^{\mathrm{p}} = \Sigma_0^{\mathrm{p}} = \Pi_0^{\mathrm{p}} = \mathrm{P}$,

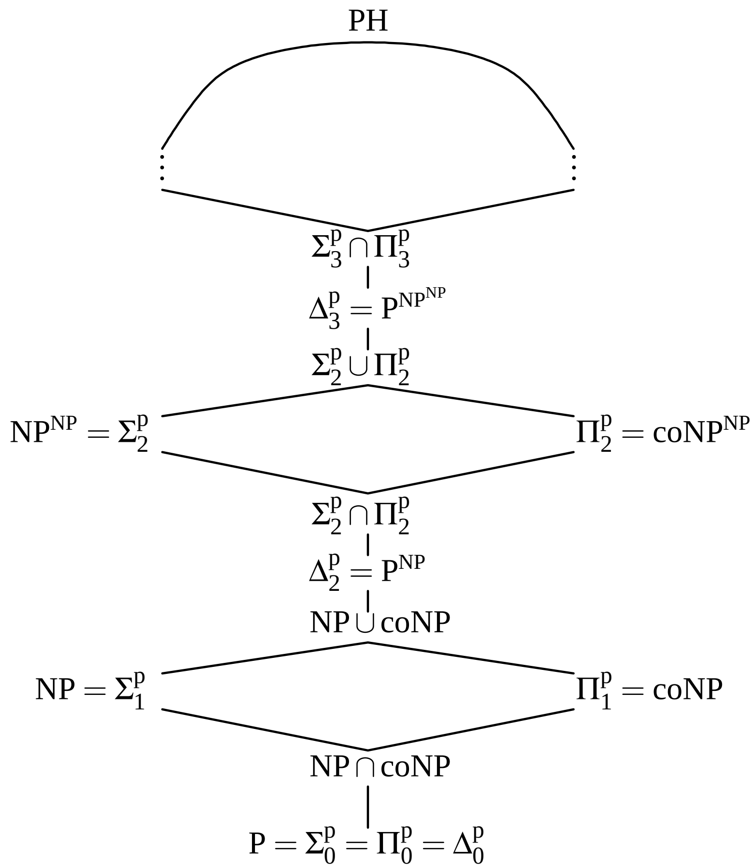


Figure 2.1: Polynomial hierarchy (the figure is due to Rothe [Rot05]).

- *For every* $i \geq 0$, $\Delta_{i+1}^p = \mathrm{P}^{\Sigma_i^p}$, $\Sigma_{i+1}^p = \mathrm{NP}^{\Sigma_i^p}$, *and* $\Pi_{i+1}^p = \mathrm{coNP}^{\Sigma_i^p}$,
- $\mathrm{PH} = \bigcup_{i \geq 0} \Sigma_i^p$.

Note that $\Delta_1^p = \mathrm{P}$, $\Sigma_1^p = \mathrm{NP}$, and $\Pi_1^p = \mathrm{coNP}$. Figure 2.2, which is taken from [Rot05], illustrates the inclusions in the polynomial hierarchy.

Hemaspaandra, Hemaspaandra, and Rothe proved that the Dodgson winner problem is $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete [HHR97]. The complexity class $\mathrm{P}_{\|}^{\mathrm{NP}}$ can be described as follows: $\mathrm{P}_{\|}^{\mathrm{NP}}$ is the class that contains the sets that can be accepted by a DPOTM M accessing an NP oracle with the restriction that the machine M first has to compute the list of all query strings, and pass them over to the oracle at once, the oracle gives the answers in one step for all query strings. This type of oracle access is called *parallel oracle access*. There are several other characterizations of $\mathrm{P}_{\|}^{\mathrm{NP}}$, for instance, L^{NP} or $\mathrm{P}^{\mathrm{NP}[\log]}$ introduced by Papadimitiriou and Zachos [PZ83]. $\mathrm{P}^{\mathrm{NP}[\log]} = \mathrm{P}_{\|}^{\mathrm{NP}}$ has been proven by Hemaspaandra [Hem87]. $\mathrm{P}_{\|}^{\mathrm{NP}}$ is between $\mathrm{NP} \cup \mathrm{coNP}$ and P^{NP} in the polynomial hierarchy. There are several other natural problems which are $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete, e.g., Odd Minimum Vertex Cover (see [Wag87]), YoungWinner (see [RSV02, RSV03]), and YoungRanking (see [RSV02, RSV03]).

2.3 Basic Facts from Parameterized Complexity Theory

Parameterized complexity is a new field in computational complexity theory introduced by Downey and Fellows in the late 1980s. The main goal of parameterized complexity is to analyze the behaviour of computationally intractable problems. For a detailed representation, see, e.g., the textbooks by Downey and Fellows [DF99] and Flum and Grohe [FG06] and the surveys by Lindner and Rothe [LR08] and Cesati [Ces03].

In parameterized complexity, a *parameterized language* is a subset $L \subseteq \Sigma^* \times \Sigma^*$. For each pair of strings $(x,y) \in \Sigma^* \times \Sigma^*$, we say that y is the *parameter*. We consider only positive integers as parameters in this thesis, thus we can define the domain of the language L as $\Sigma^* \times \mathbb{N}^+$. Then, a *parameterized decision problem* takes as an input a pair $(x,y) \in \Sigma^* \times \mathbb{N}^+$ and outputs "Yes" if $(x,y) \in L$, and "No" if $(x,y) \notin L$.

Just like in classical complexity theory, in parameterized complexity theory we can also classify the problems according to tractability and intractability. We start with the formal definition of fixed-parameter tractability.

Definition 2.9. *Let $L \subseteq \Sigma^* \times \mathbb{N}^+$. L is said to be* fixed-parameter tractable *if there exists an algorithm with running time $f(k)n^\alpha$ that decides on input $(x,y) \in \Sigma^* \times \mathbb{N}^+$ whether $(x,y) \in L$, where $n = |x|$, $k = |y|$ is the parameter, α is a constant independent of k, and f is some computable function.*

We also say that the fixed-parameter tractable problems belong to the class FPT (Fixed Parameter Tractable), which is the parameterized analog of P. The basic idea behind fixed-parameter tractability is to split the input into two parts, one easy to solve part (this would be x in our definition), where we get the polynomial n^α running time and a hard part (this is then the parameter y), which we "turn over to the devil" (as suggested by Downey and Fellows [DF99, p. 8]), which accounts for the function $f(k)$. That is, if we fix the parameter k, it is easy to determine whether $(x,y) \in L$.

Before we start with fixed-parameter intractability, we have to establish the parameterized analog of the polynomial-time many-one reducibility, the so called parameterized reducibility.

Definition 2.10. *Let A and B be two parameterized problems with $A, B \subseteq \Sigma^* \times \mathbb{N}^+$, where Σ is a fixed alphabet. We say that A* is fpt many-one reducible to *B if there is an algorithm Ψ that computes from a given instance $(x,k) \in A$ an instance $(x',k') \in B$, such that:*

(1) *For all* $(x,k) \in \Sigma^* \times \mathbb{N}^+$, $(x,k) \in A \Longleftrightarrow (x',k') \in B$ *holds.*

(2) *There exists a function* f *such that* $k' = f(k)$.

(3) *The running time of* Ψ *is bounded by* $g(k) \cdot |x|^\alpha$, *for some arbitrary function* g *and constant* α.

Turning now to fixed-parameter intractability, we say that a parameterized problem is *fixed-parameter intractable* if there is no FPT-algorithm that solves the problem. Akin to polynomial hierarchy in classical complexity theory, there is also a hierarchy in parameterized complexity theory to classify fixed-parameter intractable problems, the so-called *W-hierarchy*:

$$FPT = W[0] \subseteq W[1] \subseteq W[2] \subseteq \cdots .$$

Instead of giving the very complex formal definition of the W-hierarchy and its members, we only describe intuitively the two classes relevant to this thesis, namely W[1] and W[2]. To do so, we first define the Short Nondeterministic Turing Machine Computation problem.

Name: Short Nondeterministic Turing Machine Computation.
Instance: A single-tape NTM M, an input string x, and a positive integer k.
Parameter: k.
Question: Is there a computational path of M on input x, such that M reaches a final accepting state in at most k steps?

This problem was proven to be W[1]-complete by Cai et al. [CCDF97]. This result can be seen as the analogon of Cook's theorem for parameterized complexity. Cesati [Ces03, p. 658] suggested the following characterization of W[1]-membership:

> ”Turing way to W[1]-membership: To show that a parameterized problem L belongs to W[1], devise a parameterized reduction from L to the Short Nondeterministic Turing Machine Computation problem.“

The class W[2] can be characterized analogously with the W[2]-complete Short Multi-Tape Nondeterministic Turing Machine Computation problem.

Name: Short Multi-Tape Nondeterministic Turing Machine Computation.
Instance: A multi-tape NTM M, an input string x, and a positive integer k.
Parameter: k.
Question: Is there a computational path of M on input x, such that M reaches a final accepting state in at most k steps?

Cesati's description for W[2] is then [Ces03, p. 663]:

> ”Turing way to W[2]-membership: To show that a parameterized problem L belongs to W[2], devise a parameterized reduction from L to the Short Multi-Tape Nondeterministic Turing Machine Computation problem.“

Although both W[1]-complete and W[2]-complete problems are fixed parameter intractable, somehow the W[1]-complete problems seem to be easier. It is unlikely, that a problem solved by a short multi-tape NTM in k steps can always be solved by a short single-tape NTM in k steps as well. There are many natural W[1]-complete and W[2]-complete problems. The parameterized versions of Independent Set, Set Packing, and Clique are, for instance, W[1]-complete, and the parameterized versions of Dominating Set, Hitting Set, and Set Cover, on the other hand, are W[2]-complete.

2.4 Graphs

Last but not least, in this chapter, we present some basic notions from graph theory. The history of graph theory goes back into the year 1736, when the Swiss mathematician and physicist Leonhard Euler (1707-1782) published the first paper in this field. The problem he solved is known as the Seven Bridges of Königsberg.

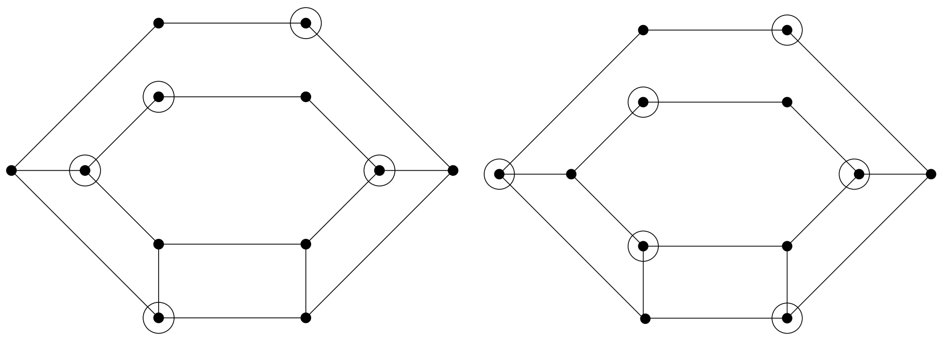
Many problems proven to be $W[2]$-complete are derived from problems concerning graphs. In the following we present the definition of an undirected graph along with two related graph problems.

Definition 2.11. *An* undirected graph *G is a pair $G = (V,E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is a finite (nonempty) set of* vertices *and $E = \{\{v_i, v_j\} \mid 1 \leq i < j \leq n\}$ is the set of* edges. *Any two vertices connected by an edge are called* adjacent. *The vertices adjacent to a vertex v are called the* neighbors *of v, and the set of all neighbors of v is denoted by $N[v]$, i.e., $N[v] = \{u \in V \mid \{u,v\} \in E\}$. The* degree of vertex v in graph G *is the number of neighbors of v, denoted by $deg_G(v)$, i.e., $deg_G(v) = \|N[v]\|$. The* minimum degree of G *is defined as $mindeg(G) = min_{v \in V}(deg_G(v))$.*

In this thesis we only consider undirected graphs without multiple edges or self-loops, i.e., edges of the form $\{v,v\}$ are not allowed. The two graph problems related to this work are based on the notion of dominating sets, which we will define next.

Definition 2.12. *Let $G = (V,E)$ be a graph, where V is the set of vertices and E is the set of edges. A subset $V' \subseteq V$ is called a* dominating set *if for all vertices $v \in V$, either $v \in V'$ or there exists at least one vertex $u \in V'$ such that $\{u,v\} \in E$. If V' is a dominating set and $\|V'\| = k$ then we will say that V' is a* k-dominating set. *If in addition there are no adjacent vertices in V', we will say that V' is an* independent k-dominating set of G.

Figure 2.2 shows a 5-dominating set and an independent 6-dominating set for the same graph. The encircled vertices correspond to a 5-dominating set and an independent 6-dominating set in figure (a) and (b), respectively.



(a) 5-dominating set (b) Independent 6-dominating set

Figure 2.2: Examples for k-dominating set and independent k-dominating set.

The first parameterized graph problem we consider is k-Dominating Set, which was proved to be W[2]-complete by Downey and Fellows [DF99].

Name: k-Dominating Set.
Instance: A graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, and a positive integer k.
Parameter: k.
Question: Does G have a k-dominating set?

The second parameterized graph problem is Independent k-Dominating Set, which was also shown by Downey and Fellows to be W[2]-complete [DF95].

Name: Independent k-Dominating Set.
Instance: A graph $G = (V, E)$, where V is the set of vertices and E is the set of edges, and a positive integer k.
Parameter: k.
Question: Does G have an independent k-dominating set?

For more details and results about graphs we recommend the textbooks of Harary [Har69] and Diestel [Die05].

Chapter 3

Elections

As most commonly in literature, votes will here be represented nonsuccinctly: one ballot per voter. Note that some papers (e.g., [FHH06b, FHHR07a, FHHR08a]) also consider succinct input representations for elections where multiplicities of votes are given in binary.

Voting systems [1] are one of the most popular and rudimentary ways of reaching common decisions. Voting systems aggregate individuals' preferences into a collective decision. The outcome of the election always depends on the aggregation rules used in the voting system. Lots of different voting systems were introduced in the literature, for a brief overview we point the reader to the work of Brams and Fishburn [BF02]. In this chapter we define the voting systems and the aspects under which they shall be examined.

3.1 Voting Systems

In general an election is a pair $E = (C,V)$, where C is a finite set of candidates and V is a finite collection of voters who express their preferences over all candidates in C. How the voter preferences are represented depends on the voting system used. We distinguish between two models. As in most papers on electoral control (exceptions are, e.g., [PRZ07, FHHR08a]), we define the control problems in the unique-winner model. In this model, via the control action, the chair seeks to either make a designated candidate the unique winner (in the constructive case) or to prevent a designated candidate from being a unique winner (in the destructive case). As we have seen in Chapter 1, different aggregation rules can yield different winners for the same election. Voting systems have two main

[1] Recall, that an election is not a voting system. An election is an event, where individuals can express their preferences, whereas a voting system is a procedure to aggregate the different preferences in an election to yield a common decision.

characteristics. First, the form how ballots are represented. They can be represented as approval/disapproval vectors like in approval voting, as rankings over the candidate set just like in Borda count, or as a preference table in an irrational-voter model. Second, the way how the voting rules determine the winner of the election. In the following we describe two families of voting systems that are important to this thesis.

Preference-based systems: Let $E = (C,V)$ be an election with candidate set $C = \{c_1, c_2, \dots, c_m\}$. Each voter has to specify a weak preference order $c_{j_1} \succcurlyeq c_{j_2} \succcurlyeq \dots \succcurlyeq c_{j_m}$ over all candidates, where $\{j_1, j_2, \dots, j_m\} = \{1, 2, \dots, m\}$. This ranking is a linear ordering—with or without ties—of all candidates, where the leftmost candidate is the most preferred and the rightmost candidate is the most despised one.

Scoring protocols: Let $E = (C,V)$ be an election with candidate set $C = \{c_1, c_2, \dots, c_m\}$. Given a scoring vector $(\alpha_1, \alpha_2, \dots, \alpha_m)$ of nonnegative integers such that $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m$. Each voter has to rank his or her candidates, and gives α_j points to the candidate he or she ranks on the jth place.

As one can see, scoring protocols are also preference-based systems. Of course, there are other families of voting systems beyond the two introduced here, e.g., irrational voting systems as mentioned above. In the rational-voter model, it is required from each voter to specify a weak preference order over the set of candidates. In the irrational-voter model, each voter specifies his or her preferences via preference table. Unlike in the rational-voter model, in the irrational-voter model the voters' preferences don't necesseraly have to be transitive. However, irrational voting systems are beyond the scope of this thesis, we are well aware of the fact that the behavior of voters doesn't always follow a rational pattern. For a detailled discussion and more information on the irrational-voter model, see the papers of Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe [FHHR07a, FHHR08a, FHHR08c]. Let us now describe the voting systems used in this work.

Approval voting: Every voter draws a line between his or her acceptable and inacceptable candidates by specifying a 0-1 approval vector, where 0 represents disapproval (a "No"-vote) and 1 represents approval (a "Yes"-vote), yet does not rank them. The winners are those candidates with maximum number of approvals. Just as described in Chapter 1, approval voting can be seen as a preference-based system, where all the approved candidates tie for first place, and all the disapproved candidates tie for last place. It should be noted that approval voting is not a scoring protocol, since each voter can approve of a different number of candidates. The scoring protocol version of approval voting would be k-approval voting, where each voter has to approve of exactly k candidates, i.e., k-approval is the scoring protocol with scoring vector $(\alpha_1, \alpha_2, \ldots, \alpha_m)$ with $\alpha_1 = \cdots = \alpha_k = 1$ and $\alpha_{k+1} = \cdots = \alpha_m = 0$ where $k \leq m$ is required.

SP-AV: Sincere-strategy preference-based approval voting is a hybrid system of preference-based and approval voting. Each voter has to specify a tie-free linear ordering over all candidates. This is usually a left-to-right ranking (separated by a space) of all candidates (e.g., $a\ b\ c$), with the leftmost candidate being the most preferred one. The border between the approved and disapproved candidates is indicated by inserting a straight line into such a ranking, where all candidates left of this line are approved and all candidates right of this line are disapproved (e.g., "$a \mid b\ c$" means that a is approved, while both b and c are disapproved). There are also two important properties to keep in mind. First, we require admissibility, which means that the highest ranked candidate has to be approved of and the least preferred candidate has to be disapproved of by each voter. Second, each voter's ballot has to be sincere, that is each candidate left of the approval line has to be approved of and every candidate right of the approval line has to be disapproved of. The winners of an election are again those candidates with the highest number of approvals. We will give the formal definition of SP-AV and a detailed discussion of these two requirements in Chapter 4. SP-AV is, just as approval voting, a preference-based system yet not a scoring protocol.

Fallback voting: Each voter has to decide for each candidate if he or she wants to approve of, or disapprove of the candidate. Furthermore, each voter has to give a strict preference ranking (i.e., irreflexive, antisymmetric, transitive and complete) for the candidates he or she approves of. The aggregation procedure takes place on several levels. On the first level, only the highest ranked candidates are considered. Every candidate with a strict majority (i.e., more than half of the voters approved of the candidate) is a winner. If there is no such candidate, then we move to the second level. Now we consider the two highest ranked candidates in each ballot. If there is a candidate with a strict majority, he or she is the winner. If there are more than one candidates with strict majority then each candidate with the highest number of approvals is a winner. Otherwise, we keep moving to the next levels step by step, until there is a candidate with a strict majority. If no such candidate was found, each candidate with the highest number of approvals is a winner of the election. We will give the formal definition of Fallback voting in Chapter 5. Note, that fallback voting is also a preference-based system, even if the voters only have to rank the approved candidates, since the candidates a voter disapproved of can be considered as candidates tied for last place. Fallback voting is not a scoring protocol.

Plurality: Each voter has to give a strict preference ranking over all candidates, the candidate with the most first places is the winner. In contrast with the aforementioned voting systems, plurality is a scoring protocol with scoring vector $(\alpha_1, \alpha_2, \ldots, \alpha_m)$, where $\alpha_1 = 1$ and $\alpha_2 = \alpha_3 = \cdots = \alpha_m = 0$.

Condorcet: Each voter has to specify a tie-free linear ordering over all candidates. The winner of the election is the candidate who wins by a strict majority of votes against all other candidates in a head-to-head contest. Note, that a Condorcet winner does not always exist, but if there is one, then that candidate is the unique winner. Even though we only allow rational voter preferences, as we could see in Example 1.1, the resulting societal preference can indeed be irrational!

Dodgson: Each voter has to specify a tie-free linear ordering over all candidates. If there exists a Condorcet winner then in that case that candidate is also the Dodgson winner. Otherwise the candidate who is "closest" to being a Condorcet winner is a winner of the election. To define closeness, Dodgson first assigned a score to each candidate, the so-called DodgsonScore(C, c, V) of candidate c in election $E = (C, V)$, defined as the smallest number of sequential swaps of adjacent candidates in the voters' preference lists which makes candidate c a Condorcet winner of the election. The candidate with the smallest Dodgson score is the winner of the election, namely the *Dodgson winner*.

A voting system is useful in practice only if its winner determination is easy, i.e., if the winners of the election can be found within polynomial-time. Except for Dodgson's system, all the above mentioned voting systems have an easy winner problem. The Dodgson winner problem was shown $\mathrm{P}_{\|}^{\mathrm{NP}}$-complete by Hemaspaandra, Hemaspaandra, and Rothe [HHR97].

For some of the proofs discussed in the following two chapters, two properties of voting systems will be defined. First, the notion of a voiced voting system will be introduced.

Definition 3.1. *Let $\mathscr{E}$ be any voting system. $\mathscr{E}$ is said to be* voiced *if in every one-candidate election, this candidate wins.*

And, second, we need the unique version of the *Weak Axiom of Revealed Preference*, denoted by Unique-WARP. It says that, if a candidate c is the unique winner in a set C of candidates, then c is the unique winner in every subset of C that includes c.

3.2 Bribery and Manipulation

There are three main types of how to affect the outcome of an election, namely, procedural control, manipulation, and bribery. In *control*, an external agent—usually called the chair—seeks to change the outcome of the election by modifying the structure of the election via adding/deleting/partitioning either candidates or voters. In *manipulation*, a coalition of evil voters tries to influence the result of the election by voting strategically (i.e., they might not express their real preferences if it helps to reach their desired outcome). In *bribery*, an external actor—the briber—tries to reach his or her desired outcome by changing some voters' votes. In this section, we introduce manipulation and bribery. Control shall be considered in the next section.

Faliszewski et al. [FHH06a] introduced the problem of bribery for election systems. There are many different settings for bribery. The simplest one is when the briber tries to influence as few voters as possible. In this case, there is an integer k specified in the problem description, which is the limit of the number of votes allowed to be altered.

We now formally define our bribery and manipulation problems, where each problem is defined somewhat different than so far, by stating the problem instance together with two questions, one for the constructive and one for the destructive case.

Let $\mathscr{E}$ be any voting system. Bribery is defined as follows.

Name: $\mathscr{E}$-bribery.

Instance: An election $E = (C,V)$, where C is the candidate set and V is the collection of voters specified via their preference lists over all candidates, a distinguished candidate $c \in C$ and a nonnegative integer k.

Question (constructive): Is it possible to change at most k votes in V such that c is the unique winner of the resulting election (C,V')?

Question (destructive): Is it possible to change at most k votes in V such that c is not a unique winner of the resulting election (C,V')?

Talking about bribery without bringing money into play is, however, not really drawn from life. It is much more likely that each voter has his or her individual price for changing his or her preference. In such a scenario, the prices of the voters would be a part of the problem instance.

Name: $\mathscr{E}$-\$bribery.

Instance: An election $E = (C,V)$, where C is the candidate set and $V = \{v_1, v_2, \dots, v_m\}$ is the collection of voters specified via their preference lists over all candidates, their prices $(p_1, p_2, \dots, p_m)$, a distinguished candidate c and a nonnegative integer B (the briber's budget).

Question (constructive): Is there a subset $V' \subseteq V$ such that $\sum_{v_i \in V'} p_i \leq B$ and bribing these voters would make c the unique winner of the election?

Question (destructive): Is there a subset $V' \subseteq V$ such that $\sum_{v_i \in V'} p_i \leq B$ and bribing these voters would prevent c from being a unique winner of the election?

It is also possible to assign a weight to each voter, or to make special restrictions how the briber should bribe a voter. For all these special types of bribery and many interesting facts about them we point the reader to the papers [Fal08, FHH06b, FHHR07a].

Manipulation is not much different from bribery. Let $\mathscr{E}$ be again a voting system.

Name: $\mathscr{E}$-Manipulation.
Instance: An election (C,V), a designated candidate $c \in C$, and a collection $V' \subseteq V$ of manipulators.
Question (constructive): Is it possible to cast the votes of V' such that c is the unique winner of the election (C,V) under the election system $\mathscr{E}$?
Question (destructive): Is it possible to cast the votes of V' such that c is not a unique winner of the election (C,V) under the election system $\mathscr{E}$?

The difference between bribery and manipulation lies in the collection of voters whose votes can be changed. In bribery the briber can freely choose voters he or she wants to influence, in manipulation the collection of voters to be influenced is predefined. For further information on manipulation, we refer to [FHHR].

The problem $\mathscr{E}$-Weighted-Manipulation can be defined again analogously with the difference that there is a weight attributed to each vote.

3.3 Control

In this section we introduce the 22 control actions we will work with. The control problems considered here were introduced by Bartholdi, Tovey, and Trick [BTT92] for constructive control and by Hemaspaandra, Hemaspaandra, and Rothe [HHR07a] for destructive control. In constructive control scenarios, the chair's goal is to make a favorite candidate the unique winner, in destructive control scenarios on the other hand, the chair tries to prevent a despised candidate from being a unique winner. As is common, the chair is assumed to have complete knowledge of the voters' preference rankings and approval strategies (see [HHR07a] for a detailed discussion of this assumption).

We now formally define our control problems.

Control by Adding Candidates

In this control scenario, the chair seeks to reach his or her goal by adding to the election, which originally involves only "qualified" candidates, some new candidates who are chosen from a given pool of spoiler candidates.

In their study of control for approval voting, Hemaspaandra, Hemaspaandra, and Rothe [HHR07a] took into account only the case of adding an *unlimited* number of spoiler candidates (which is the original variant of this problem as defined by Bartholdi, Tovey, and Trick [BTT92]). We consider the same variant of this problem here to make our results comparable with those established in [HHR07a], but for completeness we, in addition, consider the case of adding a *limited* number of spoiler candidates, where the prespecified limit is part of the problem instance. This variant of this problem was introduced by Faliszewski et al. [FHHR07a, FHHR08a] in analogy with the definitions of control by deleting candidates and of control by adding or deleting voters. They showed that, for the election system Copeland$^{\alpha}$ they investigated, the complexity of these two problems can change drastically depending on the parameter α, see [FHHR08a].

We first define the unlimited variant of control by adding candidates.

Name: Control by Adding an Unlimited Number of Candidates.
Instance: An election $(C \cup D, V)$ and a designated candidate $c \in C$, where the set C of qualified candidates and the set D of spoiler candidates are disjoint.
Question (constructive): Is it possible to choose a subset $D' \subseteq D$ such that c is the unique winner of election $(C \cup D', V)$?
Question (destructive): Is it possible to choose a subset $D' \subseteq D$ such that c is not a unique winner of election $(C \cup D', V)$?

This is a quite natural control action. Imagine a fictitious presidential election, where the candidate with the most supporters is the winner, and a few months before the election there are two leading big parties whose candidates are candidate a and candidate b. We assume that a is expected to be slightly ahead of b in the polls. Now, if the chair doesn't want candidate a to win, he or she could introduce a third candidate, say c, who has a manifesto similar to that of candidate a. It is most likely that some of a's supporters would rather prefer to support candidate c. These very same supporters would now make their second favourite candidate (candidate a) lose the election against candidate b.

The problem Control by Adding a Limited Number of Candidates is defined analogously, with the only difference being that the chair seeks to reach his or her goal by adding at most ℓ spoiler candidates, where ℓ is part of the problem instance.

Control by Deleting Candidates

In this control scenario, the chair seeks to reach his or her goal by deleting (up to a given number of) candidates.

Name: Control by Deleting Candidates.
Instance: An election (C,V), a designated candidate $c \in C$, and a nonnegative integer ℓ.
Question (constructive): Is it possible to delete up to ℓ candidates from C such that c is the unique winner of the resulting election?
Question (destructive): Is it possible to delete up to ℓ candidates (other than c) from C such that c is not a unique winner of the resulting election?

This is again a natural control action. Let us recall our little story from the adding candidates case. At the point, where all three candidates run for president, candidate b is in the most promising position. But, if candidate c backed out because he or she knows that he or she has no chance to win the election, and that would result in a fiasco of candidate a as well, candidate a would have again the best chances to win the election.

Control by Partition and Run-Off Partition of Candidates

There are two partition-of-candidates control scenarios. In both scenarios, the chair seeks to reach his or her goal by partitioning the candidate set C into two subsets, C_1 and C_2, after which the election is conducted in two stages. In control by partition of candidates, the election's first stage is held within only one group, say C_1, and this group's winners that survive the tie-handling rule used (see the next paragraph) run against all members of C_2 in the second and final stage. In control by run-off partition of candidates, the election's first stage is held separately within both groups, C_1 and C_2, and the winners of both subelections that survive the tie-handling rule used run against each other in the second and final stage.

We use the two tie-handling rules proposed by Hemaspaandra, Hemaspaandra, and Rothe [HHR07a]: ties-promote (TP) and ties-eliminate (TE). In the TP model, all the first-stage winners of a subelection, (C_1,V) or (C_2,V), are promoted to the final round. In the TE model, a first-stage winner of a subelection, (C_1,V) or (C_2,V), is promoted to the final round exactly if this person is that subelection's unique winner.

Note that partitioning the candidate set C into C_1 and C_2 is, in some way, similar to deleting C_2 from C to obtain subelection (C_1,V) and to deleting C_1 from C to obtain subelection (C_2,V). Also, depending on the tie-handling rule used, the final stage of the election may have a reduced number of candidates.

Name: Control by Partition of Candidates.
Instance: An election (C,V) and a designated candidate $c \in C$.
Question (constructive): Is it possible to partition C into C_1 and C_2 such that c is the unique winner of the final stage of the two-stage election in which the winners of subelection (C_1,V) that survive the tie-handling rule used run against all candidates in C_2 (with respect to the votes in V)?
Question (destructive): Is it possible to partition C into C_1 and C_2 such that c is not a unique winner of the final stage of the two-stage election in which the winners of subelection (C_1,V) that survive the tie-handling rule used run against all candidates in C_2 (with respect to the votes in V)?

Control by partition of candidates is in use for example at the Eurovision Song Contest Finals, where some artists are directly qualified for the final (usually the artists representing countries who are the Song Contest's biggest financial supporters), whereas numerous other candidates have to participate in a qualifying round first.

Next, we define control by run-off partition of candidates.

Name: Control by Run-Off Partition of Candidates.
Instance: An election (C,V) and a designated candidate $c \in C$.
Question (constructive): Is it possible to partition C into C_1 and C_2 such that c is the unique winner of the final stage of the two-stage election in which the winners of subelection (C_1,V) that survive the tie-handling rule used run (with respect to the votes in V) against the winners of subelection (C_2,V) that survive the tie-handling rule used?
Question (destructive): Is it possible to partition C into C_1 and C_2 such that c is not a unique winner of the final stage of the two-stage election in which the winners of subelection (C_1,V) that survive the tie-handling rule used run (with respect to the votes in V) against the winners of subelection (C_2,V) that survive the tie-handling rule used?

Let us examine the example of the Eurovision Song Contest Finals again, in which every European nation organises a national voting first to determine their representative who is then entitled to participate in the European Song Contest Finals. Admittedly this is a poor comparison in some sence, since in this example there are not only two subelections but many more. Nevertheless it is appropriate enough to represent how this control action works.

One could ask now, what is the computational difference between the two tie-handling rules? Well, quite a lot. Later in the results, we will show how significant differences between the two tie-handling rules can be.

Control by Adding Voters

In this control scenario, the chair seeks to reach his or her goal by introducing new voters into a given election. These additional voters are chosen from a given pool of voters whose preferences and approval strategies over the candidates from the original election are known. Again, the number of voters that can be added is prespecified.

Name: Control by Adding Voters.
Instance: An election (C,V), a collection W of additional voters with known preferences over C, a designated candidate $c \in C$, and a nonnegative integer ℓ.
Question (constructive): Is it possible to choose a subset $W' \subseteq W$ with $\|W'\| \leq \ell$ such that c is the unique winner of election $(C, V \cup W')$?
Question (destructive): Is it possible to choose a subset $W' \subseteq W$ with $\|W'\| \leq \ell$ such that c is not a unique winner of election $(C, V \cup W')$?

A good example for control by adding voters are local elections in Germany. A few years ago only German citizens with registered domicile in a given city were allowed to vote in that city for mayor. Nowadays, everyone (German or foreigner) is allowed to participate in the local elections with the only restriction that they have a registered domicile in that city.

Control by Deleting Voters

The chair here seeks to reach his or her goal by suppressing (up to a prespecified number of) voters.

Name: Control by Deleting Voters.
Instance: An election (C,V), a designated candidate $c \in C$, and a nonnegative integer ℓ.
Question (constructive): Is it possible to delete up to ℓ voters from V such that c is the unique winner of the resulting election?
Question (destructive): Is it possible to delete up to ℓ voters from V such that c is not a unique winner of the resulting election?

This control action is broadly used. In the United States presidential election in 2000, for instance, 58.000 names were erased off the voters list by Governor Jeb Bush and Secretary Harris of Florida, with the explanation that these voters were people with criminal records. The winner of the election was George W. Bush, who happens to be Jeb Bush's older brother. Quite a coincidence, isn't it?

Control by Partition of Voters

In this scenario, the election is conducted in two stages again, and the chair now seeks to reach his or her goal by partitioning the voters V into two subcommittees, V_1 and V_2. In the first stage, the subelections (C,V_1) and (C,V_2) are held separately in parallel, and the winners of each subelection who survive the tie-handling rule used, move forward to the second and final stage in which they compete against each other.

Name: Control by Partition of Voters.
Instance: An election (C,V) and a designated candidate $c \in C$.
Question (constructive): Is it possible to partition V into V_1 and V_2 such that c is the unique winner of the final stage of the two-stage election in which the winners of subelection (C,V_1) that survive the tie-handling rule used run (with respect to the votes in V) against the winners of subelection (C,V_2) that survive the tie-handling rule used?
Question (destructive): Is it possible to partition V into V_1 and V_2 such that c is not a unique winner of the final stage of the two-stage election in which the winners of subelection (C,V_1) that survive the tie-handling rule used run (with respect to the votes in V) against the winners of subelection (C,V_2) that survive the tie-handling rule used?

A good example for control by partition of voters is again the United States presidential election in 2000, where George W. Bush won against Al Gore, even though Al Gore had a total of approximately 500.000 more votes than George W. Bush. What a misfortunate situation for Al Gore since the president of the United States is not directly elected by the people, but elected by the electors of the Electoral College, where different states have different voting powers. Just as in the case of control by partition of candidates, here we have again more than two subelections, which doesn't really equal our definition of control by partition of voters. Although it illustrates this control scenario.

Immunity, Susceptibility, Vulnerability, and Resistance

The following notions—which are taken from Bartholdi, Tovey, and Trick [BTT92]—will be central to our complexity analysis of the control problems.

Definition 3.2. *Let $\mathscr{E}$ be an election system and let Φ be some given type of control.*

1. *$\mathscr{E}$ is said to be* immune to Φ-control *if*

   (a) *Φ is a constructive control type and it is never possible for the chair to turn a designated candidate from being not a unique winner into being the unique winner via exerting Φ-control, or*

(b) *Φ is a destructive control type and it is never possible for the chair to turn a designated candidate from being the unique winner into being not a unique winner via exerting Φ-control.*

2. *$\mathcal{E}$ is said to be* susceptible to Φ-control *if it is not immune to Φ-control.*

3. *$\mathcal{E}$ is said to be* vulnerable to Φ-control *if $\mathcal{E}$ is susceptible to Φ-control and the control problem associated with Φ is solvable in polynomial time.*

4. *$\mathcal{E}$ is said to be* resistant to Φ-control *if $\mathcal{E}$ is susceptible to Φ-control and the control problem associated with Φ is* NP*-hard.*

Immunity, susceptibility, resistance, and vulnerability can be defined analogously for bribery and manipulation as well. By definition, all resistance and vulnerability results in particular require susceptibility. To avoid a tedious proof covering each of the 22 types of control separately, we will use the general susceptibility results and links between susceptibility cases established by Hemaspaandra, Hemaspaandra, and Rothe [HHR07a].[2]

Theorem 3.1 ([HHR07a])**.** 1. *A voting system is susceptible to constructive control by adding candidates if and only if it is susceptible to destructive control by deleting candidates.*

2. *A voting system is susceptible to constructive control by deleting candidates if and only if it is susceptible to destructive control by adding candidates.*

3. *A voting system is susceptible to constructive control by adding voters if and only if it is susceptible to destructive control by deleting voters.*

4. *A voting system is susceptible to constructive control by deleting voters if and only if it is susceptible to destructive control by adding voters.*

Since the above statements are easy to see, we won't present their proofs here. The proofs of the following two theorems are not trivial, so we will specify them.

Theorem 3.2 ([HHR07a])**.** 1. *If a voting system is susceptible to constructive control by partition of voters (in model TE or TP), then it is susceptible to constructive control by deleting candidates.*

2. *If a voting system is susceptible to constructive control by partition or run-off partition of candidates (in model TE or TP), then it is susceptible to constructive control by deleting candidates.*

[2]Although [HHR07a] does not consider the case of control by adding a limited number of candidates explicitly, it is immediate that all proofs for the "unlimited" case in [HHR07a] work also for this "limited" case.

3. *If a voting system is susceptible to constructive control by partition of voters in model TE, then it is susceptible to constructive control by deleting voters.*

4. *If a voting system is susceptible to destructive control by partition or run-off partition of candidates (in model TE or TP), then it is susceptible to destructive control by deleting candidates.*

Proof. The proof is due to [HHR07a]. We start with part 1. Let (C,V) be an election, $c \in C$ a candidate who is not the unique winner of the election under voting system $\mathscr{E}$. Suppose that voting system $\mathscr{E}$ is susceptible to constructive control by partition of voters (in model TE or TP), i.e., c can be made the unique winner of the election by exerting control by partition of voters (in model TE or TP). Let $C' \subset C$ be the set of candidates who attends in the final stage of the election. Since c is the winner of the final stage, i.e., c is the unique winner of the election (C',V), deleting the subset $C-C'$ of candidates makes c the unique winner of the resulting election. Note that $C' \neq C$, since in that case each candidate would participate in the final stage, thus c couldn't win the election via control by partitioning of voters. Thus, the voting system $\mathscr{E}$ is susceptible to constructive control by deleting candidates.

The same argumentation holds also for part 2.

Let us turn to part 3. Let (C,V) be an election, $c \in C$ a candidate who is not the unique winner of the election under voting system $\mathscr{E}$. Suppose that voting system $\mathscr{E}$ is susceptible to constructive control by partition of voters in model TE. Then, there is a successful partition (V_1,V_2) of V such that c is the unique winner of the resulting two stage election. Because c is in the final round of the election, he or she has to be a winner of one of the subelecions, w.l.o.g say of (C,V_1). Since we are in the TE model, c is the unique winner of subelection (C,V_1). Thus, by deleting V_2 of the voter's set V, c is the unique winner of the resulting election, so voting system $\mathscr{E}$ is susceptible to constructive control by deleting voters.

For part 4, let (C,V) be an election, $c \in C$ a candidate who is the unique winner of the election under voting system $\mathscr{E}$. Suppose that voting system $\mathscr{E}$ is susceptible to destructive control by partition of candidates (in model TE or TP). Let $C' \subseteq C$ be again the set of candidates who attends in the final stage of the election. If $c \in C'$, then deleting the subset $C-C'$ of candidates makes c not a unique winner of the election. Otherwise, there is a subelection, say (C_1,V), where c participated, but lost. Thus, deleting the subset $C-C_1$ of candidates prevents c from being the unique winner of the election. Thus, the voting system $\mathscr{E}$ is susceptible to destructive control by deleting candidates. The same proof holds also for the run-off partition case. ❑

There are also three susceptibility links for voiced voting systems.

Theorem 3.3 ([HHR07a]**).**

1. *If a voiced voting system is susceptible to destructive control by partition of voters (in model TE or TP), then it is susceptible to destructive control by deleting voters.*

2. *Each voiced voting system is susceptible to constructive control by deleting candidates.*

3. *Each voiced voting system is susceptible to destructive control by adding candidates.*

Proof. ([HHR07a]) We first prove part 1. Let $\mathscr{E}$ be a voiced voting system applied to the election (C,V) with the unique winner $c \in C$. Suppose that $\mathscr{E}$ is susceptible to destructive control by partition of voters (in model TE or TP). Let us assume, that $\mathscr{E}$ is immune to destructive control by deleting voters, i.e., for any subset $V' \subseteq V$, c is the unique winner of (C,V'). This yields that c is the unique winner of both first stage subelections (C,V_1) and (C,V_2) in the partition of voters case. Thus, c is the only participant in the final stage and is the unique winner of the election. This is a contradiction to our basic assumption that $\mathscr{E}$ is susceptible to destructive control by partition of voters. Thus, our second assumption that $\mathscr{E}$ is immune to destructive control by deleting voters is false.

For part 2, let $\mathscr{E}$ be a voiced voting system applied to the election (C,V), where $c \in C$ is not the unique winner of the election. Deleting all the candidates corresponding to the set $C - \{c\}$ would leave c as the only candidate, and since $\mathscr{E}$ is a voiced voting system, this candidate is the unique winner of the election. Thus, $\mathscr{E}$ is susceptible to constructive control by deleting candidates.

Part 3 follows from part 2 of this theorem and part 2 of Theorem 3.1. ❑

In the following two chapters, we will analyze how two voting systems, sincere-strategy preference-based approval voting and fallback voting, behave under different control actions.

Chapter 4

Control in SP-AV

This chapter considers in detail a voting system introduced by Brams and Sanver [BS06], that combines approval and preference-based voting, with respect to our 22 control actions.

4.1 Definitions and Conventions

To distinguish this system from other systems that Brams and Sanver introduced with the same purpose of combining approval and preference-based voting [BS], we call the variant considered here (including the conventions and rules to be explained below) *sincere-strategy preference-based approval voting* (SP-AV, for short).

Definition 4.1 ([BS06]). *Let (C,V) be an election, where the voters both indicate approvals/disapprovals of the candidates and provide a tie-free linear ordering of all candidates. For each voter $v \in V$, an* AV strategy *of v is a subset $S_v \subseteq C$ such that v approves of all candidates in S_v and disapproves of all candidates in $C - S_v$. The list of AV strategies for all voters in V is called an* AV strategy profile *for (C,V). (We sometimes also speak of V's* AV strategy profile *for C.) For each $c \in C$, let $score_{(C,V)}(c) = \|\{v \in V \mid c \in S_v\}\|$ denote the number of c's approvals. Every candidate c with the largest $score_{(C,V)}(c)$ is a winner of the election (C,V).*

An AV strategy S_v of a voter $v \in V$ is said to be admissible *if S_v contains v's most preferred candidate and does not contain v's least preferred candidate. S_v is said to be* sincere *if for each $c \in C$, if v approves of c then v also approves of each candidate ranked higher than c (i.e., there are no gaps allowed in sincere approval strategies). An AV strategy profile for (C,V) is* admissible *(respectively,* sincere*) if the AV strategies of all voters in V are admissible (respectively, sincere).*

Admissibility and sincerity are quite natural requirements. In particular, requiring the voters to be sincere ensures that their preference rankings and their approvals/disapprovals are not contradictory. Admissibility yields a candidate set with at least two candidates. Note that an AV strategy is never admissible for less than two candidates. We mention in passing that in the work of Erdélyi, Nowak and Rothe [ENR08a] it is specifically required for single-candidate elections that each voter must approve of this candidate. In this thesis, we drop this requirement just like in [ENR08b] for two reasons. First, it in fact is not needed because the one candidate in a single-candidate election will always win—even with zero approvals (i.e., SP-AV is a "voiced" voting system). Second, it is very well comprehensible that a voter, when given just a single candidate, can get some satisfaction from denying this candidate his or her approval, even if he or she knows that this disapproval won't prevent the candidate from winning.

Note further that admissible AV strategies are not dominated in a game-theoretic sense [BF78]. Informally, an AV strategy S_v dominates an other AV strategy S_u, if S_v is always as good as, or better than S_u regardless of the setup. Sincere strategies for at least two candidates are always admissible if voters are neither allowed to approve of everybody nor to disapprove of everybody (i.e., if we require voters v to have only AV strategies S_v with $\emptyset \neq S_v \neq C$), a convention adopted by Brams and Sanver [BS06] and also adopted here [1]. Henceforth, we will assume that only sincere AV strategy profiles are considered (which by the above convention, whenever there are at least two candidates, necessarily are admissible), i.e., a vote with an insincere strategy will be considered void.

We extend our conventions on how to represent preferences presented in Section 3 as follows. In our constructions, we sometimes also insert a subset $B \subseteq C$ into such approval rankings, where we assume some arbitrary, fixed order of the candidates in B (e.g., "$a \mid B \; c$" means that a is approved of, while all $b \in B$ and c are disapproved of).

Definition 4.2. *Let $k \geq 1$ be a fixed integer. In k-*approval voting *(k-AV, for short), every voter approves of exactly k of the $m \geq k$ candidates,*[2] *and all candidates with the largest number of approvals win.* Sincere-strategy preference-based k-approval voting *(SP-k-AV, for short) in addition requires the voters to either approve or disapprove of the candidates via a sincere AV strategy, where the above-mentioned conventions apply: For elections with one candidate, each voter must approve of this candidate; and for elections (C,V) with at least two candidates, each voter $v \in V$ is required to have an AV strategy S_v with $\emptyset \neq S_v \neq C$, which implies $||C|| \geq k+1$ for $k > 1$.*

Before starting with the control scenarios, we have to carefully examine the control

[1]Brams and Sanver [BS06] actually preclude only the case $S_v = C$ for voters v. However, an AV strategy that disapproves of all candidates obviously is sincere, yet not admissible, which is why we also exclude the case of $S_v = \emptyset$.

[2]When there are less than k candidates, k-approval voting is not applicable.

actions considered. Control actions—specifically, those with respect to control via deleting or partitioning candidates or via partitioning voters—may have an undesirable impact on the resulting election in that they might violate our conventions about admissible AV strategies. That is why we define the following rule that preserves (or re-enforces) our conventions under such control actions:

> Whenever during or after a control action it happens that we obtain an election (C,V) with $\|C\| \geq 2$ and for some voter $v \in V$ we have $S_v = \emptyset$ or $S_v = C$, then each such voter's AV strategy is changed to approve of his or her top candidate and to disapprove of his or her bottom candidate. This rule re-enforces $\emptyset \neq S_v \neq C$ for each $v \in V$.

We have to tailor the control problems to sincere-strategy preference-based approval voting by requiring every election occuring in every control problems [3] to have a sincere AV strategy profile and to satisfy the above conventions and rules. In particular, this means that when the number of candidates is reduced (due to deleting candidates or partitioning candidates or voters), approval lines may have to be moved in accordance with the above rule.

For example, approval voting is known to be immune to eight of the twelve types of candidate control considered in [HHR07a]. The proofs of these results crucially employ the links between immunity/susceptibility for various control types in Theorems 3.1, 3.2, 3.3 and the fact that approval voting satisfies the unique version of the Weak Axiom of Revealed Preference (see [HHR07a, BTT92]). In contrast with approval voting, sincere-strategy preference-based approval voting does not satisfy Unique-WARP, and we will see later in Section 4.2.2 that it indeed is susceptible to each type of control considered here.

Proposition 4.1. *Sincere-strategy preference-based approval voting does not satisfy Unique-WARP.*

Proof. Consider the election (C,V) with candidate set $C = \{a,b,c,d\}$ and voter collection $V = \{v_1, v_2, v_3, v_4\}$. Removing candidate d changes the profile as follows according to the SP-AV rules:

$$\begin{array}{ll} v_1: & b\ c\ a\ |\ d \\ v_2: & c\ |\ a\ d\ b \\ v_3: & a\ b\ c\ |\ d \\ v_4: & b\ a\ c\ |\ d \end{array} \qquad \begin{array}{c} \text{is changed to} \\ \text{(by removing } d\text{):} \end{array} \qquad \begin{array}{l} b\ c\ |\ a \\ c\ |\ a\ b \\ a\ b\ |\ c \\ b\ a\ |\ c \end{array}$$

[3] Be it before, during, or after a control action—so, in particular, this also applies to the subelections in the partitioning cases.

Control by	Plurality Constr.	Plurality Destr.	SP-AV Constr.	SP-AV Destr.	AV Constr.	AV Destr.
Adding an Unlimited Number of Candidates	R	R	**R**	**R**	I	V
Adding a Limited Number of Candidates	R	R	**R**	**R**	I	V
Deleting Candidates	R	R	**R**	**R**	V	I
Partition of Candidates	TE: R TP: R	TE: R TP: R	**TE: R** **TP: R**	**TE: R** **TP: R**	TE: V TP: I	TE: I TP: I
Run-off Partition of Candidates	TE: R TP: R	TE: R TP: R	**TE: R** **TP: R**	**TE: R** **TP: R**	TE: V TP: I	TE: I TP: I
Adding Voters	V	V	**R**	**V**	R	V
Deleting Voters	V	V	**R**	**V**	R	V
Partition of Voters	TE: V TP: R	TE: V TP: R	**TE: R** **TP: R**	**TE: V** **TP: R**	TE: R TP: R	TE: V TP: V

Table 4.1: Overview of SP-AV results. Key: I means immune, R means resistant, V means vulnerable, TE stands for ties-eliminate, and TP for ties-promote. Results for SP-AV are new; their proofs are either new or draw on proofs from [HHR07a]. Results for plurality and approval voting, stated here to allow comparison, are due to Bartholdi, Tovey, and Trick [BTT92] and to Hemaspaandra, Hemaspaandra, and Rothe [HHR07a]. (The results for control by adding a limited number of candidates for plurality and approval voting, though not stated explicitly in [BTT92, HHR07a], follow immediately from the proofs of the corresponding results for the "unlimited" variant of the problem.)

Note that the approval/disapproval line has been moved in voters v_1, v_3, and v_4. Although c was the unique winner in (C,V), c is not a winner in $(\{a,b,c\},V)$ (in fact, b is the unique winner in $(\{a,b,c\},V)$). Thus, SP-AV does not satisfy Unique-WARP. ❑

4.2 Results for SP-AV

4.2.1 Overview

Theorem 4.1 below (see also Table 4.1) shows the complexity results regarding control of elections for SP-AV. As mentioned in the introduction, with 19 resistances and only three vulnerabilities, this system has more resistances and fewer vulnerabilities to control (for our 22 control types) than is currently known for any other natural voting system with a polynomial-time winner problem.

Theorem 4.1. *Sincere-strategy preference-based approval voting is resistant and vulnerable to the* 22 *types of control defined in Section 3.3 as shown in Table 4.1.*

4.2.2 Susceptibility

We start with susceptibility to candidate control.

Lemma 4.1. *SP-AV is susceptible to constructive and destructive control by adding candidates (in both the "limited" and the "unlimited" variant of the problem), by deleting candidates, and by partition of candidates (with or without run-off and for each in both tie-handling models, TE and TP).*

Proof. From Theorem 3.3 and the obvious fact that SP-AV is a voiced voting system, it immediately follows that SP-AV is susceptible to constructive control by deleting candidates and to destructive control by adding candidates (in both the "limited" and the "unlimited" variant of the problem).

Now, consider the election (C,V) with candidate set $C = \{a,b,c,d,e,f\}$ and voter collection $V = \{v_1, v_2, \dots, v_6\}$ and the following partition of C into $C_1 = \{a,c,d\}$ and $C_2 = \{b,e,f\}$:

	(C,V)	is partitioned into	(C_1,V)	and	(C_2,V)
v_1:	$a\ b\ c \mid d\ e\ f$		$a\ c \mid d$		$b \mid e\ f$
v_2:	$b\ c \mid a\ d\ e\ f$		$c \mid a\ d$		$b \mid e\ f$
v_3:	$a\ c \mid b\ d\ e\ f$		$a\ c \mid d$		$b \mid e\ f$
v_4:	$b\ a\ c \mid d\ e\ f$		$a\ c \mid d$		$b \mid e\ f$
v_5:	$a\ b\ d\ e\ c \mid f$		$a\ d \mid c$		$b\ e \mid f$
v_6:	$a\ b\ d\ f\ c \mid e$		$a\ d \mid c$		$b\ f \mid e$

With six approvals, c is the unique winner of (C,V). However, a is the unique winner of (C_1,V), which implies that c is not promoted to the final stage, regardless of whether we use the TE or TP tie-handling rule and regardless of whether we employ a partition of candidates with or without run-off. Thus, SP-AV is susceptible to destructive control by partition of candidates (with or without run-off and for each in both tie-handling models, TE and TP). By Theorem 3.2, SP-AV is also susceptible to destructive control by deleting candidates. By Theorem 3.1 in turn, SP-AV is also susceptible to constructive control by adding candidates (in both the "limited" and the "unlimited" variant of the problem).

Finally, we modify the above election as follows. Let (C,V') be identical to (C,V), except that $V' = \{v_1, v_2, \dots, v_5, v_7\}$ and v_7 has the sincere approval strategy: $a\ e\ d\ f\ c \mid b$. Note that a is not the unique winner of (C,V'), as a loses to c by 5 to 6. However, if we partition C into $C_1 = \{a,c,d\}$ and $C_2 = \{b,e,f\}$, then a is the unique winner in (C_1,V') and b is the unique winner in (C_2,V'). Since both subelections have a unique winner, it does not matter whether the TE rule or the TP rule is applied. The final-stage election is $(\{a,b\},V')$ in the case of run-off partition of candidates, and it is $(\{a,b,e,f\},V')$ in the case of partition of candidates. Since a wins against b in the former case by 4 to 2 and in the latter case by 5 to 4 (and e and f do even worse than b in this case), a is the unique winner in both cases. Thus, SP-AV is susceptible to constructive control by partition of candidates (with or without run-off and for each in both models, TE and TP). ❑

We now examine susceptibility in regards to voter control.

Lemma 4.2. *SP-AV is susceptible to constructive and destructive control by adding voters, by deleting voters, and by partition of voters in both tie-handling models, TE and TP.*

Proof. Consider the election (C,V) with candidate set $C = \{a,b,c,d,e,f\}$ and voter collection $V = \{v_1, v_2, \dots, v_8\}$ and partition V into $V_1 = \{v_1, v_2, v_3, v_4\}$ and $V_2 = \{v_5, v_6, v_7, v_8\}$. Thus, we change:

	(C,V)	into	(C,V_1)	and	(C,V_2)
v_1:	$a\ b\ c\ \mid\ d\ e\ f$		$a\ b\ c\ \mid\ d\ e\ f$		
v_2:	$a\ c\ \mid\ b\ d\ e\ f$		$a\ c\ \mid\ b\ d\ e\ f$		
v_3:	$c\ b\ a\ d\ \mid\ e\ f$		$c\ b\ a\ d\ \mid\ e\ f$		
v_4:	$a\ b\ \mid\ d\ e\ c\ f$		$a\ b\ \mid\ d\ e\ c\ f$		
v_5:	$a\ d\ c\ \mid\ b\ e\ f$				$a\ d\ c\ \mid\ b\ e\ f$
v_6:	$e\ b\ c\ d\ \mid\ a\ f$				$e\ b\ c\ d\ \mid\ a\ f$
v_7:	$d\ e\ c\ f\ \mid\ b\ a$				$d\ e\ c\ f\ \mid\ b\ a$
v_8:	$d\ f\ \mid\ b\ a\ c\ e$				$d\ f\ \mid\ b\ a\ c\ e$

With six approvals, c is the unique winner of (C,V). However, a is the unique winner of (C,V_1) and d is the unique winner of (C,V_2), which implies that c is not promoted to the final stage, regardless of whether we use the TE or TP tie-handling rule. (In the final-stage election $(\{a,d\},V)$, d wins by 5 to 3.) Thus, SP-AV is susceptible to destructive control by partition of voters in models TE and TP. By Theorem 3.3 and since SP-AV is a voiced system, SP-AV is also susceptible to destructive control by deleting voters. Finally, by Theorem 3.1, SP-AV is also susceptible to constructive control by adding voters.

Now, if we let a and c change their roles in the above election and argument, we see that SP-AV is also susceptible to constructive control by partition of voters in models TE and TP. By Theorem 3.2, susceptibility to constructive control by partition of voters in model TE implies susceptibility to constructive control by deleting voters. Again, by Theorem 3.1, SP-AV is also susceptible to destructive control by adding voters. ❑

4.2.3 Candidate Control

Theorems 4.2 and 4.4 below show that sincere-strategy preference-based approval voting is fully resistant to candidate control. This result should be contrasted with that of

Hemaspaandra, Hemaspaandra, and Rothe [HHR07a], who proved immunity and vulnerability for all cases of candidate control within approval voting (see Table 4.1). In fact, SP-AV has the same resistances to candidate control as plurality, and we will show that the construction presented in [HHR07a] to prove plurality resistant also works for sincere-strategy preference-based approval voting in all cases of candidate control except one—namely, except for constructive control by deleting candidates. Theorem 4.4 establishes resistance for this one missing case.

All resistance results in this section follow via a reduction from the NP-complete problem Hitting Set introduced in Section 2.2. Note that some of our proofs for SP-AV are based on constructions and arguments presented in [HHR07a] to prove the corresponding results for approval voting or plurality, whereas the remainder of our results require new insights to make the proof work for SP-AV. For completeness, we will present each construction here (even if the modification of a previous construction is rather straightforward), noting the differences to the related previous constructions.

Theorem 4.2. *SP-AV is resistant to all types of constructive and destructive candidate control defined in Section 3.3 except for constructive control by deleting candidates.*

Resistance of SP-AV to constructive control by deleting candidates, which is the missing case in Theorem 4.2, will be shown as Theorem 4.4 below.

The proof of Theorem 4.2 is based on a construction for plurality in [HHR07a], except that only the arguments for *destructive* candidate control are given there (simply because plurality was shown resistant to all cases of constructive candidate control already by Bartholdi, Tovey, and Trick [BTT92] via different constructions). We now provide the proof of Theorem 4.2 and the construction from [HHR07a] (slightly modified so as to be formally conform with the SP-AV voter representation) in order to (i) show that the same construction can be used to establish all but one resistances of SP-AV to *constructive* candidate control, and (ii) explain why constructive control by deleting candidates (which is missing in Theorem 4.2) does *not* follow from this construction.

Proof of Theorem 4.2. Susceptibility holds by Lemma 4.1 in each case. The resistance proofs are based on a reduction from Hitting Set and employ Construction 4.3 below, slightly modified so as to be formally conform with the SP-AV voter representation.

Construction 4.3 ([HHR07a])**.** *Let $(B, \mathscr{S}, k)$ be a given instance of Hitting Set, where $B = \{b_1, b_2, \dots, b_m\}$ is a set, $\mathscr{S} = \{S_1, S_2, \dots, S_n\}$ is a collection of subsets $S_i \subseteq B$, and $k \leq m$ is a positive integer. Define the election (C, V), where $C = B \cup \{c, w\}$ is the candidate set and where V consists of the following voters:*

1. *There are $2(m-k) + 2n(k+1) + 4$ voters of the form:*

$$c \mid w \ B.$$

2. *There are* $2n(k+1)+5$ *voters of the form:*

$$w \mid c \ B.$$

3. *For each* i*,* $1 \leq i \leq n$*, there are* $2(k+1)$ *voters of the form:*

$$S_i \mid c \ w \ (B - S_i).$$

4. *For each* j*,* $1 \leq j \leq m$*, there are two voters of the form:*

$$b_j \mid w \ c \ (B - \{b_j\}).$$

Since $score_{(\{c,w\},V)}(c) - score_{(\{c,w\},V)}(w) = 2k(n-1)+2n-1$ is positive (because of $n \geq 1$), c is the unique winner of election $(\{c,w\},V)$. The key observation is the following proposition, which can be proven as in [HHR07a].

Proposition 4.2 ([HHR07a])**.** *1. If* $\mathscr{S}$ *has a hitting set* B' *of size* k*, then* w *is the unique SP-AV winner of election* $(B' \cup \{c,w\},V)$*.*

2. *Let* $D \subseteq B \cup \{w\}$*. If* c *is not the unique SP-AV winner of election* $(D \cup \{c\},V)$*, then there exists a set* $B' \subseteq B$ *such that*
   (a) $D = B' \cup \{w\}$*,*
   (b) w *is the unique SP-AV winner of election* $(B' \cup \{c,w\},V)$*, and*
   (c) B' *is a hitting set of* $\mathscr{S}$ *of size less than or equal to* k*.*

Proof.

1. Suppose that B' is a hitting set of $\mathscr{S}$ of size k. Then we have the following scores in election $(B' \cup \{c,w\},V)$:

$$\begin{aligned} score(c) &= 2(m-k)+2n(k+1)+4, \\ score(w) &= 2(m-k)+2n(k+1)+5, \\ score(b_j) &\leq 2n(k+1)+2 \qquad \text{for each } j,\ 1 \leq j \leq m. \end{aligned}$$

   Thus, w is the unique SP-AV winner of election $(B' \cup \{c,w\},V)$.

2. Let $D \subseteq B \cup \{w\}$. Suppose c is not the unique SP-AV winner of election $(D \cup \{c\},V)$.

(a) Since for each $b \in D \cap B$, $score_{(D\cup\{c\},V)}(b) < score_{(D\cup\{c\},V)}(c)$, and c is not the unique SP-AV winner of election $(D \cup \{c\}, V)$, c can only lose the election against w. Thus, $D = B' \cup \{w\}$, where $B' \subseteq B$.

(b) This part follows immediatly from part (a).

(c) Let ℓ be the number of sets in $\mathscr{S}$ not hit by B', then:

$$\begin{aligned} score_{(B'\cup\{c,w\},V)}(w) &= 2n(k+1)+5+2(m-\|B'\|), \\ score_{(B'\cup\{c,w\},V)}(c) &= 2(m-k)+2n(k+1)+4+2(k+1)\ell. \end{aligned}$$

From part (a) we know, that $score_{(B'\cup\{c,w\},V)}(w) \geq score_{(B'\cup\{c,w\},V)}(c)$, i.e.,

$$2n(k+1)+5+2(m-\|B'\|) \geq 2(m-k)+2n(k+1)+4+2(k+1)\ell.$$

Since all variables are integers, the above inequality implies

$$0 \geq \|B'\| - k + (k+1)\ell$$

thus, $\ell = 0$, and so B' is a hitting set of $\mathscr{S}$ of size less than or equal to k.

❑

Corollary 4.1. *SP-AV is resistant to constructive and destructive control by adding candidates (both in the limited and the unlimited version of the problem).*

Proof. This corollary follows immediately from Proposition 4.2, via mapping the Hitting Set instance $(B, \mathscr{S}, k)$ to the set $\{c, w\}$ of qualified candidates and the set B of spoiler candidates, to the voter collection V, and by having c be the designated candidate in the destructive case and by having w be the designated candidate in the constructive case. ❑

Corollary 4.2. *SP-AV is resistant to destructive control by deleting candidates.*

Proof. Let the election (C,V) be given as in Construction 4.3 with distinguished candidate c. We claim that $\mathscr{S}$ has a hitting set of size at most k, if and only if c can be prevented from being a unique SP-AV winner by deleting at most $m-k$ candidates.

From left to right: Suppose, $\mathscr{S}$ has a hitting set $B' \subseteq B$ of size k. According to Proposition 4.2, c is not the unique SP-AV winner of the election $(B' \cup \{c,w\}, V)$. Thus, by deleting the set $B - B'$ of candidates, where $\|B - B'\| = m - k$, c is prevented to be a unique SP-AV winner of the election.

From right to left: Suppose, that c can be prevented from being a unique SP-AV winner by deleting at most $m-k$ candidates. Let $D \subseteq B \cup \{w\}$ be the set of deleted candidates, such that $c \notin D$. It immediately follows from Proposition 4.2, that $C - D - \{c\} = B' \cup \{w\}$, where B' is a hitting set of $\mathscr{S}$ of size at most k. ❑

Corollary 4.3. *SP-AV is resistant to constructive and destructive control by partition of candidates and run-off partition of candidates (for each in both models TE and TP).*

Proof. We only prove the constructive case. Let the election (C,V) be given as in Construction 4.3 with distinguished candidate w. We claim that $\mathscr{S}$ has a hitting set of size at most k, if and only if w can be made the unique SP-AV winner by partition or run-off partition of candidates (for each in both models TE and TP).

From left to right: Suppose, $\mathscr{S}$ has a hitting set $B' \subseteq B$ of size k. Partition the set of candidates into the two subsets $C_1 = B' \cup \{c,w\}$ and $C_2 = C - C_1$. According to Proposition 4.2, w is the unique SP-AV winner of the election $(B' \cup \{c,w\},V)$. Then, the score of w is at least $2(m-k)+4n(k+1)+9$ in the final stage, and the score of any participant from the other subelection is at most $2n(k+1)+2$, thus w is the unique winner of the election.

From right to left: Suppose, there exists a partition such that w is the unique SP-AV winner of the election. In this case, c is not the unique SP-AV winner of the election. Then, there has to be a subset $D \subseteq B \cup \{w\}$ of candidates such that c is not the unique SP-AV winner of the election $(D \cup \{c\},V)$. Due to Proposition 4.2, there exists a size k hitting set of $\mathscr{S}$.

For the destructive case simply change the roles of c and w. ❑

Corollaries 4.1, 4.2, and 4.3 complete the proof for Theorem 4.2. ❑ Theorem 4.2

Turning now to the missing case mentioned in Theorem 4.2 above: Why does Construction 4.3 not work for constructive control by deleting candidates? Informally put, the reason is that c is the only serious rival of w in the election (C,V) of Construction 4.3, so by simply deleting c the chair could make w the unique SP-AV winner, regardless of whether $\mathscr{S}$ has a hitting set of size k. However, via a different construction, we can prove resistance also in this case.

Theorem 4.4. *SP-AV is resistant to constructive control by deleting candidates.*

Proof. Susceptibility holds by Lemma 4.1. To prove resistance, we provide a reduction from Hitting Set. Let $(B,\mathscr{S},k)$ be a given instance of Hitting Set, where $B = \{b_1,b_2,\ldots,b_m\}$ is a set, $\mathscr{S} = \{S_1,S_2,\ldots,S_n\}$ is a collection of subsets $S_i \subseteq B$, and $k < m$ is a positive integer.[4]

Define the election (C,V), where $C = B \cup \{w\}$ is the candidate set and V is the collection of voters. We assume that the candidates in B are in an arbitrary but fixed order, and for each voter below, this order is also used in each subset of B. For example, if

[4]Note that if $k = m$ then B is always a hitting set of size at most k (provided that $\mathscr{S}$ contains only nonempty sets—a requirement that doesn't affect the NP-completeness of the problem), and we thus may require that $k < m$.

$B = \{b_1, b_2, b_3, b_4\}$ and some subset $S_i = \{b_1, b_3\}$ of B occurs in some voter then this voter prefers b_1 to b_3, and so does any other voter whose preference list contains S_i.

V consists of the following $4n(k+1)+4m-2k+3$ voters:

1. For each i, $1 \leq i \leq n$, there are $2(k+1)$ voters of the form:
$$S_i \ | \ (B - S_i) \ w.$$
2. For each i, $1 \leq i \leq n$, there are $2(k+1)$ voters of the form:
$$(B - S_i) \ w \ | \ S_i.$$
3. For each j, $1 \leq j \leq m$, there are two voters of the form:
$$b_j \ | \ w \ (B - \{b_j\}).$$
4. There are $2(m-k)$ voters of the form:
$$B \ | \ w.$$
5. There are three voters of the form:
$$w \ | \ B.$$

Since for each $b_j \in B$, the difference

$$score_{(C,V)}(w) - score_{(C,V)}(b_j) = 2n(k+1)+3-(2n(k+1)+2+2(m-k)) = 1-2(m-k)$$

is negative (due to $k < m$), w loses to each member of B and so does not win election (C,V).

We claim that $\mathscr{S}$ has a hitting set B' of size k if and only if w can be made the unique SP-AV winner by deleting at most $m-k$ candidates.

From left to right: Suppose $\mathscr{S}$ has a hitting set B' of size k. Then, for each $b_j \in B'$,

$$score_{(B' \cup \{w\},V)}(w) - score_{(B' \cup \{w\},V)}(b_j) =$$
$$2n(k+1)+2(m-k)+3-(2n(k+1)+2+2(m-k)) = 1,$$

since the approval line is moved for $2(m-k)$ voters in the third group, thus transferring their approvals from members of $B - B'$ to w. So w is the unique SP-AV winner of election $(B' \cup \{w\}, V)$. Since $B' \cup \{w\} = C - (B - B')$, it follows from $\|B\| = m$ and $\|B'\| = k$ that deleting $m-k$ candidates from C makes w the unique SP-AV winner.

From right to left: Let $D \subseteq B$ be any set such that $\|D\| \leq m-k$ and w is the unique SP-AV winner of election $(C-D,V)$. Let $B' = (C-D)-\{w\}$. Note that $B' \subseteq B$ and that we have the following scores in $(B' \cup \{w\},V)$:

$$\begin{aligned}
score_{(B'\cup\{w\},V)}(w) &= 2(n-\ell)(k+1)+2(m-\|B'\|)+3, \\
score_{(B'\cup\{w\},V)}(b_j) &\leq 2n(k+1)+2(k+1)\ell+2+2(m-k) \quad \text{for each } b_j \in B',
\end{aligned}$$

where ℓ is the number of sets $S_i \in \mathscr{S}$ that are not hit by B', i.e., $B' \cap S_i = \emptyset$. Recall that for each i, $1 \leq i \leq n$, all of the $2(k+1)$ voters of the form $S_i \mid (B-S_i)\ w$ in the first voter group have ranked the candidates in the same order. Thus, for each i, $1 \leq i \leq n$, whenever $B' \cap S_i = \emptyset$ one and the same candidate in B' benefits from moving the approval line, namely the candidate occurring first in our fixed ordering of B'. Call this candidate b and note that

$$score_{(B'\cup\{w\},V)}(b) = 2n(k+1)+2(k+1)\ell+2+2(m-k).$$

Since w is the unique SP-AV winner of $(B' \cup \{w\},V)$, w has more approvals than any candidate in B' and in particular more than b. Thus, we have

$$\begin{aligned}
& score_{(B'\cup\{w\},V)}(w) - score_{(B'\cup\{w\},V)}(b) \\
&= 2(n-\ell)(k+1)+2(m-\|B'\|)+3-2n(k+1)-2\ell(k+1)-2-2(m-k) \\
&= 1+2(k-\|B'\|)-4\ell(k+1) > 0.
\end{aligned}$$

Solving this inequality for ℓ, we obtain

$$0 \leq \ell < \frac{1+2(k-\|B'\|)}{4(k+1)} < \frac{4+4k}{4(k+1)} = 1.$$

Thus $\ell = 0$. It follows that $1+2(k-\|B'\|) > 0$, which implies $\|B'\| \leq k$. Thus, B' is a hitting set of size at most k. ❑

4.2.4 Voter Control

Turning now to control by adding and by deleting voters, it is known from [HHR07a] that approval voting is resistant to constructive control and is vulnerable to destructive control (see Table 4.1).[5] These proofs can be modified so as to also apply to sincere-strategy preference-based approval voting.

[5] Procaccia, Rosenschein, and Zohar [PRZ07] proved in their interesting "multi-winner" model (which generalizes Bartholdi, Tovey, and Trick's model [BTT92] by adding a utility function and some other parameters) that approval voting is resistant to constructive control by adding voters. According to Footnote 13 of [HHR07a], this resistance result immediately follows from the corresponding resistance result in [HHR05, HHR07a], essentially due to the fact that lower bounds in more flexible models are inherited from more restrictive models.

Theorem 4.5. *SP-AV is resistant to constructive control by adding voters and by deleting voters and is vulnerable to destructive control by adding voters and by deleting voters.*

Proof. Susceptibility holds by Lemma 4.2 in all cases. To prove resistance to constructive control by adding voters (respectively, by deleting voters), the construction of [HHR07a, Thm. 4.43] (respectively, of [HHR07a, Thm. 4.44]) works, modified only by specifying voter preferences consistently with the voters' approval strategies (and, in the deleting-voters case, by adding a dummy candidate who is disapproved and ranked last by every voter in the construction to ensure an admissible AV strategy profile). These constructions provide polynomial-time reductions from the NP-complete problem Exact Cover by Three-Sets defined in Section 2.2.

We now give the proof for these two resistance results. In both cases, we start from an X3C instance $(B, \mathscr{S})$.

In the case of constructive control by adding voters, for a given X3C instance $(B, \mathscr{S})$, where $B = \{b_1, b_2, \dots, b_{3m}\}$, $m > 1$, is a set and $\mathscr{S} = \{S_1, S_2, \dots, S_n\}$ is a collection of subsets $S_i \subseteq B$ with $\|S_i\| = 3$ for each i, we define the election (C, V), with candidate set $C = B \cup \{w\}$ and with V consisting of $m-2$ registered voters each of the form

$$B \mid w.$$

Further, we define W to consist of the following n unregistered voters: For each i, $1 \leq i \leq n$, there is one voter of the form

$$w \ S_i \mid (B - S_i).$$

We claim that $\mathscr{S}$ has an exact cover for B if and only if w can be made the unique SP-AV winner by adding at most m voters.

From left to right: Suppose $\mathscr{S}$ contains an exact cover for B. Add the m voters of W corresponding to this exact cover to V. Let $W' \subseteq W$ be the set of unregistered voters thus added. Then $score_{(C,V \cup W')}(w) = m$ and $score_{(C,V \cup W')}(b_i) = m-1$ for all $1 \leq i \leq 3m$, so w is the unique winner.

From right to left: Let W' be any subset of W such that $\|W'\| \leq m$ and w is the unique winner of the election $(C, V \cup W')$. It follows that $\|W'\| = m$, and each $b_i \in B$ can gain only one point. Thus, the m voters in W' correspond to an exact cover for B.

In the case of constructive control by deleting voters, define the value $\ell_j = \|\{S_i \in \mathscr{S} \mid b_j \in S_i\}\|$ for each j, $1 \leq j \leq 3m$. Define the election (C, V), where $C = B \cup \{w, d\}$ is the set of candidates, w is the distinguished candidate, and V is the following collection of $2n$ voters:

1. For each i, $1 \leq i \leq n$, there is one voter of the form:

$$S_i \mid (B - S_i) \ w \ d.$$

2. For each i, $1 \leq i \leq n$, there is one voter of the form:

$$w \ B_i \ | \ (B - B_i) \ d,$$

where $B_i = \{b_j \in B \,|\, i \leq n - \ell_j\}$.

Note that $score_{(C,V)}(w) = n$ and $score_{(C,V)}(b_j) = n$ for all $b_j \in B$.

We claim that $\mathscr{S}$ has an exact cover for B if and only if w can be made the unique SP-AV winner by deleting at most m voters.

From left to right: Suppose $\mathscr{S}$ contains an exact cover for B. Delete the m voters corresponding to this exact cover. Let $V' \subseteq V$ be the set of voters thus deleted. Then $score_{(C,V-V')}(w) = n$, $score_{(C,V-V')}(d) < n$, and $score_{(C,V-V')}(b_j) = n-1$ for all $b_j \in B$. Thus w is the unique winner.

From right to left: Let V' be any subset of V such that $\|V'\| \leq m$ and w is the unique winner of the election $(C, V - V')$. We can assume that the voters corresponding to V' have disapproved of the distinguished candidate w. Since each candidate $b_j \in B$ must lose at least one point and by our assumption that only voters from the first group have been deleted, it follows that the deleted voters correspond to a cover. Since the number of deleted voters is at most m, they correspond to an exact cover for B.

The polynomial-time algorithms showing that approval voting is vulnerable to destructive control by adding voters and by deleting voters [HHR07a, Thm. 4.24] can be straightforwardly adapted to also work for sincere-strategy preference-based approval voting, since no approval lines are moved in these control scenarios. For completeness, we provide these proofs.

In the case of destructive control by adding voters, the input to the algorithm is an election (C,V), a collection W of additional voters (where each voter v in $V \cup W$ has a sincere AV strategy S_v with $\emptyset \neq S_v \neq C$), a distinguished candidate $c \in C$, and a nonnegative integer ℓ. The output will be either a subset $W' \subseteq W$ of voters such that $\|W'\| \leq \ell$, and adding the voters of W' to V ensures that c is not a unique winner, or it will be "control impossible" if no such subset exists. If $C = \{c\}$ then output "control impossible" and halt, since one candidate is always the unique winner independent of the number of voters. If $\|C\| > 1$ and c is already not the unique SP-AV winner of the election (C,V) then output $W' = \emptyset$ and halt. Otherwise, for each candidate $d \neq c$ define $surplus(c,d) = score(c) - score(d)$. Among all candidates $i \neq c$ such that there exist $surplus_{(C,V)}(c,i)$ voters in W who approve of i and who disapprove of c, let j be one such candidate for which $surplus_{(C,V)}(c,j)$ is minimum. Output the $surplus_{(C,V)}(c,j)$ voters from W who approve of j and disapprove of c. If there is no such candidate j, then output "control impossible" and halt.

In the case of destructive control by deleting voters, the input to the algorithm is an election (C,V) (where each voter $v \in V$ has a sincere AV strategy S_v with $\emptyset \neq S_v \neq C$), a distinguished candidate $c \in C$, and a nonnegative integer ℓ. The output will be either a subset $V' \subseteq V$ of voters such that $\|V'\| \leq \ell$ and deleting the voters of V' from V ensures that c is not a unique winner, or it will be "control impossible" if no such subset exists. If $C = \{c\}$ then again output "control impossible" and halt. If $\|C\| > 1$ and c already is not a unique SP-AV winner of the election (C,V), then output $V' = \emptyset$ and halt. Otherwise, let $j \neq c$ be the candidate for whom $surplus_{(C,V)}(c, j)$ is minimum. If $surplus_{(C,V)}(c, j) > \ell$ then output "control impossible" and halt. Otherwise, output the $surplus_{(C,V)}(c, j)$ voters from V who approve of c and disapprove of j. ❑

We now prove that, just like plurality, sincere-strategy preference-based approval voting is resistant to constructive and destructive control by partition of voters in model TP. In fact, the proof presented in [HHR07a] for plurality in these two cases also works for SP-AV with minor modifications. In contrast, approval voting is vulnerable to the destructive variant of this control type [HHR07a].

Theorem 4.6. *SP-AV is resistant to constructive and destructive control by partition of voters in model TP.*

Proof. The proof is again based on Construction 4.3, but the reduction is now from Restricted Hitting Set (see Section 2.2). Now, the key observation is the following proposition, which can be proven as in [HHR07a].

Proposition 4.3 ([HHR07a])**.** *Let $(B, \mathscr{S}, k)$ be a given Restricted Hitting Set instance, where $B = \{b_1, b_2, \ldots, b_m\}$ is a set, $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ is a collection of subsets $S_i \subseteq B$, and $k \leq m$ is a positive integer such that $n(k+1)+1 \leq m-k$. If (C,V) is the election resulting from $(B, \mathscr{S}, k)$ via Construction 4.3, then the following three statements are equivalent:*

1. *$\mathscr{S}$ has a hitting set of size less than or equal to k.*
2. *V can be partitioned such that w is the unique SP-AV winner in model TP.*
3. *V can be partitioned such that c is not the unique SP-AV winner in model TP.*

Proof. It is trivial that if V can be partitioned such that w is the unique SP-AV winner in model TP, then c is not the unique SP-AV winner. Thus, statement two implies statement three.

Let $(B, \mathscr{S}, k)$ be a given Restricted Hitting Set instance as defined above in Proposition 4.3. Suppose, that $\mathscr{S}$ has a hitting set B' of size less than or equal to k. Partition V into V_1 and V_2 such that V_1 consists of one voter of the form $\;c \mid w\; B,\;$ and for each $b_j \in B'$ one voter of the form $\;b_j \mid w\; c\; (B - \{b_j\}),\;$ and $V_2 = V - V_1$. The winners of subelection (C,V_1) are all candidates corresponding to B' and w, since they all have a score of 1, all the other candidates have no points at all. In the second subelection (C,V_2), c is the unique winner, since $score_{(C,V_2)}(b_j) \leq 2n(k+1)+2$ for each $1 \leq j \leq n$, $score_{(C,V_2)}(w) = 2n(k+1)+5$, and $score_{(C,V_2)}(c) = 2(m-k)+2n(k+1)+3$. Note, that $score_{(C,V_2)}(c) - score_{(C,V_2)}(w) = 2(m-k-1) > 0$ due to the restriction in the Restricted Hitting Set instance. By Proposition 4.2, w is the unique SP-AV winner of the final stage. This proves that the first statement implies the second and third statements.

Showing that statement three implies statement one will complete the proof. Suppose there is a partition of V into V_1 and V_2 such that c is not the unique SP-AV winner in the TP model. We claim that c is an SP-AV winner of one of the subelections (C,V_1) or (C,V_2). For a contradiction, suppose that c is not a winner in both subelections. Then, there have to be two candidates $x,y \in B \cup \{w\}$ such that candidate x is the winner of (C,V_1) and y is the winner of (C,V_2). Then, the following holds:

$$\begin{aligned} score_{(C,V_1)}(x) + score_{(C,V_2)}(y) &\geq score_{(C,V_1)}(c) + score_{(C,V_2)}(c) + 2 \\ &\geq score_{(C,V)}(c) + 2. \end{aligned} \tag{4.2.1}$$

Since c has the highest score of all candidates in (C,V), we have $x \neq y$. Then,

$$\begin{aligned} score_{(C,V_1)}(x) + score_{(C,V_2)}(y) &\leq score_{(C,V)}(w) + score_{(C,V)}(b_j) \\ &\leq 2n(k+1)+5+2n(k+1)+2 \\ &\leq 2n(k+1)+5+2(m-k) \\ &\leq score_{(C,V)}(c)+1, \end{aligned}$$

which contradicts to Equation 4.2.1. Thus, c is a winner of one of the subelections (C,V_1) or (C,V_2) and will participate in the final round of the election.

Since c is not a unique SP-AV winner of the final stage of the election $(D \cup \{c\}, V)$, where $D \subseteq B \cup \{w\}$, by Proposition 4.2, $\mathscr{S}$ has a hitting set of size less than or equal to k. ❑

Theorem 4.6 now follows immediately from Proposition 4.3. ❑ Theorem 4.6

Finally, we turn to control by partition of voters in model TE. For this control type, Hemaspaandra et al. [HHR07a] proved approval voting resistant in the constructive case and vulnerable in the destructive case. We have the same results for sincere-strategy preference-based approval voting. Our resistance proof in the constructive case (see the proof of Theorem 4.7) is similar to the corresponding proof of resistance in [HHR07a]. However, while our polynomial-time algorithm showing vulnerability for SP-AV in the destructive case (see the proof of Theorem 4.8) is based on the corresponding polynomial-time algorithm for approval voting in [HHR07a], it extends their algorithm in a nontrivial way.

Theorem 4.7. *SP-AV is resistant to constructive control by partition of voters in model TE.*

Proof. Susceptibility holds by Lemma 4.2. The proof of resistance is based on the construction of [HHR07a, Thm. 4.46] with only minor changes. Let an X3C instance $(B,\mathscr{S})$ be given, where $B = \{b_1, b_2, \ldots, b_{3m}\}$, $m > 1$, is a set and $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$ is a collection of subsets $S_i \subseteq B$ with $\|S_i\| = 3$ for each i. Without loss of generality, we may assume that $n \geq m$. Define the value $\ell_j = \|\{S_i \in \mathscr{S} \mid b_j \in S_i\}\|$ for each j, $1 \leq j \leq 3m$ as in the proof of Theorem 4.5.

Define the election (C,V), where $C = B \cup \{w,x,y\} \cup Z$ is the candidate set with the distinguished candidate w, $Z = \{z_1, z_2, \ldots, z_n\}$, and where V is defined to consist of the following $4n+m$ voters:

1. For each i, $1 \leq i \leq n$, there is one voter of the form:

$$y \;\; S_i \;\mid\; w \;\; ((B - S_i) \cup \{x\} \cup Z).$$

2. For each i, $1 \leq i \leq n$, there is one voter of the form:

$$y \;\; z_i \;\mid\; w \;\; (B \cup \{x\} \cup (Z - \{z_i\})).$$

3. For each i, $1 \leq i \leq n$, there is one voter of the form:

$$w \;\; (Z - \{z_i\}) \;\; B_i \;\mid\; x \;\; y \;\; z_i \;\; (B - B_i),$$

   where $B_i = \{b_j \in B \mid i \leq n - \ell_j\}$.

4. There are $n+m$ voters of the form:

$$x \;\mid\; y \;\; (B \cup \{w\} \cup Z).$$

Since the above construction is only slightly modified from the proof of [HHR07a, Thm. 4.46], so as to be formally conform with the SP-AV voter representation, the same argument as in that proof shows that $\mathscr{S}$ has an exact cover for B if and only if w can be made the unique SP-AV winner by partition of voters in model TE. Note that, in the present control scenario, approval voting and SP-AV can differ only in the run-off, but the construction ensures that they don't differ there.

From left to right, if $\mathscr{S}$ has an exact cover for B then partition the set of voters as follows: V_1 consists of the m voters of the form $y \;\; S_i \;\mid\; w \;\; ((B - S_i) \cup \{x\} \cup Z)$ that correspond to the sets in the exact cover, of the $n+m$ voters who approve of only x, and of the n voters who approve of y and z_i, $1 \leq i \leq n$. Let $V_2 = V - V_1$. It follows that w is the unique SP-AV winner of both subelection (C,V_2) and the run-off, simply because no candidate proceeds to the run-off from the other subelection, (C,V_1), in which x and y tie for winner with a score of $n+m$ each.

From right to left, suppose w can be made the unique SP-AV winner by partition of voters in model TE. Let (V_1,V_2) be a partition of V such that w is the unique SP-AV winner of the run-off. According to model TE, w must also be the unique SP-AV winner of one subelection, say of (C,V_1). Note that each voter of the form $y\ z_i \mid w\ (B \cup \{x\} \cup (Z - \{z_i\}))$ has to be in V_2 (otherwise, we would have $score_{(C,V_1)}(w) = score_{(C,V_1)}(z_i)$ for at least one i, and so w would not be the unique SP-AV winner of (C,V_1) anymore). However, if there were more than m voters of the form $y\ S_i \mid w\ ((B - S_i) \cup \{x\} \cup Z)$ in V_2 then $score_{(C,V_2)}(y) > n+m$, and so y would be the unique SP-AV winner of the other subelection, (C,V_2). But then, also in the SP-AV model, y would win the run-off against w because $score_{(\{w,y\},V)}(y) = 3n+m > n = score_{(\{w,y\},V)}(w)$, which contradicts the assumption that w has been made the unique SP-AV winner by the partition (V_1,V_2). Hence, there are at most m voters of the form $y\ S_i \mid w\ ((B - S_i) \cup \{x\} \cup Z)$ in V_2, and these m voters correspond to an exact cover of B. ❑

Theorem 4.8. *SP-AV is vulnerable to destructive control by partition of voters in model TE.*

Proof. Susceptibility holds by Lemma 4.2. To prove vulnerability, we describe a polynomial-time algorithm showing that (and how) the chair can exert destructive control by partition of voters in model TE for sincere-strategy preference-based approval voting. Our algorithm extends the polynomial-time algorithm designed by Hemaspaandra et al. [HHR07a] to prove approval voting vulnerable to this type of control. Specifically, our algorithm adds Loop 2 below to their algorithm, and we will explain below why it is necessary to add this second loop.

We adopt the following notation from [HHR07a]. Let (C,V) be an election, and for each voter $v \in V$, let $S_v \subseteq C$ denote v's AV strategy. In each iteration of Loop 1 in the algorithm below, we will consider three candidates, a, b, and c. Define the following five numbers:

$$W_c = \|\{v \in V \mid a \notin S_v,\ b \notin S_v,\ c \in S_v\}\|, \qquad L_c = \|\{v \in V \mid a \in S_v,\ b \in S_v,\ c \notin S_v\}\|,$$
$$D_a = \|\{v \in V \mid a \in S_v,\ b \notin S_v,\ c \notin S_v\}\|, \quad D_b = \|\{v \in V \mid a \notin S_v,\ b \in S_v,\ c \notin S_v\}\|, \text{ and}$$
$$D_{ac} = \|\{v \in V \mid a \in S_v,\ b \notin S_v,\ c \in S_v\}\|.$$

In addition, we introduce the following notation. Given an election (C,V) and two distinct candidates $x,y \in C$, let $diff(x,y)$ denote the number of voters in V who prefer x to

y minus the number of voters in V who prefer y to x. Define B_x to be the set of candidates $y \neq x$ in C such that $\mathit{diff}(y,x) \geq 0$.

The input to our algorithm is an election (C,V), where each voter $v \in V$ has a sincere AV strategy S_v with $\emptyset \neq S_v \neq C$ (otherwise, the input is considered malformed and outright rejected), and a distinguished candidate $c \in C$. On this input, our algorithm works as follows.

1. **Checking the trivial cases:** can be done as in the case of approval voting, see the proof of [HHR07a, Thm. 4.21]. In particular, if $C = \{c\}$ then output "control impossible" and halt, since c cannot help but win. If C contains more candidates than only c but c already is not the unique SP-AV winner in (C,V) then output the (successful) partition $(V,\emptyset)$ and halt. Otherwise, if $\|C\| = 2$ then output "control impossible" and halt, as c is the unique SP-AV winner of (C,V) in the current case, and so, however the voters are partitioned, c must win—against the one rivalling candidate—at least one subelection and also the run-off.

2. **Loop 1:** For each $a,b \in C$ such that $\|\{a,b,c\}\| = 3$, check whether V can be partitioned into V_1 and V_2 such that $\mathit{score}_{(C,V_1)}(a) \geq \mathit{score}_{(C,V_1)}(c)$ and $\mathit{score}_{(C,V_2)}(b) \geq \mathit{score}_{(C,V_2)}(c)$. As shown in the proof of [HHR07a, Thm. 4.21], this is equivalent to checking

   $$W_c - L_c \leq D_a + D_b. \tag{4.2.2}$$

   If (4.2.2) fails, this a and b cannot prevent c from being the unique winner of at least one subelection and thus also of the run-off, so we move on to test the next a and b in this loop. If (4.2.2) holds, however, output the partition (V_1,V_2) and halt, where V_1 consists of the voters contributing to D_a, of the voters contributing to D_{ac}, and of $\min(W_c,D_a)$ voters contributing to W_c, and where $V_2 = V - V_1$.

3. **Loop 2:** For each $d \in B_c$, partition V as follows. Let V_1 consist of all voters in V who approve of d, and let $V_2 = V - V_1$. If d is the unique winner of (C,V_1), then output (V_1,V_2) as a successful partition and halt. Otherwise, go to the next $d \in B_c$.

4. **Termination:** If in no iteration of either Loop 1 or Loop 2 a successful partition of V was found, then output "control impossible" and halt.

Let us give a short explanation of why Loop 2 is needed for SP-AV by stressing the difference with approval voting. As shown in the proof of [HHR07a, Thm. 4.21], if none of the trivial cases applied, then condition (4.2.2) holds for some $a,b \in C$ with $\|\{a,b,c\}\| = 3$ if and only if destructive control by partition of voters in model TE is possible for approval voting. Thus, for approval voting, if Loop 1 was not successful for any such a and b, we may immediately jump to the termination stage, where the algorithm outputs “control impossible” and halts. In contrast, if none of the trivial cases apply, then the existence of candidates a and b with $\|\{a,b,c\}\| = 3$ who satisfy (4.2.2) is *not* equivalent to destructive control by partition of voters in model TE being possible for SP-AV: it is a sufficient, yet not a necessary condition. The reason is that even if there are no candidates a and b who can prevent c from winning one subelection (in some partition of voters) and from proceeding to the run-off, it might still be possible that c loses or ties the run-off due to our rule of moving the approval line in order to re-enforce our conventions for SP-AV in this control scenario.

Indeed, if Loop 1 was not successful, c will lose or tie the run-off exactly if there exists a candidate $d \neq c$ such that $\mathit{diff}(d,c) \geq 0$ and d can win one subelection (for some partition of voters). This is precisely what is being checked in Loop 2. Indeed, note that the partition (V_1,V_2) chosen in Loop 2 for $d \in B_c$ is the best possible partition for d in the following sense: If d is not the unique SP-AV winner of subelection (C,V_1) then, for each $W \subseteq V$, d is not the unique SP-AV winner of subelection (C,W). To see this, simply note that if d is not the unique SP-AV winner of (C,V_1), then there is some candidate x with $\mathit{score}_{(C,V_1)}(x) = \mathit{score}_{(C,V_1)}(d) = \|V_1\|$, which by our choice of V_1 implies $\mathit{score}_{(C,W)}(x) \geq \mathit{score}_{(C,W)}(d)$ for each subset $W \subseteq V$. ❑

Comparing Theorem 4.6 and Theorem 4.8, one can see that destrutive control by partition of voters in SP-AV yield different results depending on the tie-handling rule used.

4.3 Bribery and Manipulation in SP-AV

Faliszewski et al. [FHH06a] introduced the problem of bribery for voting systems. In this scenario, an external agent seeks to influence the outcome of an election by bribing some of the voters so as to change their preferences over the candidates.

Faliszewski et al. [FHH06a, FHH06c] proved that approval voting is resistant to constructive bribery (even in the unweighted, unpriced variant of the problem). Since the voters’ approval lines are not moved in bribery (as the number of candidates remains unchanged), the same result for SP-AV can be shown analogously.

Theorem 4.9. *SP-AV is resistant to constructive bribery.*

Faliszewski et al. showed the NP-hardness of constructive approval-bribery via reduction from the NP-complete problem X3C (see Section 2.2).

In contrast, Theorem 4.10 below shows vulnerability to destructive bribery for approval voting and SP-AV, even in the weighted, priced variant of the problem.

Theorem 4.10. *Both approval voting and SP-AV are vulnerable to destructive bribery, even in the weighted, priced version of the problem.*

Proof. Again, since the voters' approval lines are not moved in bribery, it is enough to prove the theorem for approval voting; an analogous proof also works for SP-AV. We describe a polynomial-time greedy algorithm that determines how the briber can bribe some voters, without exceeding his or her budget, such that a distinguished candidate will not be a unique approval winner if that is at all possible.

We introduce the following notation. Let w_v denote the weight and let p_v denote the price of voter v. Let a and b be any two distinct candidates in an election E with $score_E(a) > score_E(b)$. Define the *cost-effectiveness of voter v for the candidates a and b* by

$$\mathrm{CE}_v(a,b) = \begin{cases} \dfrac{p_v}{2w_v} & \text{if } a \text{ is approved and } b \text{ is disapproved by } v \\ \dfrac{p_v}{w_v} & \text{if either both } a \text{ and } b \text{ are approved or both } a \text{ and } b \text{ are disapproved} \\ \infty & \text{otherwise.} \end{cases}$$

The cost-effectiveness gives us the price per unit weight that b can gain on a when voter v is being bribed suitably. We say *a voter v_i is more cost-effective than a voter v_j* if v_i's cost-effectiveness is less than v_j's. The input of the algorithm is an election (C,V), with distinguished candidate $c \in C$, and a budget k. Recall that each voter $v \in V$ has a nontrivial sincere AV strategy S_v with $\emptyset \neq S_v \neq C$, a weight w_v and a price p_v. The algorithm works as follows:

(1) **Trivial cases:** If $C = \{c\}$ then output "bribery impossible" and halt, since there is no other candidate who could win the election. If c is already not the unique winner then output the given AV strategy profile for (C,V) and halt.

(2) **Loop:** For each candidate $a \in C$, compute the cost-effectiveness $\mathrm{CE}_v(c,a)$ for each voter $v \in V$. Keep bribing the most cost-effective voter v as long as the budget k is not exceeded in the following way: If (c is approved and a is disapproved by v) or (both c and a are approved by v) or (both c and a are disapproved by v) then bribe v to approve of a and to disapprove of c. If this makes sure that c isn't the unique winner anymore, then output the thus modified AV strategy profile and halt. Otherwise go to the next $a \in C$.

(3) **Termination:** If we haven't found a candidate a in any loop iteration, such that we could successively bribe voters to reach our goal, then output "bribery impossible" and halt.

Let us give a short explanation of the loop. Since the distinguished candidate c has a surplus of $score_{(C,V)}(c) - score_{(C,V)}(a)$ points against candidate a, the briber wishes to buy at least that many points as cheap as possible. This is exactly what the loop does: The briber seeks greedily to bribe the cheapest voters, one after the other, until either the budget limit k is exceeded (in which case the algorithm halts without success) or until the bribery was performed successfully. ❑

Note that our algorithm presented not only solves the decision problem for destructive bribery but also determines how to successfully bribe if that is possible.

We are turning now our attention to manipulation.

Theorem 4.11. *Constructive and destructive approval-manipulation, approval-weighted-manipulation, SP-AV-manipulation and SP-AV-weighted-manipulation are in* P.

Proof. We only give the proof for constructive SP-AV-weighted-manipulation. Let $(C, V \cup V')$ be an election, where C is the set of candidates, $V = \{v_1, v_2, \ldots, v_n\}$ is the collection of voters who already cast their votes, $V' = \{v_{n+1}, v_{n+2}, \ldots, v_{n+m}\}$ is the collection of manipulators, let $c \in C$ be the distinguished candidate, and let w_i be the weight assigned to voter $v_i \in V \cup V'$ with $1 \leq i \leq n+m$.

If $score_{(C,V)}(c) > score_{(C,V)}(x)$ for each $x \in C$ with $x \neq c$, then each manipulator puts c left of the approval line. This way, no candidate can get closer to c, thus c is the unique SP-AV winner of election $(C, V \cup V')$. Otherwise, there is at least one candidate $x \in C$ with $score_{(C,V)}(c) \leq score_{(C,V)}(x)$. Let x be a candidate with highest score in (C,V). If

$$score_{(C,V)}(x) - score_{(C,V)}(c) \geq \sum_{i=n+1}^{m} w_i,$$

then it is impossible for the manipulators to make candidate c the unique SP-AV winner, since c can gain on at most $\sum_{i=n+1}^{m} w_i$ points. Otherwise, the manipulators' votes have the following form:

$$c \mid (C - \{c\}).$$

Clearly, c gains on $\sum_{i=n+1}^{m} w_i$ points, all the other candidates get no points at all. Thus, c is then the unique SP-AV winner.

The destructive cases goes analogously with the difference that if it is possible to make c not a unique SP-AV winner, then the manipulators cast their votes in the following way:

$$(C - \{c\}) \mid c.$$

Clearly, the algorithm runs in polynomial time and it again not only solves the decision problem for constructive weighted manipulation but also determines how to successfully manipulate if that is possible.

Since SP-AV-manipulation is the single weight special case of SP-AV-weighted-manipulation, it follows that SP-AV-manipulation is also in P. ❑

4.4 Conclusions and Open Questions

We have shown that Brams and Sanver's SP-AV system [BS06] is resistant to 19 of the 22 previously studied types of control. On the one hand, like Copeland voting [FHHR08b], SP-AV is fully resistant to constructive control, yet unlike Copeland it additionally is broadly resistant to destructive control. On the other hand, like plurality [BTT92, HHR07a], SP-AV is fully resistant to candidate control, yet unlike plurality it additionally is broadly resistant to voter control. Thus, for these 22 types of control, SP-AV has more resistances, by three, and fewer vulnerabilities to control than is currently known for any other natural voting system with a polynomial-time winner problem.

We have also shown that both approval voting and SP-AV are vulnerable to destructive bribery, even when weights and prices are assigned to the voters.

An interesting open direction is to investigate control in SP-AV or even in any other voting system in a parameterized point of view. That is, for example, fix the number of candidates allowed to delete, is the resulting problem fixed-parameter tractable or intractable?

The main open question is still if there is a natural voting system with an easy winner-determination procedure which is resistant against any control action. In the following chapter we will investigate fallback voting with respect to our 22 control actions.

Chapter 5

Control in Fallback Voting

This chapter considers in detail the second voting system introduced by Brams and Sanver [BS06], that combines approval and preference-based voting, with respect to our 22 control actions.

5.1 Definitions and Conventions

We start with the definition of fallback voting.

Definition 5.1 ([BS06])**.** *Let (C,V) be an election, where the voters both indicate approvals/disapprovals of the candidates. For each voter $v \in V$, an* AV strategy *of v is defined analogously to SP-AV as a subset $S_v \subseteq C$ such that v approves of all candidates in S_v and disapproves of all candidates in $C - S_v$. Each voter $v \in V$ also provides a tie-free linear ordering of all candidates in S_v. The list of AV strategies for all voters in V is again called an* AV strategy profile *for (C,V).*

For each $c \in C$, let $score_{(C,V)}(c) = \|\{v \in V \mid c \in S_v\}\|$ denote the number of c's approvals, and let $score^i_{(C,V)}(c)$ be the level i score *of c which is the number of c's approvals ranked on i^{th} position or higher.*

Winner determination:

1. *On the first level, only the highest ranked approved candidates are considered. If there is a candidate $c \in C$ who has strict majority on this level, then c is the unique level 1 FV winner of the election.*

2. *If there is no level 1 winner, the two highest ranked approved candidates are considered in each approval strategy. If there is exactly one candidate $c \in C$ who has strict majority on this level, then c is the unique level 2 FV winner of the election.*

If there are at least two candidates with strict majority, then a level 2 FV winner is a candidate with the highest level 2 score.

3. *If there is no level 1 or level 2 winner, the voters descend level by level to lower levels until there is at least one candidate who was approved by the strict majority of the voters. Denote this level by level i. If there is only one such candidate, he or she is the unique level i FV winner of the election. If there are more than one candidates with strict majority, then a candidate with the highest level i score is a level i FV winner of the election.*

4. *Otherwise, every candidate with the highest* $score_{(C,V)}(c)$ *is a winner of election* (C,V).

Note that a level 1 FV winner is always the unique FV winner of an election. In contrast to sincere-strategy preference-based approval voting, in fallback voting, regardless of the control action, no changes have to be made on the ballots. Analogously to SP-AV, we also require sincerity in fallback voting, but unlike SP-AV, fallback voting allows for a voter to have an empty approval strategy, i.e., $S_v = \emptyset$ or even a complete approval strategy, thus, $S_v = C$, i.e., admissibility is not required.

We will represent votes in fallback voting like in SP-AV with the difference that all candidates right from the approval line are represented as a set without ranking (e.g., "$a\ \ b\ \mid\ \{c,d\}$" means that a and b are approved of and a is ranked first place, b is ranked second, whereas c and d are both disapproved of, and they are not ranked).

Fallback voting, like SP-AV, does not satisfy Unique-WARP.

Proposition 5.1. *Fallback voting does not satisfy Unique-WARP.*

Proof. Consider the election (C,V) with candidate set $C = \{a,b,c,d\}$ and voter collection $V = \{v_1, v_2, \ldots, v_6\}$:

$$\begin{array}{rl} v_1 = v_2 = v_3: & a\ \ c\ \mid\ \{b,\ d\} \\ v_4 = v_5: & b\ \ d\ \ c\ \mid\ \{a\} \\ v_6: & d\ \ a\ \ c\ \mid\ \{b\} \end{array}$$

There is no level 1 FV winner, and the unique level 2 FV winner of the election (C,V) is candidate a with $score^2_{(C,V)}(a) = 4$. By removing candidate b from the election, we get the subelection (C',V) with $C' = \{a,c,d\}$. There is again no level 1 FV winner. However, there are two candidates on the second level with strict majority, namely candidate a and c. Since $score^2_{(C',V)}(c) = 5$ is higher than $score^2_{(C',V)}(a) = 4$, the unique level 2 FV winner of the subelection (C',V) is candidate c. Thus, FV does not satisfy Unique-WARP. ❑

Control by	SP-AV		FV		AV	
	Constr.	Destr.	Constr.	Destr.	Constr.	Destr.
Adding an Unlimited Number of Candidates	R	R	**R**	**R**	I	V
Adding a Limited Number of Candidates	R	R	**R**	**R**	I	V
Deleting Candidates	R	R	**R**	**R**	V	I
Partition of Candidates	TP: R TP: R	TP: R TP: R	**TE: R** **TE: R**	**TE: R** **TE: R**	TE: V TP: I	TE: I TP: I
Run-off Partition of Candidates	TP: R TP: R	TP: R TP: R	**TE: R** **TE: R**	**TE: R** **TE: R**	TE: V TP: I	TE: I TP: I
Adding Voters	R	V	**S**	**S**	R	V
Deleting Voters	R	V	**S**	**S**	R	V
Partition of Voters	TE: R TP: R	TE: V TP: R	**TE: S** **TE: S**	**TE: S** **TE: S**	TE: R TP: R	TE: V TP: V

Table 5.1: Overview of fallback voting results. Key: I stands again for immune, S for susceptible, R for resistant, V for vulnerable, TE means ties-eliminate, and TP means ties-promote. For entries with “S” it is open whether resistance or vulnerability holds.

5.2 Results for Fallback Voting

5.2.1 Overview

Theorem 5.1 and Table 5.1 show the results regarding control of elections for fallback voting.

Theorem 5.1. *Fallback voting is resistant, vulnerable, and susceptible to the* 22 *types of control defined in Section 3.3 as shown in Table 5.1.*

5.2.2 Susceptibility

Again, we start with the susceptibility statements for candidate control.

Lemma 5.1. *Fallback voting is susceptible to constructive and destructive control by adding candidates (in both the “limited” and “unlimited” cases), by deleting candidates, and by partition of candidates (with or without run-off and for each in both models TE and TP).*

Proof. From Theorem 3.3 and the fact that FV is a voiced voting system, follows that FV is susceptible to constructive control by deleting candidates, and to destructive control by adding candidates (in both the “limited” and “unlimited” cases).

Now, consider the election (C,V) given in Proposition 5.1. The unique FV winner of the election is candidate a. Partition the set of candidates as follows: let $C_1 = \{a,c,d\}$

and $C_2 = \{b\}$. The unique FV winner of subelection (C_1,V) is candidate c as shown in Proposition 5.1. In both partition and run-off partition of candidates and for each in both models TE and TP, candidate b runs against candidate c in the final stage of the election. The unique FV winner is in each case candidate c. Thus, FV is susceptible to destructive control by partition of candidates (with or without run-off and for each in both models TE and TP). By Theorem 3.2 FV is also susceptible to destructive control by deleting candidates. By Theorem 3.1 FV is also susceptible to constructive control by adding candidates (in both the "limited" and "unlimited" cases).

In the following, let's change the roles of a and c, with candidate c being our distinguished candidate. In election (C,V), c loses against candidate a. By partitioning the candidates as described above, c gets the unique FV winner of the election. Thus, FV is susceptible to constructive control by partition of candidates (with or without run-off and for each in both models TE and TP). ❑

We now turn to susceptibility to voter control.

Lemma 5.2. *Fallback voting is susceptible to constructive and destructive control by adding voters, by deleting voters, and by partition of voters (in both models TE and TP).*

Proof. Consider the election (C,V), where $C = \{a,b,c,d\}$ is the set of candidates and $V = \{v_1,v_2,v_3,v_4\}$ is the collection of voters. Partition V into $V_1 = \{v_1,v_2\}$ and $V_2 = V - V_1$. Thus, we get:

	(C,V)	into	(C,V_1)	and	(C,V_2)
$v_1:$	$a\ c\ \mid \{b,d\}$		$a\ c\ \mid \{b,d\}$		
$v_2:$	$d\ c\ \mid \{a,b\}$		$d\ c\ \mid \{a,b\}$		
$v_3:$	$b\ a\ c\ \mid \{d\}$				$b\ a\ c\ \mid \{d\}$
$v_4:$	$b\ a\ \mid \{c,d\}$				$b\ a\ \mid \{c,d\}$

Clearly, candidate a is the unique level 2 FV winner of (C,V). However, c is the unique level 2 FV winner of (C,V_1) and b is the unique level 1 FV winner of (C,V_2), and so a is not promoted to the final stage. Thus, FV is susceptible to destructive control by partition of candidates in both models TE and TP. By Theorem 3.3 and the fact that FV is a voiced voting system, FV is susceptible to destructive control by deleting voters. By Theorem 3.1, FV is also susceptible to constructive control by adding voters.

By changing the roles of a and c again, we can see that FV is susceptible to constructive control by partition of voters in both models TE and TP. By Theorem 3.2 FV is also susceptible to constructive control by deleting voters. And finally, again by Theorem 3.1, FV is susceptible to destructive control by adding voters. ❑

5.2.3 Candidate Control

In the case of fallback voting, we cannot use the constructions presented in [HHR07a] and in Section 4.2.3 for the resistance proofs, but we either have to introduce new constructions, or, in some cases, significantly modify Construction 4.3.

All resistance results in this section follow via reduction from the NP-complete problem Hitting Set defined in Section 2.2.

Theorem 5.2. *Fallback voting is resistant to all types of constructive and destructive candidate control defined in Section 3.3 except for constructive control by deleting candidates.*

Proof. We start with the construction for these thirteen control scenarios.

Construction 5.3. *Let $(B, \mathscr{S}, k)$ be a given instance of Hitting Set, where $B = \{b_1, b_2, \ldots, b_m\}$ is a set, $\mathscr{S} = \{S_1, S_2, \ldots, S_n\}$, $n > 1$, is a collection of subsets $S_i \subseteq B$, and $k < m$ is a positive integer. Define the election (C,V), where $C = B \cup \{c,d,w\}$ is the candidate set and where V consists of the following $6n(k+1)+4m+11$ voters:*

1. *There are $2m+1$ voters of the form:*

$$c \ \mid \ B \cup \{d,w\}.$$

2. *There are $2n+2k(n-1)+3$ voters of the form:*

$$c \ \ w \ \mid \ B \cup \{d\}.$$

3. *There are $2n(k+1)+5$ voters of the form:*

$$w \ \ c \ \mid \ B \cup \{d\}.$$

4. *For each i, $1 \leq i \leq n$, there are $2(k+1)$ voters of the form:*

$$d \ \ S_i \ \ c \ \mid \ (B - S_i) \cup \{w\}.$$

5. *For each j, $1 \leq j \leq m$, there are two voters of the form:*

$$d \ \ b_j \ \ w \ \mid \ (B - \{b_j\}) \cup \{c\}.$$

6. *There are $2(k+1)$ voters of the form:*

$$d \ \ w \ \ c \ \mid \ B.$$

Since there is no level 1 FV winner in election $(\{c,d,w\},V)$, and

$$\begin{aligned} score^2_{(\{c,d,w\},V)}(c) &= 2(m-k)+6n(k+1)+9, \\ score^2_{(\{c,d,w\},V)}(d) &= 2n(k+1)+2(m+k+1), \text{ and} \\ score^2_{(\{c,d,w\},V)}(w) &= 4n(k+1)+2m+10, \end{aligned}$$

c is the unique level 2 FV winner of the election $(\{c,d,w\},V)$.

The crux of the proofs for the five control problems is the following proposition.

Proposition 5.2. *1. If $\mathscr{S}$ has a hitting set B' of size k, then w is the unique FV winner of election $(B' \cup \{c,d,w\},V)$.*

2. Let $D \subseteq B \cup \{d,w\}$. If c is not the unique FV winner of election $(D \cup \{c\},V)$, then there exists a set $B' \subseteq B$ such that

(a) $D = B' \cup \{d,w\}$,

(b) w is the unique level 2 FV winner of election $(B' \cup \{c,d,w\},V)$, and

(c) B' is a hitting set of $\mathscr{S}$ of size less than or equal to k.

Proof.

1. Suppose that B' is a hitting set of $\mathscr{S}$ of size k. Then there is no level 1 FV winner in election $(B' \cup \{c,d,w\},V)$, and we have the following level 2 scores:

$$\begin{aligned} score^2_{(B' \cup \{c,d,w\},V)}(c) &= 4n(k+1)+2(m-k)+9, \\ score^2_{(B' \cup \{c,d,w\},V)}(d) &= 2n(k+1)+2(m+k+1), \\ score^2_{(B' \cup \{c,d,w\},V)}(w) &= 4n(k+1)+2(m-k)+10, \\ score^2_{(B' \cup \{c,d,w\},V)}(b_j) &\leq 2n(k+1)+2 \qquad \text{for each } b_j \in B'. \end{aligned}$$

   Thus, w is the unique level 2 FV winner of election $(B' \cup \{c,d,w\},V)$.

2. Let $D \subseteq B \cup \{d,w\}$. Suppose c is not the unique FV winner of election $(D \cup \{c\},V)$.

   (a) Other than c, only w is approved by a majority of voters. Thus, if c is not a unique FV winner of the election $(D \cup \{c\},V)$, then clearly $w \in D$. In any election $(D \cup \{c\},V)$, candidate w has no level 1 strict majority, and candidate c has not later than on level 2 strict majority. Thus, w has to tie or beat c on level 2. For a contradiction, suppose $d \notin D$. Then, $score^2_{(D \cup \{c\},V)}(c) \geq 4n(k+1)+2m+11$. The overall score of w is $score_{(D \cup \{c\},V)}(w) = 4n(k+1)+2m+10$, which contradicts our assumption, that w ties or beats c on level 2. Thus, $D = B' \cup \{d,w\}$, where $B' \subseteq B$.

(b) This part follows immediately from part (a).

(c) Let ℓ be the number of sets in $\mathscr{S}$ not hit by B', then:

$$\begin{aligned} score^2_{(B'\cup\{c,w\},V)}(w) &= 4n(k+1)+10+2(m-\|B'\|), \\ score^2_{(B'\cup\{c,w\},V)}(c) &= 2(m-k)+4n(k+1)+9+2(k+1)\ell. \end{aligned}$$

From part (a) we know, that $score_{(B'\cup\{c,w\},V)}(w) \geq score_{(B'\cup\{c,w\},V)}(c)$, i.e.,

$$4n(k+1)+10+2(m-\|B'\|) \geq 2(m-k)+4n(k+1)+9+2(k+1)\ell.$$

Since all variables are integers, the above inequality implies

$$0 \geq \|B'\| - k + (k+1)\ell$$

thus, $\ell = 0$, and so B' is a hitting set of $\mathscr{S}$ of size less than or equal to k.

❑

The following corollaries can be proven analogously as the corresponding corollaries in Section 4.2.3. For the sake of completeness we will present the proofs for each corollary.

Corollary 5.1. *Fallback voting is resistant to constructive and destructive control by adding candidates (both in the limited and the unlimited version of the problem).*

Proof. This corollary follows immediately from Proposition 5.2, via mapping the Hitting Set instance $(B,\mathscr{S},k)$ to the set $\{c,d,w\}$ of qualified candidates and the set B of spoiler candidates, to the voter collection V, and by having c be the designated candidate in the destructive case, and by having w be the designated candidate in the constructive case. ❑

Corollary 5.2. *Fallback voting is resistant to destructive control by deleting candidates.*

Proof. Let the election (C,V) be given as in Construction 5.3 with distinguished candidate c. We claim that $\mathscr{S}$ has a hitting set of size at most k, if and only if c can prevented from being a unique FV winner by deleting at most $m-k$ candidates.

From left to right: Suppose $\mathscr{S}$ has a hitting set B' of size k. Delete the $m-k$ candidates $B-B'$. Now, both candidates c and w have strict majority on level 2, but $score_{(\{c,d,w\}\cup B',V)}(c) = 4n(k+1)+2(m-k)+9$ and $score_{(\{c,d,w\}\cup B',V)}(w) = 4n(k+1)+2(m-k)+10$, thus w is the unique level 2 FV winner.

From right to left: Suppose that c can be prevented from being a unique FV winner by deleting at most $m-k$ candidates. Let $D \subseteq B\cup\{d,w\}$ be the set of deleted candidates, such that $c \notin D$. It immediately follows from Proposition 5.2, that $C-D-\{c\} = B'\cup\{d,w\}$, where B' is a hitting set of $\mathscr{S}$ of size at most k. ❑

Corollary 5.3. *Fallback voting is resistant to constructive and destructive control by partition of candidates and run-off partition of candidates (for each in both models TE and TP).*

Proof. We only prove the constructive case. For the destructive case we simply change the roles of c and w. Let the election (C,V) be given as in Construction 5.3 with distinguished candidate w. We claim that $\mathscr{S}$ has a hitting set of size at most k, if and only if w can be made the unique FV winner by exerting control by partition or run-off partition of candidates (for each in both models TE and TP).

From left to right: Suppose, $\mathscr{S}$ has a hitting set $B' \subseteq B$ of size k. Partition the set of candidates into the two subsets $C_1 = B' \cup \{c,d,w\}$ and $C_2 = C - C_1$. According to Proposition 5.2, w is the unique level 2 FV winner of the subelection $(B' \cup \{c,d,w\},V)$. Then, the score of w in the final stage is at least $2(m-k)+4n(k+1)+9$, and the opponents of w in the final stage are only candidates from B and their score is at most $2n(k+1)+2$. Thus, w is the only candidate in the final stage with strict majority so w is the unique FV winner of the resulting election.

From right to left: Suppose, there exists a partition such that w is the unique FV winner of the election. In this case, c is not an FV winner of the election. Then, there has to be a subset $D \subseteq B \cup \{d,w\}$ of candidates such that c is not a unique FV winner of the election $(D \cup \{c\},V)$. Due to Proposition 5.2, there exists a size k hitting set of $\mathscr{S}$. ❑

Construction 5.3 does not work for constructive control by deleting candidates in fallback voting for the same reason as why Construction 4.3 doesn't work for constructive control by deleting candidates in SP-AV, namely, by deleting c the chair could make w the unique FV winner, regardless of whether or not $\mathscr{S}$ has a hitting set of size k. The following theorem provides a construction with which we are able to show resistance in this case too.

Theorem 5.4. *Fallback voting is resistant to constructive control by deleting candidates.*

Proof. Susceptibility holds by Lemma 5.1. Let $(B,\mathscr{S},k)$ be a Hitting Set instance with $B = \{b_1,b_2,\dots,b_m\}$, and a collection $\mathscr{S} = \{S_1,S_2,\dots,S_n\}$ of subsets $S_i \subseteq B$, and a positive integer $k \leq m$.

Define the election (C,V), where $C = B \cup C' \cup D \cup E \cup \{w\}$ is the set of candidates, where $C' = \{c_1,c_2,\dots,c_{k+1}\}$, $D = \{d_1,d_2,\dots,d_p\}$, $E = \{e_1,e_2,\dots,e_n\}$, and w is the distinguished candidate. The number of candidates in D is $p = \sum_{i=1}^{n}(n+k-\|S_i\|) = n^2+kn-\sum_{i=1}^{n}\|S_i\|$. Then, V is the following collection of $2(n+k+1)+1$ voters:

1. For each i, $1 \leq i \leq n$, there is one voter of the form:

$$S_i \;\; D_i \;\; w \;\mid\; (B-S_i) \cup C' \cup E,$$

where $D_i = \{d_{(i-1)(n+k)-\sum_{j=1}^{i-1}(\|S_j\|)+1},\dots,d_{i(n+k)-\sum_{j=1}^{i}(\|S_j\|)}\}$.

2. For each j, $1 \leq j \leq k+1$, there is one voter of the form:

$$E \;\; C' - \{c_j\} \;\; c_j \;\mid\; B \cup D \cup \{w\}.$$

3. There are $k+1$ voters of the form:

$$w \;\mid\; B \cup C' \cup D \cup E.$$

4. There are n voters of the form:

$$C' \;\mid\; B \cup D \cup E \cup \{w\}.$$

5. There is one voter of the form:

$$C' \;\; w \;\mid\; B \cup D \cup E.$$

Note that there is no unique FV winner in the above election, the candidates in C' and w are all level $n+k+1$ winners.

We claim that $\mathscr{S}$ has a hitting set of size k for B if and only if w can be made the unique FV winner by deleting at most k candidates.

From left to right: Suppose $\mathscr{S}$ has a hitting set B' of size k. Delete the correspondig candidates. Now, candidate w is the unique level $(n+k)$ FV winner of the resulting election.

From right to left: Suppose w can be made the unique FV winner by deleting at most k candidates. Since in election (C,V) there were $k+1$ candidates other than w with strict majority on level $n+k+1$, by deleting k candidates, there is still at least one candidate other than w with strict majority on level $n+k+1$. Thus, w must be a unique FV winner on a level lower or equal than $n+k$. This is only possible, if in all n votes in the first voter group w moved forward at least one place. This, on the other hand, is only possible if $\mathscr{S}$ has a hitting set B' of size k. ❑

5.3 Conclusions and Open Questions

We have shown that Brams and Sanver's fallback voting system [BS06] is, like plurality, fully resistant to candidate control. The 8 voter control cases are all susceptible but it still remains open whether FV is resistant or vulnerable to these control actions.

Also the question whether fallback voting is still intractable under parameterized complexity theoretic aspects for these control actions is still open.

Chapter 6

Optimal Lobbying

In the American political system citizens and corporations are allowed to make contributions to representatives. We can make the assumption that a politician who accepts such a donation will vote according to the wishes of the contributor (if the donation is big enough). Christian et al. [CFRS06] introduced and studied the problem Optimal-Lobbying under parameterized complexity theoretic aspects. We investigate in this chapter the more general weighted version of Optimal-Lobbying, Optimal-Weighted-Lobbying. We describe a greedy algorithm for Optimal-Weighted-Lobbying and we determin the approximation ratio of our algorithm.

6.1 Framework

Suppose there are *m* voters who vote on *n* referenda, and there is an external actor, which is referred to as "The Lobby" and seeks to influence the outcome of these referenda by making voters change their votes. It is again assumed that The Lobby has complete information about the voters' original votes, and that The Lobby's budget allows for influencing the votes of a certain number, say *k*, of voters. Formally, the Optimal-Lobbying problem as a decision problem is defined as follows [CFRS06]:

Name: Optimal-Lobbying
Instance: An $m \times n$ 0-1 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 jth column has a strict majority of ones (respectively, zeros) if and only if the jth entry of the target vector x of The Lobby is one (respectively zero)?

Optimal-Lobbying can be rephrased as a parameterized problem, where the parameter is the number k of voters to be influenced.

Theorem 6.1 ([CFRS06])**.** *Optimal-Lobbying is $W[2]$-complete.*

Proof Sketch. We just give a sketch of proof in this place, for the complete proof we refer to the original paper from Christian et al. [CFRS06]. One of the basic techniques to prove $W[2]$-completeness is to show in the first step the $W[2]$-hardness of the problem via reduction from a known $W[2]$-hard problem, in the second step to prove membership to $W[2]$ via reduction from the underlying problem to a problem known to be in $W[2]$.

To show $W[2]$-hardness, Christian et al. reduced from the $W[2]$-complete problem k-Dominating Set. Let (G,k) be a given instance of k-Dominating Set, where $G = (V,E)$ is a graph with the set of vertices $V = \{v_1, v_2, \ldots, v_n\}$, and the set of edges E. Furthermore, let n be odd and $mindeg(G) \geq k$. Note, that k-Dominating Set remains $W[2]$-complete under the above restrictions [CFRS06].

Define the lobbying matrix L as follows. L has $2n-2k+1$ rows, where the top n rows are labeled $v_1, v_2, \ldots, v_n$. L has $n+1$ columns labeled as $r_t, r_1, r_2, \ldots, r_n$. The first column, r_t, has only zeros in the top n rows and ones in the bottom $n-2k+1$ rows, thus there are $2k-1$ more zeros than ones in r_t. In any column r_i, where $i = 1,2,\ldots,n$, in the top n columns there is a zero in a row v_j if and only if $v_j \in N[v_i]$. Initially, in the bottom $n-2k+1$ rows every entry is a one. For each column r_i, where $i = 1,2,\ldots,n$, arbitrarily flip $n-k-\|N[r_i]\|+1$ entries from one to zero. Note, that now in all these n columns there are one more zeros than ones. Let the target vector be $x = 1^{n+1}$, and let the parameter be k.

Now, (G,k) is a yes-instance of k-Dominating Set if and only if (L,x,k) is a yes-instance of Optimal-Lobbying.

From left to right: Suppose that G has a k-dominating set. Choose the corresponding k rows in the top n rows in L. In the first column, r_t flip all the k entries from zero to one, then there are exactly one more ones than zeros in r_t. Since the selected k rows correspond to a dominating set, for each of the remaining n columns we can flip at least one zero to a one. Thus, there is a majority of ones in each column.

From right to left: Suppose that we can choose k rows in L, such that flipping some entries in these rows will lead to a majority of ones in each column. Since initially there were $2k-1$ more zeros than ones in r_t, the k chosen rows must have been among the top n rows. On the other hand, in each of the n remaining columns at least one entry was flipped from zero to one, that is, each of the vertices were neighbours of at least one vertice out of the k chosen ones, thus these k vertices correspond to a dominating set. Thus, Optimal-Lobbying is $W[2]$-hard.

We are not going to show here the membership of Optimal-Lobbying in the class $W[2]$, we point the reader to the proof in [CFRS06], where the reduction goes onto the $W[2]$-complete problem Independent-k-Dominating Set.

This result is considered strong evidence that Optimal-Lobbying is intractable, even for small values of the parameter k. However, even though the optimal goal of The Lobby cannot be achieved efficiently, it might be approximable within some factor. That is, given an $m \times n$ 0-1 matrix L and a target vector $x \in \{0,1\}^n$, The Lobby might try to reach its target by changing the votes of as few voters as possible.

We consider the more general problem Optimal-Weighted-Lobbying, where we assume that influencing the 0-1 vector of each voter v_i exacts some price, $price(v_i) \in \mathbb{Q}_0^+$. In this scenario, The Lobby seeks to minimize the amount of money spent to reach its goal. The formal definition of the minimization problem is:

Name: Optimal-Weighted-Lobbying

Instance: An $m \times n$ 0-1 matrix L (whose rows represent the voters, whose columns represent the referenda, and whose 0-1 entries represent No/Yes votes), there is a cost function c that maps from the set of rows to the rational numbers with $c(v_i) = price(v_i)$, and a target vector $x \in \{0,1\}^n$.

Question: Find a minimum cost subset of the set of rows 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 jth column has a strict majority of ones (respectively, zeros) if and only if the jth entry of the target vector x of The Lobby is one (respectively zero).

The problem Optimal-Lobbying is the unit-prices special case of Optimal-Weighted-Lobbying, i.e., where $price(v_i) = 1$ for each voter v_i. It follows that Optimal-Weighted-Lobbying (redefined as a parameterized rather than an optimization problem, where the parameter is The Lobby's budget of money to be spent) inherits the W[2]-hardness lower bound from its special case Optimal-Lobbying.

Proposition 6.1. *Optimal-Weighted-Lobbying is W[2]-hard.*

In the following section, we describe and analyze an efficient greedy algorithm for approximating Optimal-Weighted-Lobbying.

6.2 Results

Let a matrix $V \in \{0,1\}^{m\times n}$ be given, where the columns $r_1, r_2, \ldots, r_n$ of L represent the referenda and the rows $v_1, v_2, \ldots, v_m$ of L represent the voters. Without loss of generality, we may assume that The Lobby's target vector is of the form $x = 1^n$ (and thus may be dropped from the problem instance), since if there is a zero in x at position j, we can simply flip this zero to one and also flip the corresponding zeros and ones in column r_j.

For each column r_j, define the *deficit* d_j to be the minimum number of zeros that need to be flipped to ones such that there are strictly more ones than zeros in this column. Let $D_0 = \sum_{j=1}^{n} d_j$ be the sum of all initial deficits.

Figure 6.1 gives the greedy algorithm [1], which proceeds by iteratively choosing a most "cost-effective" row of L and flipping to ones all those zeros in this row that belong to columns with a positive deficit, until the deficits in all columns have decreased to zero. We assume that ties between rows with equally good cost-effectiveness are broken in any simple way, e.g., in favor of the tied v_i with lowest i.

Let R be the set of columns of L whose deficits have already vanished at the beginning of an iteration, i.e., all columns in R already have a strict majority of ones. Let $v_{i\rceil R^c}$ denote the entries of v_i restricted to those columns not in R, and let $\#_0(v_{i\rceil R^c})$ denote the number of zeros in $v_{i\rceil R^c}$. (For i such that $\#_0(v_{i\rceil R^c}) = 0$, we consider $\mathit{price}(v_i)/\#_0(v_{i\rceil R^c})$ to be $+\infty$.) During an iteration, the *cost per flipped entry in row v_i* (for decreasing the deficits in new columns by flipping v_i's zeros to ones) is $\mathit{price}(v_i)/\#_0(v_{i\rceil R^c})$. We say a voter v_i is *more cost-effective* than a voter v_j if v_i's cost per flipped entry is less than v_j's. When our algorithm chooses to alter a row v_i, we will think of its price being distributed equally among the new columns with decreased deficit, and at that instant will permanently associate with every flipped entry, e_k, in that row its portion of the cost, i.e., $\mathit{cost}(e_k) = \mathit{price}(v_i)/\#_0(v_{i\rceil R^c})$.

Clearly, the greedy algorithm in Figure 6.1 always stops, and its running time is polynomial, since the while loop requires only linear (in the input size) time and has to be executed at most $D_0 = \sum_{j=1}^{n} d_j \leq n \cdot \lceil (m+1)/2 \rceil$ times (note that at most $\lceil (m+1)/2 \rceil$ flips are needed in each column to achieve victory for The Lobby's position).

Now, enumerate the D_0 entries of L that have been flipped in the order in which they were flipped by the algorithm. Let $e_1, e_2, \ldots, e_{D_0}$ be the resulting enumeration. Let OPT be the money that would be spent by The Lobby for an optimal choice of voters such that its target is reached.

Lemma 6.1. *For each $k \in \{1, 2, \ldots, D_0\}$, we have $\mathit{cost}(e_k) \leq \mathrm{OPT}/(D_0 - k + 1)$.*

[1] The special case of this algorithm, where each column has to be "covered" only once is the same as the classical greedy algorithm for Weighted Set Covering, see, e.g., [Vaz03].

1. **Input:** A matrix $L \in \{0,1\}^{m \times n}$.
2. **Initialize:**

   Compute the deficits d_j, $1 \le j \le n$.

   $D \leftarrow \sum_{j=1}^{n} d_j$. $/*$ Initially, $D = D_0$. $*/$

   $X \leftarrow \emptyset$.
3. **While** $D \neq 0$ **do**

   Let R be the set of columns r_j with $d_j = 0$.

   Find a voter whose cost-effectiveness is greatest, say v_i.

   Let $\gamma_i = price(v_i)/\#_0(v_{i\rceil R^c})$.

   Choose v_i and flip all zeros in $v_{i\rceil R^c}$ to ones.

   For each flipped entry e in v_i, let $cost(e) = \gamma_i$.

   $/*$ $cost(e)$ will be used in our analysis. $*/$

   $X \leftarrow X \cup \{i\}$.

   $d_j \leftarrow d_j - 1$, for each column r_j for which a zero was flipped.

   $D \leftarrow \sum_{j=1}^{n} d_j$.
4. **Output:** X.

Figure 6.1: Greedy algorithm for Optimal-Weighted-Lobbying

Proof. Let I denote a voter set that realizes the optimal expenditure, OPT, for reducing the deficit to zero. Now, our analysis will follow the structure of the while loop of the algorithm. So consider the algorithm at some point where the current deficit, D, is strictly greater than zero and we are starting a pass through the while loop. So the entry we will next flip will be named e_{D_0-D+1}.

If we were to at this point consider changing all the rows associated with I to all ones, this certainly would reduce the deficit to zero in the current matrix, as it in fact would even reduce the deficit to zero in the original matrix, and any prior passes through the while loop never flipped any entry in a way that went against the Lobby's goal (increased any deficit). Now, the first important thing to note is that there must be a collection A of exactly D zeros in the rows associated with I such that flipping just those zeros reduces the deficit in the current matrix to zero. (This is clearly true due to the way deficits are computed and the separateness of the columns and their deficits.)

So by buying the rows of I at cost OPT we certainly can flip all the D entries composing A, i.e., we could image the cost as being distributed equally, and so we could view each such flipped entry as being purchased at cost OPT/D. However, the second important thing to note is that this means there is some element i of I that contains at least one element of A such that for that element, at this moment, $price(v_i)/\#_0(v_{i\rceil R^c})$ is at most OPT/D.[2] Since our algorithm chooses the most cost-effective row, it will choose a row with at least this cost-effectiveness.

Thus the first element of this iteration through the while loop, which will be e_{D_0-D+1}, is bought at cost at most OPT/D. So, for $k = D_0 - D + 1$ the claim of this lemma is satisfied, since $D_0 - (D_0 - D + 1) + 1 = D$.

However, note that each additional entry that we flip during this same pass through the while loop (e.g., some possibly empty prefix of e_{D_0-D+2}, e_{D_0-D+3}, etc.) will not only satisfy the claim of this lemma, but also will do even better, as it is (reducing the deficit by one and is) being bought at the cost of OPT/D, and the claim of the lemma was merely requiring that these additional elements be bought for, respectively, the strictly higher costs $\mathrm{OPT}/(D-1)$, $\mathrm{OPT}/(D-2)$, etc.

So, for each pass through the while loop, each entry flipped meets or beats the cost bound stated in this lemma. ❑

Theorem 6.2. *The greedy algorithm presented in Figure 6.1 approximates the problem Optimal-Weighted-Lobbying with approximation ratio at most*

$$\sum_{i=1}^{D_0} \frac{1}{i} \leq 1 + \ln D_0 \leq 1 + \ln\left(n \left\lceil \frac{m+1}{2} \right\rceil\right).$$

Proof. The total price of the set of voters X picked by the greedy algorithm is the sum of the costs of those entries flipped. That is, $price(X) = \sum_{i \in X} price(v_i) = \sum_{k=1}^{D_0} cost(e_k) \leq \left(1 + \frac{1}{2} + \cdots + \frac{1}{D_0}\right) \cdot \mathrm{OPT}$, where the last inequality follows from Lemma 6.1. ❑

[2] The reason is as follows. Consider for the moment associating with each element of each row of I that contains at least one element of A the portion of OPT indicated by the row's price divided by the number of elements of A in that row. Clearly, the average of those D element costs trivially must equal OPT/D, the overall average cost. So since the weighted average equals OPT/D, then at least one of the values being averaged must be less than or equal to OPT/D—and let us suppose that value is associated with row $i' \in I$. Since A is a set of D items that (starting from the current matrix) reduces the deficit by D, every element of A in row i' must reduce the deficit associated with its column by 1. And so every member of A in row i' must be a member of $v_{i'\rceil R^c}$. So the value associated with each member of A in row i', which we already argued is at most OPT/D, is greater than or equal to the value $price(v_{i'})/\#_0(v_{i'\rceil R^c})$ that the algorithm computes for each element of A—and indeed each element of $v_{i'\rceil R^c}$—in that row.

	r_1	r_2	r_3	$\cdots$	r_n	$price(v_i)$
v_1	0	1	1	$\cdots$	1	1
v_2	1	0	1	$\cdots$	1	$1/2$
v_3	1	1	0	$\cdots$	1	$1/3$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
v_n	1	1	1	$\cdots$	0	$1/n$
v_{n+1}	0	0	0	$\cdots$	0	$1+\varepsilon$
v_{n+2}	1	0	0	$\cdots$	0	2
v_{n+3}	0	1	0	$\cdots$	0	2
v_{n+4}	0	0	1	$\cdots$	0	2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
v_{2n+1}	0	0	0	$\cdots$	1	2

Table 6.1: A tight example for the greedy algorithm in Figure 6.1

Since the input size is lower-bounded by $m \cdot n$, Theorem 6.2 establishes a logarithmic approximation ratio for Optimal-Weighted-Lobbying (and also for Optimal-Lobbying). Note that the proof of Theorem 6.2 establishes an approximation ratio bound that is (sometimes nonstrictly) stronger than $\sum_{i=1}^{D_0} 1/i$. In particular, if the number of zeros flipped in successive iterations of the algorithm's while loop are $\ell_1, \ell_2, \ldots, \ell_p$, where $\ell_1 + \ell_2 + \cdots + \ell_p = D_0$, then the proof gives a bound on the approximation ratio of

$$\frac{\ell_1}{D_0} + \frac{\ell_2}{D_0 - \ell_1} + \cdots + \frac{\ell_p}{D_0 - (\ell_1 + \cdots + \ell_{p-1})} \quad = \quad \sum_{j=1}^{p} \frac{\ell_j}{D_0 - \sum_{k=1}^{j-1} \ell_k}.$$

This is strictly better than $\sum_{i=1}^{D_0} 1/i$ except in the case that each ℓ_j equals 1. And this explains why, in the example we are about to give that shows that the algorithm can at times yield a result with ratio essentially no better than $\sum_{i=1}^{D_0} 1/i$, each ℓ_j will equal 1.

Now, we show that the $\sum_{i=1}^{D_0} 1/i$ approximation ratio stated in Theorem 6.2 is essentially the best possible that can be stated for the greedy algorithm of Figure 6.1. Consider the example given in Table 6.1. The prices for changing the voters' 0-1 vectors are shown in the right-most column of Table 6.1: Set $price(v_i) = 1/i$ for each $i \in \{1, 2, \ldots, n\}$, set $price(v_i) = 2$ for each $i \in \{n+2, n+3, \ldots, 2n+1\}$, and set $price(v_{n+1}) = 1 + \varepsilon$, where $\varepsilon > 0$ is a fixed constant that can be set arbitrarily small. Note that, for each j, $1 \leq j \leq n$, we have $d_j = 1$, and hence $D_0 = n$.

When run on this input, our greedy algorithm sequentially flips, for $i = n, n-1, \ldots, 1$, the single zero-entry of voter v_i to a one. Thus the total money spent is $1 + 1/2 + \cdots +$

$1/n = 1 + 1/2 + \cdots + 1/D_0$. On the other hand, the optimal choice consists of influencing just voter v_{n+1} by flipping all of v_{n+1}'s entries to ones, which costs only $1 + \varepsilon$.

6.3 Combinatorial Reverse Auctions

Combinatorial auctions are economically efficient allocations of wares in multiagent systems. In combinatorial multi-unit auctions a seller would like to sell a set of different items, where he or she offers a different number of units of each item. A group of bidders give bids, specifying how many units they want to buy from each item and which overall price they would pay for it. The seller wants to maximize his revenue.

The opposite situation is, in which a buyer seeks to get a certain ammount of goods on minimal cost, this case is called a combinatorial reverse auction. This setting is not unnatural, it is indeed used in many market scenarios, for example, in procurement. We next give the formal definition stated as a minimization problem.

Name: Multi-Unit Combinatorial Reverse Auction.

Instance: An $m \times n$ matrix L (whose rows represents the sellers, whose columns represent the items), with entries $\lambda_{i,j} \in \mathbb{N}^+$ (representing the number of units of item j the buyer is willing to buy from seller i), where $1 \leq i \leq m$ and $1 \leq j \leq n$, the prices of the bids $(p_1, p_2, \ldots, p_m)$, where $p_i \in \mathbb{Q}_0^+$ for all $1 \leq i \leq m$, and a target vector $x = (x_1, x_2, \ldots, x_n)$, where $x_j \in \mathbb{N}^+$ for all $1 \leq j \leq n$.

Question: Find a minimum price subset of the rows, such that the auctioneer receives all of the units of items he or she is asking (i.e.,

$$min \sum_{i=1}^{m} p_i y_i \text{ such that } \sum_{i=1}^{m} \lambda_{i,j} y_i \geq x_j,$$

where $y_i \in \{0,1\}$ and $1 \leq j \leq n$).

Sandholm et al. [SSGL02] proposed a greedy approximation algorithm for the Multi-Unit Combinatorial Reverse Auctions problem with the same logarithmic approximation ratio as ours. Note, that the Optimal-Weighted-Lobbying problem is a special case of the Multi-Unit Combinatorial Reverse Auctions problem, with each $\lambda_{i,j} \in \{0,1\}$.

6.4 Conclusions and Open Problems

Christian et al. [CFRS06] introduced the optimal lobbying problem and showed it complete for W[2], and so generally viewed as intractable in the sense of parameterized complexity. We proposed an efficient greedy algorithm for approximating the optimal solution of this problem, even if generalized by assigning prices to voters. The greedy algorithm achieves a logarithmic approximation ratio and we prove that that is essentially the best approximation ratio that can be proven for this algorithm. We also show the connection between optimal lobbying and combinatorial reverse auctions introduced by Sandholm et al. [SSGL02].

We mention as an interesting open issue whether more elaborate algorithms can achieve better approximation ratios. Furthermore, it would be interesting to investigate more general models of lobbying, for instance, when voters start with initial probabilities of voting for an issue and each voter has known costs for increasing their probabilities of voting according "The Lobby's" agenda by each of a finite set of increments. Also other evaluation criteria could be considered, such as winning majority, where the average probability of all the voters for an issue is above a predefined threshold.

A further direction for future work could be the investigation of combinatorial auctions and reverse auctions, with respect to parameterized complexity.

Chapter 7

Junta Distributions for SAT

This chapter is motivated by the notion of "frequently self-knowingly correct algorithms," which was proposed by Homan and Hemaspaandra [HHb] in their work on how to frequently find winners of Dodgson elections [Dod76].

7.1 A Motivation: How to Find Dodgson Winners Frequently

Recall the definition of Dodgson's voting system: If (C,V) is an election and c is some designated candidate in C, we call (C,V,c) a *Dodgson triple*. An election is won by those candidates who are "closest" to being a Condorcet winner. More precisely, given a Dodgson election (C,V), every candidate c in C is assigned a score, which is denoted by DodgsonScore(C,V,c), and is defined to be the smallest number of sequential exchanges of adjacent preferences in the voters' preference orders needed to make c a Condorcet winner with respect to the resulting preference orders. Whoever has the lowest Dodgson score wins.

The problem Dodgson-Winner is defined as follows:

Name: Dodgson-Winner.
Instance: An election (C,V) and a designated candidate c in C.
Question: Is c a Dodgson winner in (C,V)?

The search version of this decision problem can easily be stated. As mentioned earlier, Hemaspaandra, Hemaspaandra, and Rothe [HHR97] have shown that determining the Dodgson winner is $\mathrm{P}_{||}^{\mathrm{NP}}$-complete.

It certainly is not desirable to have an election system whose winner problem is hard, as only systems that can be evaluated efficiently are actually used in practice. Fortunately, there are a number of positive results on Dodgson elections and related systems as well. In particular, Bartholdi, Tovey, and Trick [BTT89b] proved that for elections with a bounded number of candidates or voters Dodgson winners are asymptotically easy to determine. Fishburn [Fis77] proposed a "homogeneous" variant of Dodgson elections that Rothe, Spakowski, and Vogel [RSV03] proved to have a polynomial-time winner problem. McCabe-Dansted, Pritchard, and Slinko [MPS] proposed a scheme (called Dodgson Quick) that approximates Dodgson's rule with an exponentially fast convergence. Homan and Hemaspaandra ([HHa], see also McCabe-Dansted, Pritchard, and Slinko [MPS]) proposed a greedy heuristic that finds Dodgson winners with a "guaranteed high frequency of success." To capture a strengthened version of this property formally, they introduced the notion of a "frequently self-knowingly correct algorithm", and they noted [HHa]:

> "The closest related concepts [...] are probably those involving proofs to be verified, such as NP certificates and the proofs in interactive proof systems."

This statement notwithstanding, we show how average-case polynomial time relates to the notion of frequently self-knowingly correct algorithm.

7.2 Average-Case Complexity Theory

To prove a problem NP-hard is a common method of showing its computational intractability as described in Chapter 2. Many NP-hard problems, however, are eminently important in practice, which is why various approaches to coping with NP-hardness have been proposed, including approximation algorithms (see, Chapter 6), parameterized complexity (see, Section 2.3), heuristics/algorithms that are always efficient (i.e., polynomial-time) albeit not always correct (for example, the class P-close [Sch86]), algorithms that are always correct albeit not always efficient (for example, the class APT [MP79]), etc. A particularly interesting approach is to show that some NP-hard problems can be solved efficiently on the average.

The theory of average-case complexity was initiated by Levin [Lev86]. A problem's average-case complexity can be viewed as a more significant measure than its worst-case complexity in many cases, for example in cryptographic applications. We here follow Goldreich's presentation [Gol97]. Another excellent introduction to this theory is that of Wang [Wan97].

Intuitively, Levin observed that many hard problems—including those that are NP-hard in the traditional worst-case complexity model—might nonetheless be easy to solve "on the average," i.e., for "most" inputs or for "most practically relevant" inputs. He proposed to define the complexity of problems with respect to some suitable distribution on the input strings.

We now define the notion of a distributional problem and the complexity class AvgP. In this chapter, we consider two heuristic algorithms: the algorithm Greedy-Winner intends to solve the decision problem Dodgson-Winner, and the algorithm Greedy-Score intends to compute the Dodgson score of some given candidate. Both heuristics work well sufficiently often, provided that the number of voters greatly exceeds the number of candidates.

Here, we define only distributional search problems; the definition of distributional decision problems is analogous.

Definition 7.1 ([Lev86], see also [Gol97, Wan97]). *1. A* distribution function $\mu : \Sigma^* \to [0,1]$ *is a nondecreasing function from strings to the unit interval that converges to one (i.e.,* $\mu(0) \geq 0$*,* $\mu(x) \leq \mu(y)$ *for each* $x < y$*, and* $\lim_{x\to\infty} \mu(x) = 1$*). The* density function associated with μ *is defined by* $\mu'(0) = \mu(0)$ *and* $\mu'(x) = \mu(x) - \mu(x-1)$ *for each* $x > 0$*. That is, each string* x *gets weight* $\mu'(x)$ *with this distribution.*

2. *A* distributional (search) problem *is a pair* (f,μ)*, where* $f : \Sigma^* \to \Sigma^*$ *is a function and* $\mu : \Sigma^* \to [0,1]$ *is a distribution function.*

3. *A function* $t : \Sigma^* \to \mathbb{N}$ *is* polynomial on the average with respect to some distribution μ *if there exists a constant* $\varepsilon > 0$ *such that*

$$\sum_{x\in\Sigma^*} \mu'(x) \cdot \frac{t(x)^{\varepsilon}}{|x|} < \infty.$$

4. *The class* AvgP *consists of all distributional problems* (f,μ) *for which there exists an algorithm* $\mathscr{A}$ *computing* f *such that the running time of* $\mathscr{A}$ *is polynomial on the average with respect to the distribution* μ*.*

In Section 7.4, we will focus on the standard uniform distribution $\hat{\mu}$ on Σ^*, which is defined by

$$\hat{\mu}'(x) = \frac{1}{|x|(|x|+1)2^{|x|}}.$$

That is, we first choose an input size n at random with probability $1/(n(n+1))$, and then we choose an input string of that size n uniformly at random.

In Section 7.4, we we will make use of polynomial-time benign algorithm schemes. This notion was introduced by Impagliazzo [Imp95] to provide an alternative view on the definition of Levin's class AvgP (average polynomial time, see [Lev86]). Impagliazzo defines AvgP to be the class of distributional problems (f, μ_n) such that there is an algorithm computing f in polynomial time on the average with respect to the input ensemble μ_n.

We in this chapter use the following notation. For any distribution μ and for each $n \in \mathbb{N}$ let μ_n be the restriction of μ to strings of length exactly n, and let $\mu_{\leq n}$ be the restriction of μ to strings of length at most n. (When discussing benign algorithms, the length-0 string ε is routinely completely excluded from the probability distribution—it is by convention given weight zero—and so for such cases, e.g., Definition 7.2, the "for each $n \in \mathbb{N}$" should be viewed as changed to saying "for each $n \in \mathbb{N}^+$.")

Definition 7.2 ([Imp95]). 1. *An algorithm computes a function f with* benign faults *if it either outputs an element of the image of f or "?," and if it outputs anything other than ?, it is correct.*

2. *Let μ be a distribution on Σ^*. A* polynomial-time benign algorithm scheme *for a function f on μ is an algorithm $\mathscr{A}(x, \delta)$ such that:*

   (a) $\mathscr{A}$ runs in time polynomial in $|x|$ and $1/\delta$.

   (b) $\mathscr{A}$ computes f with benign faults.

   (c) For each δ, $0 < \delta < 1$, and for each $n \in \mathbb{N}^+$,

$$\mathrm{Prob}_{\mu_{\leq n}}[\mathscr{A}(x, \delta) = \ ?] \leq \delta.$$

The following theorem gives Impagliazzo's characterization of AvgP.

Theorem 7.1 ([Imp95]). *A problem f on μ is in AvgP if and only if it has a polynomial-time benign algorithm scheme on μ.*

7.3 Frequently Self-Knowingly Correct Algorithms

Homan and Hemaspaandra [HHa] proposed the following definition of a new type of algorithm to capture the notion of "guaranteed high success frequency" formally.

Definition 7.3 ([HHa]). 1. *Let $f : S \to T$ be a function, where S and T are sets. We say an algorithm $\mathscr{A} : S \to T \times \{$"definitely", "maybe"$\}$ is* self-knowingly correct for *f if, for each $s \in S$ and $t \in T$, whenever $\mathscr{A}$ on input s outputs $(t,$ "definitely"$)$ then $f(s) = t$.*

2. *An algorithm $\mathscr{A}$ that is self-knowingly correct for $g : \Sigma^* \to T$ is said to be* frequently self-knowingly correct for *g if*

$$\lim_{n \to \infty} \frac{\|\{x \in \Sigma^n \mid A(x) \in T \times \{\text{“maybe”}\}\}\|}{\|\Sigma^n\|} = 0.$$

In their paper [HHa], Homan and Hemaspaandra present two frequently self-knowingly correct polynomial-time algorithms, which they call Greedy-Score and Greedy-Winner. Since Greedy-Winner can easily be reduced to Greedy-Score, we focus on Greedy-Score only and briefly describe the intuition behind this algorithm; for full detail, we refer to [HHa]. (But both heuristics work well tremendously often—in a formal sense of the notion—provided that the number of voters greatly exceeds the number of candidates.)

Given a Dodgson triple (C,V,c), Greedy-Score determines the Dodgson score of c with respect to the given election (C,V). We will see that there are Dodgson triples (C,V,c) for which this problem is particularly easy to solve.

For any $d \in C - \{c\}$, let $\mathrm{Deficit}[d]$ be the number of votes c needs to gain in order to have more votes than d in a pairwise contest between c and d.

Definition 7.4. *Any Dodgson triple (C,V,c) is said to be* nice *if and only if for each candidate $d \in C - \{c\}$, there are at least* $\mathrm{Deficit}[d]$ *votes for which candidate c is exactly one position below candidate d.*

Given a Dodgson triple (C,V,c), the algorithm Greedy-Score works as follows:

1. For each candidate $d \in C - \{c\}$, determine $\mathrm{Deficit}[d]$.

2. If (C,V,c) is not nice then output (“anything”,“maybe”); otherwise, if (C,V,c) is nice then output
$$(\textstyle\sum_{d\in C-\{c\}}\mathrm{Deficit}[d], \text{“definitely”}).$$

Note that, for nice Dodgson triples, we have

$$\mathrm{DodgsonScore}(C,V,c) = \sum_{d\in C-\{c\}} \mathrm{Deficit}[d],$$

It is easy to see that Greedy-Score is a self-knowingly correct polynomial-time bounded algorithm. To show that it is even *frequently* self-knowingly correct, Homan and Hemaspaandra prove the following lemma. Their proof uses a variant of Chernoff bounds which we introduce next.

Lemma 7.1 (Lemma 11.9 of [Pap95])**.** *Let $x_1, x_2, \ldots, x_n$ be independent random variables with $x_i \in \{0,1\}$ for all i, $1 \leq i \leq n$, and let $X = \sum_{i=1}^{n} x_i$. Each variable with value* 1 *or* 0 *has probability p or $1-p$, respectively. Then for all* $0 \leq \theta \leq 1$,

$$\mathrm{Prob}[X \geq (1+\theta)pn] \leq e^{-\frac{\theta^2}{3}pn}.$$

Lemma 7.2 (Thm. 4.1.3 of [HHa])**.** *Let (C,V,c) be a given Dodgson triple, where $\|V\| = n$ and $\|C\| = m$, chosen uniformly at random among all such Dodgson elections. The probability that (C,V,c) is not nice is at most $2(m-1)e^{-\frac{n}{8m^2}}$.*

Proof. Suppose that (C,V,c) is not nice. Then, for some $d \in V - \{c\}$,

$$\|\{i \in \{1,\dots n\} \mid c <_{v_i} d\}\| - \frac{n}{2} \geq \|\{i \in \{1,\dots n\} \mid c \prec_{v_i} d\}\|.$$

This is true only if, for some $d \in V - \{c\}$, at least one of (7.3.1) and (7.3.3) is true, where

$$||\{i \in \{1,\dots n\} \mid c <_{v_i} d\}|| - \frac{n}{2} \tag{7.3.1}$$

$$\geq \quad \frac{1}{4} \cdot \frac{n}{m}, \tag{7.3.2}$$

$$||\{i \in \{1,\dots n\} \mid c \prec_{v_i} d\}|| \tag{7.3.3}$$

$$\leq \quad \frac{3}{4} \cdot \frac{n}{m}. \tag{7.3.4}$$

Using Chernoff bounds (see Lemma 7.1), it is easy to prove that the probability for both the events (7.3.1) and (7.3.3) are exponentially small in $\frac{n}{24m^2}$. Here, the fact is used that for arbitrary fixed candidates a and b, for some random vote v_i, $\mathrm{Prob}[a <_{v_i} b] = 1/2$ and $\mathrm{Prob}[a \prec_{v_i} b] = 1/m$.

Summing over all candidates $d \in C - \{c\}$, we obtain that the probability that (C,V,c) is not nice, is $\leq 2(m-1)e^{-\frac{n}{24m^2}}$. ❑

Homan and Hemaspaandra [HHa] show that the heuristic Greedy-Winner, which is based on Greedy-Score and which solves the winner problem for Dodgson elections, also is a frequently self-knowingly correct polynomial-time algorithm. This result is stated formally below.

Theorem 7.2 (Thm. 4.4.2 of [HHa])**.** *For all $m,n \in \mathbb{N}^+$, the probability that a Dodgson election (C,V) selected uniformly at random from all Dodgson elections having m candidates and n votes (i.e., all $(m!)^n$ Dodgson elections having m candidates and n votes have the same likelihood of being selected) has the property that there exists at least one candidate c such that Greedy-Winner on input (C,V,c) outputs "maybe" as its second output component is less than $2(m^2-m)e^{-\frac{n}{8m^2}}$.*

7.4 AvgP vs. Frequently Self-Knowingly Correct Algorithms

Our main result in this section relates polynomial-time benign algorithm schemes to frequently self-knowingly correct algorithms. We show that every distributional problem that has a polynomial-time benign algorithm scheme with respect to the uniform distribution must also have a frequently self-knowingly correct polynomial-time algorithm. It follows that all uniformly distributed AvgP problems have a frequently self-knowingly correct polynomial-time algorithm.

Theorem 7.3. *Suppose that $\mathscr{A}(x,\delta)$ is a polynomial-time benign algorithm scheme for a distributional problem f on $\hat{\mu}$ (the standard uniform distribution, see Section 7.2). Then there is a frequently self-knowingly correct polynomial-time algorithm $\mathscr{A}'$ for f.*

Proof. For each $n \in \mathbb{N}$, let $\delta(n) = 1/(n+1)^3$. Define algorithm $\mathscr{A}'$ as follows:

1. On input $x \in \Sigma^*$, simulate $\mathscr{A}(x, \delta(|x|))$.
2. If $\mathscr{A}(x, \delta(|x|))$ outputs ?, then output $($*anything*, “*maybe*”$)$.
3. If $\mathscr{A}(x, \delta(|x|))$ outputs $y \in T$, where $y \neq ?$, then output $(y$, “*definitely*”$)$.

By Definition 7.2, of “polynomial-time benign algorithm scheme,” algorithm $\mathscr{A}'$ runs in polynomial time. It remains to show that $\mathscr{A}'$ is frequently self-knowingly correct.

Fix an arbitrary $n \in \mathbb{N}^+$. Now, we must be careful regarding the fact that Impagliazzo’s definition of benign algorithm schemes and its “δ” guarantees are all with regard to drawing not over inputs of a given length (which is what we wish to consider) but rather regarding drawing from inputs *up to and including* a given length. Thus, there is some danger that even if a benign algorithm performs well when its length parameter is n (meaning related to strings of length up to and including n), that such a “good” error frequency might be due not to goodness at length n but rather to goodness at lengths $n-1$, $n-2$, and so on. However, if one looks carefully at the relative weights of the different lengths this is at most a quadratically weighted effect (that is, the distribution’s probability weight at length n is just quadratically less than the weight summed over all lengths less than n), and so our choice of $\delta(n) = 1/(n+1)^3$ is enough to overcome this.

Let us now handle that rigorously. Recall that n is fixed and arbitrary. Let us set the constant (for fixed n) δ' to be $1/(n+1)^3$. So, clearly

$$\mathrm{Prob}_{\hat{\mu}_{\leq n}}[\mathscr{A}(x,\delta') = ?] = \frac{\sum_{i=1}^{n-1} 1/(i(i+1))}{\sum_{i=1}^{n} 1/(i(i+1))}\mathrm{Prob}_{\hat{\mu}_{\leq n-1}}[\mathscr{A}(x,\delta') = ?] + \frac{1/(n(n+1))}{\sum_{i=1}^{n} 1/(i(i+1))}\mathrm{Prob}_{\hat{\mu}_{n}}[\mathscr{A}(x,\delta') = ?].$$

Since $\mathscr{A}$ is a benign algorithm scheme, $\mathrm{Prob}_{\hat{\mu}_{\leq n}}[\mathscr{A}(x,\delta') = ?] \leq \delta'$. So, combining this and the above equality, and solving for $\mathrm{Prob}_{\hat{\mu}_{n}}[\mathscr{A}(x,\delta') = ?]$, we have

$$\mathrm{Prob}_{\hat{\mu}_{n}}[\mathscr{A}(x,\delta') = ?] \leq \frac{\sum_{i=1}^{n} 1/(i(i+1))}{1/(n(n+1))}\left(\delta' - \frac{\sum_{i=1}^{n-1} 1/(i(i+1))}{\sum_{i=1}^{n} 1/(i(i+1))}\mathrm{Prob}_{\hat{\mu}_{\leq n-1}}[\mathscr{A}(x,\delta') = ?]\right).$$

And so, clearly, $\mathrm{Prob}_{\hat{\mu}_n}[\mathscr{A}(x,\delta') = ?] \leq n(n+1)\delta' = n(n+1)/(n+1)^3$. So

$$\lim_{n\to\infty} \frac{\|\{x \in \Sigma^n \mid \mathscr{A}'(x) \in T \times \{\text{“maybe”}\}\}\|}{\|\Sigma^n\|} = 0,$$

which completes the proof. ❑

Corollary 7.1. *Every distributional problem that under the standard uniform distribution is in* AvgP *has a frequently self-knowingly correct polynomial-time algorithm.*

Proof. Impagliazzo proved that any distributional problem on input ensemble μ_n is in AvgP if and only if it has a polynomial-time benign algorithm scheme; see Proposition 2 in [Imp95]. The claim now follows from Theorem 7.3. ❑

It is easy to see that the converse implication of that in Corollary 7.1 is not true.

Proposition 7.1. *There exist (distributional) problems with a frequently self-knowingly correct polynomial-time algorithm that are not in* AvgP *under the standard uniform distribution.*

Proof. For instance, one can define a problem that consists only of strings in $\{0\}^*$ encoding the halting problem. This problem is clearly not in AvgP, yet it is frequently self-knowingly correct. ❑

7.5 A Basic Junta Distribution for SAT

Procaccia and Rosenschein [PR07] introduced “junta distributions” in their study of NP-hard manipulation problems for elections. The goal of a junta is to be such a hard distribution (that is, to focus so much weight on hard instances) that if a problem is easy relative to a junta then it will be easy relative to any reasonable distribution (such as the uniform distribution).

Regarding Procaccia and Rosenschein’s notion of juntas, they state three “basic” conditions for a junta, and then give two additional ones that are tailored specifically to the needs of NP-hard voting manipulation problems. They state their hope that their scheme will extend more generally, using the three basic conditions and potentially additional conditions, to other mechanism problems. We will show that the three basic conditions for a junta are sufficiently weak that one can construct a junta relative to which the standard NP-complete problem SAT—and a similar attack can be carried out on a wide range of natural NP-complete problems—has a deterministic heuristic polynomial-time algorithm. This section's contribution is to give a construction indicating that the core three junta conditions, standing on their own, seem too weak.

Since we will use the Procaccia–Rosenschein junta notion in a more general setting than merely manipulation problems, we to avoid any chance of confusion will use the term "basic junta" to denote that we have removed the word "manipulation" and that we are using their three "basic" properties, and not the two additional properties that are specific to voting manipulation. Our definition of "deterministic heuristic polynomial-time algorithm" is the same as theirs, and our definition of "basic deterministic heuristic polynomial-time algorithm" is the same as their notion of "susceptible" (we avoid the word "susceptible" as that term already has term-of-art meanings in the study of the complexity of elections, e.g., in [BTT92] and the line of work it started, see also Chapter 3) except we have replaced the word "junta" with "basic junta"—and so again we are allowing their notion to be extended beyond just manipulation and mechanism problems.

Definition 7.5. 1. *(see [PR07]) Let $\mu = \{\mu_n\}_{n\in\mathbb{N}}$ be a distribution over the possible instances of an* NP*-hard problem L. (In this model, each μ_n sums to 1 over all length n instances.) We say μ is a* basic junta distribution *if and only if μ has the following properties:*

   (a) **Hardness:** *The* restriction of L to μ *is the problem whose possible instances are only $\bigcup_{n\in\mathbb{N}}\{x \mid |x| = n \text{ and } \mu_n(x) > 0\}$. Deciding this restricted problem is still* NP*-hard.*

   (b) **Balance:** *There exist constants $c > 1$ and $N \in \mathbb{N}$ such that for all $n \geq N$ and for all instances x, $|x| = n$, we have $1/c \leq \mathrm{Prob}_{\mu_n}[x \in L] \leq 1 - 1/c$.*

   (c) **Dichotomy:** *There exists some polynomial p such that for all n and for all instances x, $|x| = n$, either $\mu_n(x) \geq 2^{-p(n)}$ or $\mu_n(x) = 0$.*

2. *(see [PR07]) Let (L,μ) be a distributional decision problem (see Definition 7.1 in Appendix 7.2). An algorithm $\mathscr{A}$ is said to be a* deterministic heuristic polynomial-time algorithm for (L,μ) *if $\mathscr{A}$ is a deterministic polynomial-time algorithm and there exist a polynomial q and $N \in \mathbb{N}$ such that for each $n \geq N$,*

$$\mathrm{Prob}_{\mu_n}[x \notin L \Longleftrightarrow \mathscr{A} \text{ accepts } x] < \frac{1}{q(n)}.$$

   When such a μ and $\mathscr{A}$ exist, we'll say that L is in deterministic heuristic polynomial time (with respect to μ).

3. *(see [PR07]) Let (L,μ) be a distributional decision problem (see Definition 7.1 in Appendix 7.2). An algorithm $\mathscr{A}$ is said to be a* basic deterministic heuristic

polynomial-time algorithm for (L, μ) *if μ is a basic junta distribution (for L), $\mathscr{A}$ is a deterministic polynomial-time algorithm, and there exist a polynomial q and $N \in \mathbb{N}$ such that for each $n \geq N$,*

$$\mathrm{Prob}_{\mu_n}[x \notin L \iff \mathscr{A} \textit{ accepts } x] < \frac{1}{q(n)}.$$

When such a μ and $\mathscr{A}$ exist, we'll say that L is in basic deterministic heuristic polynomial time (with respect to μ).

We now explore their notion of deterministic heuristic polynomial time and their notion of junta, both however viewed for general NP problems and using the "basic" three conditions. We will note that the notion in such a setting is in some senses not restrictive enough and in other senses is too restrictive. Let us start with the former. We need a definition.

Definition 7.6. *We will say that a set L is* well-pierced *(respectively,* uniquely well-pierced*) if there exist sets Pos $\in$* P *and Neg $\in$* P *such that Pos $\subseteq L$, Neg $\subseteq \overline{L}$, and there is some $N \in \mathbb{N}$ such that at each length $n \geq N$, each of Pos and Neg has at least one string at length n (respectively, each of Pos and Neg has exactly one string at length n).*

Each uniquely well-pierced set is well-pierced. Note that, under quite natural encodings, such NP-complete sets as, for example, SAT certainly are well-pierced and uniquely well-pierced. (All this says is that, except for a finite number of exceptional lengths, there is one special string at each length that can easily, uniformly be recognized as in the set and one that can easily, uniformly be recognized as not in the set.) Indeed, under quite natural encodings, undecidable problems such as the halting problem are uniquely well-pierced.

Recall that juntas are defined in relation to an infinite list of distributions, one per length (so $\mu = \{\mu_n\}_{n \in \mathbb{N}}$). The Procaccia and Rosenschein definition of junta does not explicitly put computability or uniformity requirements on such distributions in the definition of junta, but it is useful to be able to make claims about that. So let us say that such a distribution is *uniformly computable in polynomial time* (respectively, is *uniformly computable in exponential time*) if there is a polynomial-time function (respectively, an exponential-time function) f such that for each i and each x, $f(i,x)$ outputs the value of $\mu_i(x)$ (say, as a rational number—if a distribution takes on other values, it simply will not be able to satisfy our notion of good uniform time).

Theorem 7.4. *Let A be any* NP*-hard set that is well-pierced. Then there exists a basic junta distribution relative to which A has a basic deterministic heuristic polynomial-time algorithm (indeed, it even has a basic deterministic heuristic polynomial-time algorithm* *whose error weight is bounded not merely by* $1/poly$ *as the definition requires, but is even bounded by* $1/2^{n^2-n}$*). Furthermore, the junta is uniformly computable in exponential time, and if we in addition assume that A is uniquely well-pierced, the junta is uniformly computable in polynomial time.*

It follows that, under quite natural encodings, almost any natural set is in basic deterministic heuristic polynomial time. For example, SAT is and the halting problem is, both under natural encodings. All it takes is for the given set to have at all but a finite number of lengths at least one element each that are uniformly easily recognizable as being in and out of the set.

Proof. Let A be well-pierced. So there exists an N, and sets *Pos* and *Neg*, that satisfy the definition of well-pierced. For each $n \geq N$, let $Pos(n)$ denote the lexicographically smallest length n string in *Pos* and let $Neg(n)$ denote the lexicographically smallest length n string in *Neg*.

Define the distribution $\nu = \{\nu_n\}_{n\in\mathbb{N}}$ as follows:

1. For each length $n \geq N$, put weight $1/2^{n^2}$ on all length n strings other than $Pos(n)$ and $Neg(n)$, and put weight $\frac{1}{2}\left(1 - \frac{2^n-2}{2^{n^2}}\right)$ on each of $Pos(n)$ and $Neg(n)$.

2. For each length $n < N$, let ν_n be the uniform distribution over that length, i.e., each length n string has weight $1/2^n$.

We now show that ν is a basic junta distribution.

1. Hardness: Since $\bigcup_{n\in\mathbb{N}}\{x \mid |x| = n \text{ and } \nu_n(x) > 0\}$ equals Σ^*, the restriction of A to ν equals A, and so is still NP-hard.

2. Balance: Since for each length $n \geq N$ both $Pos(n) \in A$ and $Neg(n) \notin A$ have almost half of the probability weight of all length n strings (namely, each has $\frac{1}{2}\left(1 - \frac{2^n-2}{2^{n^2}}\right)$), ν is balanced.

3. Dichotomy: Since for all $n \geq N$ and for all x, $|x| = n$, we have $\nu_n(x) \geq 2^{-n^2}$, and for all $n < N$ and for all x, $|x| = n$, we have $\nu_n(x) \geq 2^{-n}$, dichotomy is satisfied.

Note that the junta is uniformly computable in exponential time, and if A is uniquely well-pierced then the junta is uniformly computable in polynomial time.

Our basic deterministic heuristic polynomial-time algorithm for (A,ν) works as follows: On inputs that are a $Pos(n)$, it accepts; on inputs that are a $Neg(n)$, it rejects; and on every other input, it (for specificity, though it does not matter) accepts.

For each $n \geq N$, the error probability of this algorithm on inputs of length n is at most $(2^n-2)/2^{n^2} \leq 1/2^{n^2-n}$. ❑

In the proof we achieve the error bound $1/2^{n^2-n}$ stated in Theorem 7.4. However, this bound can easily be strengthened to $1/2^{n^k-n}$, for each fixed constant k, by altering the proof. Note that the altered algorithm will depend on k.

Loosely put, the above result says that the basic junta conditions are in some ways overinclusive. We also note that the definition of junta, and the issue of when we will have a basic deterministic heuristic polynomial-time algorithm, are exceedingly sensitive to details of encoding.[1] We mention quickly two such effects, one that indirectly suggests overinclusiveness and one that suggests underinclusiveness.

As to the former, note that *every* NP-hard set is $\leq_m^p$-reducible to a set that is in basic deterministic heuristic polynomial time. This applies even to undecidable NP-hard sets, such as $\mathrm{SAT} \oplus \mathrm{HP} =_{def} \{0x \mid x \in \mathrm{SAT}\} \cup \{1y \mid y \in \mathrm{HP}\}$, where HP denotes the halting problem. The proof is nearly immediate. Given an NP-hard set A (over some alphabet Σ that has cardinality at least two, and w.l.o.g. we assume that 0 and 1 are letters of Σ), note that $A \leq_m^p$-reduces to the set $A' = \{00x \mid x \in \Sigma^*\} \cup \{1x1^{|x|^2+2} \mid x \in A\}$, and that A' is easily seen to be in basic deterministic heuristic polynomial time (indeed, with error bound not just $1/$*poly* but even $1/$*exponential*), in particular via the basic junta (relative to A') that is the uniform distribution.

Regarding underinclusiveness, note that under the definition of basic junta, no set that at an infinite number of lengths either has all strings or has no strings can have a basic deterministic heuristic polynomial-time algorithm, since for such sets the balance condition of the notion of a basic junta can never be satisfied. It follows easily that the notion of having a basic deterministic heuristic polynomial-time algorithm is not even closed under polynomial-time isomorphisms.[2]

[1] In contrast, the "ε" exponent and $|x|$ denominator (see Definition 7.1 in Section 7.2) in Levin's [Lev86] theory of AvgP, average-case polynomial-time, were precisely designed, in that different setting, to avoid such problems—problems that one gets by following the type of asymptotic focus on one length at a time that the Procaccia and Rosenschein model adopts. On the other hand, even Levin's theory has many subtleties and downsides, and to this day has not found anything resembling the type of widespread applicability of NP-completeness theory; see any of the many surveys on that topic.

[2] To be extremely concrete, the NP-complete set $B = \{00x \mid x \in \Sigma^*\} \cup \{1x1^{|x|^2+2} \mid x \in \mathrm{SAT}\}$ is (as per the above) easily seen to be in basic deterministic heuristic polynomial time, but the NP-complete set $B' = \{xx \mid x \in \mathrm{SAT}\}$, though it is by standard techniques polynomial-time isomorphic to B (see [BH77]), is not in basic deterministic heuristic polynomial time. If the reader wonders why we did not simply use two P sets, the reason is, under the Procaccia–Rosenschein definition, one needs NP-hardness to have a junta, and one needs a junta to put something in deterministic heuristic polynomial time.

Bibliography

[Arr63] K. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 1951 (revised edition 1963).

[BEH$^+$] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In J. Laslier and R. Sanver, editors, *Handbook of Approval Voting*. Springer. To Appear.

[BF78] S. Brams and P. Fishburn. Approval voting. *American Political Science Review*, 72(3):831–847, 1978.

[BF83] S. Brams and P. Fishburn. *Approval Voting*. Birkhäuser, Boston, 1983.

[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.

[BH77] L. Berman and J. Hartmanis. On isomorphisms and density of NP and other complete sets. *SIAM Journal on Computing*, 6(2):305–322, 1977.

[BK98] S. Brams and D. Kilgour. Fallback bargaining. Technical report, 1998.

[BL05] S. Bouveret and J. Lang. Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. In *Proceedings of the 19th International Joint Conference on Artificial Intelligence*, pages 935–940. Professional Book Center, July/August 2005.

[Bla58] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[BS] 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*. Springer. 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.

[BTT89a] J. Bartholdi III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT89b] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 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.

[BU08] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. In *Proceedings of the 2nd Annual International Conference on Combinatorial Optimization and Applications*, pages 43–53. Springer-Verlag *Lecture Notes in Computer Science #5165*, August 2008.

[CCDF97] L. Cai, J. Chen, R. G. Downey, and M. R. Fellows. On the parameterized complexity of short computation and factorization. *Archive for Mathematical Logic*, 36(4-5):321–337, 1997.

[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, January 2007.

[Ces03] M. Cesati. The turing way to parameterized complexity. *Journal of Computer and System Sciences*, 67(4):654 – 685, 2003.

[CFRS06] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. In U. Endriss and J. Lang, editors, *1st International Workshop on Computational Social Choice (COMSOC 2006)*, pages 87–96 (workshop notes). Universiteit van Amsterdam, December 2006.

[CLRS01] T. Cormen, C. Leiserson, R. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press and McGraw-Hill, second edition, 2001.

[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.

[Coo71] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, pages 151–158. ACM Press, 1971.

[DF95] R. Downey and M. Fellows. Fixed-parameter tractability and completeness ii: On completeness for w[1]. *Theoretical Computer Science*, 141(1-2):109–131, April 1995.

[DF99] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, Berlin, Heidelberg, New York, 1999.

[Die05] R. Diestel. *Graph Theory*. Graduate Texts in Mathematics, Volume 173. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[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.

[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.

[DS00] J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[EHRS07a] G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. On approximating optimal weighted lobbying, and frequency of correctness versus average-case polynomial time. In *Proceedings of the 16th International Symposium on Fundamentals of Computation Theory*, pages 300–311. Springer-Verlag *Lecture Notes in Computer Science #4639*, August 2007.

[EHRS07b] G. Erdélyi, L. Hemaspaandra, J. Rothe, and H. Spakowski. On approximating optimal weighted lobbying, and frequency of correctness versus average-case polynomial time. Technical Report TR-914, Department of Computer Science, University of Rochester, Rochester, NY, March 2007.

[EKM07] P. Everaere, S. Konieczny, and P. Marquis. The strategy-proofness landscape of merging. *Journal of Artificial Intelligence Research*, 28:49–105, 2007.

[ENR08a] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting broadly resists control. In *Proceedings of the 33rd International Symposium on Mathematical Foundations of Computer Science*, pages 311–322. Springer-Verlag *Lecture Notes in Computer Science #5162*, August 2008.

[ENR08b] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. Technical Report arXiv:0806.0535v4 [cs.GT], ACM Computing Research Repository (CoRR), June 2008. Revised, September 2008.

[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, May 2008.

[FG06] J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, 2006.

[FHH06a] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proc. AAAI'06*, pages 641–646. AAAI Press, 2006.

[FHH06b] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 641–646. AAAI Press, July 2006.

[FHH06c] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? Technical Report TR-895, Department of Computer Science, University of Rochester, Rochester, NY, April 2006. Revised, September 2006.

[FHHR] 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*. Springer. To appear.

[FHHR07a] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 724–730. AAAI Press, July 2007.

[FHHR07b] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *Proc. AAAI'07*, pages 724–730. AAAI Press, 2007.

[FHHR08a] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proceedings of the 4th International Conference on Algorithmic Aspects in Information and Management*, pages 165–176. Springer-Verlag *Lecture Notes in Computer Science #5034*, June 2008.

[FHHR08b] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proc. AAIM'08*, pages 165–176. Springer-Verlag LNCS #5034, June 2008.

[FHHR08c] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and control. Technical Report arXiv:0809.4484v2 [cs.GT], ACM Computing Research Repository (CoRR), September 2008.

[FHHR09] 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*. Springer, 2009. To appear. Available as Technical Report arXiv:cs/0609112v1 [cs.GT], ACM Computing Research Repository (CoRR), September 2006.

[Fis77] P. Fishburn. Condorcet social choice functions. *SIAM Journal on Applied Mathematics*, 33(3):469–489, 1977.

[Fis78] P. Fishburn. Axioms for approval voting: Direct proof. *Journal of Economic Theory*, 19(1):180–185, October 1978.

[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.

[Gib73] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41(4):587–601, 1973.

[GJ76] M. Garey and D. Johnson. Approximation algorithms for combinatorial problems: An annotated bibliography. In J. Traub, editor, *Algorithms and Complexity*, pages 41–52. Academic Press, 1976.

[GJ79] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.

[GMHS99] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*, pages 434–435. ACM Press, 1999.

[Gol97] O. Goldreich. Notes on Levin's theory of average-case complexity. Technical Report TR97-058, Electronic Colloquium on Computational Complexity, November 1997.

[Har69] F. Harary. *Graph Theory*. Addison-Wesley, 1969.

[Hem87] L. Hemachandra. The strong exponential hierarchy collapses. In *Proceedings of the 19th ACM Symposium on Theory of Computing*, pages 110–122. ACM Press, May 1987.

[HHa] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. *Journal of Heuristics*. To appear. Full version available as [HH05].

[HHb] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. *Journal of Heuristics*. To appear. Full version available as University of Rochester Department of Computer Science Technical Report TR-881, September 2005, revised June 2007.

[HH05] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. Technical Report TR-881, Department of Computer Science, University of Rochester, Rochester, NY, September 2005. Revised, June 2007.

[HH06] C. Homan and L. Hemaspaandra. Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. In *Proceedings of the 31st International Symposium on Mathematical Foundations of Computer Science*, pages 528–539. Springer-Verlag *Lecture Notes in Computer Science #4162*, August/September 2006.

[HHR97] 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, November 1997.

[HHR05] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. In *Proceedings of the 20th National Conference on Artificial Intelligence*, pages 95–101. AAAI Press, 2005.

[HHR07a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[HHR07b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1308–1314. AAAI Press, January 2007.

[HK05] J. K. Hodge and R. E. Klima. *The Mathematics of Voting and Elections: A Dands-On Approach*. Mathematical World, Volume 22. American Mathematical Society, 2005.

[HO02] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2002.

[Imp95] R. Impagliazzo. A personal view of average-case complexity. In *Proceedings of the 10th Structure in Complexity Theory Conference*, pages 134–147. IEEE Computer Society Press, 1995.

[Lev73] L. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973. In Russian. English translation in *Problems of Information Transmission*, 9:265–266, 1973.

[Lev86] L. Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[LR08] C. Lindner and J. Rothe. Fixed-parameter tractability and parameterized complexity, applied to problems from computational social choice. In A. Holder, editor, *Mathematical Programming Glossary*. INFORMS Computing Society, October 2008.

[Mer88] S. Merrill. *Making Multicandidate Elections More Democratic*. Princeton University Press, 1988.

[MP79] A. Meyer and M. Paterson. With what frequency are apparently intractable problems difficult? Technical Report MIT/LCS/TM-126, MIT Laboratory for Computer Science, Cambridge, MA, 1979.

[MPS] J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson's rule. *Social Choice and Welfare*. To appear. Conference version available as [MPS06].

[MPS06] J. McCabe-Dansted, G. Pritchard, and A. Slinko. Approximability of Dodgson's rule. In U. Endriss and J. Lang, editors, *Proceedings of the 1st International Workshop on Computational Social Choice*, pages 331–344. Universiteit van Amsterdam, December 2006. Available online at `staff.science.uva.nl/~ulle/COMSOC-2006/proceedings.html`.

[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.

[Pap95] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 2nd edition, 1995. Reprinted with corrections.

[PR07] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, 2007.

[PRZ07] A. Procaccia, J. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control, and winner-determination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1476–1481. AAAI Press, January 2007.

[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.

[Rot05] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, Berlin, Heidelberg, New York, 2005.

[RSV02] J. Rothe, H. Spakowski, and J. Vogel. Exact complexity of Exact-Four-Colorability and of the winner problem for Young elections. In R. Baeza-Yates, U. Montanari, and N. Santoro, editors, *Foundations of Information Technology in the Era of Network and Mobile Computing*, pages 310–322.

Kluwer Academic Publishers, August 2002. Proceedings of the 17th IFIP World Computer Congress/2nd IFIP International Conference on Theoretical Computer Science.

[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, June 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.

[Sch86] U. Schöning. Complete sets and closeness to complexity classes. *Mathematical Systems Theory*, 19(1):29–42, 1986.

[Ser88] Murat R. Sertel. Characterizing approval voting. *Journal of Economic Theory*, 45(1):207–211, June 1988.

[SSGL02] T. Sandholm, S. Suri, A. Gilpin, and D. Levine. Winner determination in combinatorial auction generalizations. In *Proceedings of the 1st International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 69–76. ACM Press, July 2002.

[Vaz03] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, second edition, 2003.

[Wag87] K. Wagner. More complicated questions about maxima and minima, and some closures of NP. *Theoretical Computer Science*, 51:53–80, 1987.

[Wan97] J. Wang. Average-case computational complexity theory. In L. Hemaspaandra and A. Selman, editors, *Complexity Theory Retrospective II*, pages 295–328. Springer-Verlag, 1997.