Journal of Artificial Intelligence Research 40 (2011) 305-351 Submitted 07/10; published 01/11

Multimode Control Attacks on Elections

Piotr Faliszewski FALISZEW@AGH.EDU.PL
Department of Computer Science
AGH University of Science and Technology
Kraków, Poland

Edith Hemaspaandra EH@CS.RIT.EDU
Department of Computer Science
Rochester Institute of Technology
Rochester, NY 14623 USA

Lane A. Hemaspaandra LANE@CS.ROCHESTER.EDU
Department of Computer Science
University of Rochester
Rochester, NY 14627 USA

Abstract

In 1992, Bartholdi, Tovey, and Trick opened the study of control attacks on elections—attempts to improve the election outcome by such actions as adding/deleting candidates or voters. That work has led to many results on how algorithms can be used to find attacks on elections and how complexity-theoretic hardness results can be used as shields against attacks. However, all the work in this line has assumed that the attacker employs just a single type of attack. In this paper, we model and study the case in which the attacker launches a multipronged (i.e., multimode) attack. We do so to more realistically capture the richness of real-life settings. For example, an attacker might simultaneously try to suppress some voters, attract new voters into the election, and introduce a spoiler candidate. Our model provides a unified framework for such varied attacks. By constructing polynomial-time multiprong attack algorithms we prove that for various election systems even such concerted, flexible attacks can be perfectly planned in deterministic polynomial time.

1. Introduction

Elections are a central model for collective decision-making: Actors' (voters') preferences among alternatives (candidates) are input to the election rule and a winner (or winners in the case of ties) is declared by the rule. Bartholdi, Orlin, Tovey, and Trick initiated a line of research whose goal is to protect elections from various attacking actions intended to skew the election's results. Bartholdi, Orlin, Tovey, and Trick's strategy for achieving this goal was to show that for various election systems and attacking actions, even seeing whether for a given set of votes such an attack is possible is NP-complete. Their papers (Bartholdi, Tovey, & Trick, 1989a; Bartholdi & Orlin, 1991; Bartholdi, Tovey, & Trick, 1992) consider actions such as voter manipulation (i.e., situations where a voter misrepresents his or her vote to obtain some goal) and various types of election control (i.e., situations where the attacker is capable of modifying the structure of an election, e.g., by adding or deleting either voters or candidates). Since then, many researchers have extended Bartholdi, Orlin, Tovey, and Trick's work by providing new models, new results, and new perspectives. But to the best of our knowledge, until now no one has considered the situation in which an attacker combines multiple standard attack types into a single attack—let us call that a *multipronged* (or *multimode*) *attack*.

Studying multipronged control is a step in the direction of more realistically modeling real-life scenarios. Certainly, in real-life settings an attacker would not voluntarily limit himself or herself to a single type of attack but rather would use all available means of reaching his or her goal. For example, an attacker interested in some candidate p winning might, at the same time, intimidate p's most dangerous competitors so that they would withdraw from the election, and encourage voters who support p to show up to vote. In this paper we study the complexity of such multipronged control attacks.[1]

Given a type of multiprong control, we seek to analyze its complexity. In particular, we try to show either that one can compute in polynomial time an optimal attack of that control type, or that even recognizing the existence of an attack is NP-hard. It is particularly interesting to ask about the complexity of a multipronged attack whose components each have efficient algorithms. We are interested in whether such a combined attack (a) becomes computationally hard, or (b) still has a polynomial-time algorithm. Regarding the (a) case, we give an example of a natural election system that displays this behavior. Our paper's core work studies the (b) case and shows that even attacks having multiple prongs can in many cases be planned with perfect efficiency. Such results yield as immediate consequences all the individual efficient attack algorithms for each prong, and as such allow a more compact presentation of results and more compact proofs. But they go beyond that: They show that the interactions between the prongs can be managed without such cost as to move beyond polynomial time.

The paper's organization is as follows. In Section 2 we discuss the relevant literature. In Section 3 we present the standard model of elections and describe relevant voting systems. In Section 4 we introduce multiprong control, provide initial results, and show how existing immunity, vulnerability, and resistance results interact with this model. In Section 5 we provide a complexity analysis of candidate and voter control in maximin elections, showing how multiprong control is useful in doing so. In Section 6 we consider fixed-parameter complexity of multiprong control, using as our parameter the number of candidates. Section 7 provides conclusions and open problems. In the appendix, we show that maximin has an interesting relation to Dodgson elections: No candidate whose Dodgson score is more than m^2 times that of the Dodgson winner(s) can be a maximin winner.

2. Related Work

Since the seminal paper of Bartholdi et al. (1992), much research has been dedicated to studying the complexity of control in elections. Bartholdi et al. (1992) considered constructive control only, i.e., scenarios where the goal of the attacker is to ensure some candidate's victory. Hemaspaandra, Hemaspaandra, and Rothe (2007) extended their work to the destructive case, i.e., scenarios in which the goal is to prevent someone from winning.

A central but elusive goal of control research is finding a natural election system (with a polynomial-time winner algorithm) that is resistant to all the standard types of con-

1. In fact, our framework of multiprong control includes the unpriced bribery of Faliszewski, Hemaspaandra, and Hemaspaandra (2009a), and can be extended to include manipulation.

trol, i.e., for which all the types of control are NP-hard. Hemaspaandra, Hemaspaandra, and Rothe (2009) showed that there exist highly resistant artificial election systems. Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a) then showed that the natural system known as Copeland voting is not too far from the goal mentioned above. And Erdélyi, Nowak, and Rothe (2009) then showed a system with even more resistances than Copeland, but in a slightly nonstandard voter model (see Baumeister, Erdélyi, Hemaspaandra, Hemaspaandra, & Rothe, 2010, for discussion and Erdélyi, Piras, & Rothe, 2010b, 2010a; Menton, 2010, for some related follow-up work).

Recently, researchers also started focusing on the parameterized complexity of control in elections. Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a) provided several fixed-parameter tractability results. Betzler and Uhlmann (2009) and Liu, Feng, Zhu, and Luan (2009) showed so-called W[1]- and W[2]-hardness results for control under various voting rules. In response to the conference version (Faliszewski, Hemaspaandra, & Hemaspaandra, 2009b) of the present paper, Liu and Zhu (2010) conducted a parameterized-complexity study of control in maximin elections.

Going in a somewhat different direction, Meir, Procaccia, Rosenschein, and Zohar (2008) bridged the notions of constructive and destructive control by considering utility functions, and in this model obtained control results for multiwinner elections. In multiwinner elections the goal is to elect a whole group of people (consider, e.g., parliamentary elections) rather than just a single person. Elkind, Faliszewski, and Slinko (2010a) and Maudet, Lang, Chevaleyre, and Monnot (2010) considered two types of problems related to control by adding candidates for the case where it is not known how the voters would rank the added candidates.

Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2011) and Brandt, Brill, Hemaspaandra, and Hemaspaandra (2010) have studied control (and manipulation and bribery) in so-called single-peaked domains, a model of overall electorate behavior from political science.

There is a growing body of work on manipulation that regards frequency of (non)hardness of election problems (see, e.g., Conitzer & Sandholm, 2006; Friedgut, Kalai, & Nisan, 2008; Dobzinski & Procaccia, 2008; Xia & Conitzer, 2008b, 2008a; Walsh, 2009; Isaksson, Kindler, & Mossel, 2010). This work studies whether a given NP-hard election problem (to date only manipulation/winner problems have been studied, not control problems) can be often solved in practice (assuming some distribution of votes). (One however should keep in mind that if any polynomial-time algorithm solves any NP-hard problem extremely frequently—if it errs on only a sparse set in the formal complexity-theoretic sense of that term—then P = NP, Schöning, 1986.) Such frequency results are of course very relevant when one's goal is to protect elections from manipulative actions, and so although NP-hardness is a very important step, it is but a first step towards truly broad, satisfying security. However, in this paper we typically take the role of an attacker and design control algorithms that are fast on *all* instances.

Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009b) and Faliszewski, Hemaspaandra, and Hemaspaandra (2010b) provide an overview of some complexity-of-election issues.

3. Preliminaries

This section covers preliminaries about elections and computational complexity.

3.1 Elections

An election is a pair (C, V), where $C = \{c_1, \ldots, c_m\}$ is the set of candidates and $V = (v_1, \ldots, v_n)$ is a collection of voters. Each voter v_i is represented by his or her preference list.[2] For example, if we have three candidates, c_1, c_2, and c_3, a voter who likes c_1 most, then c_2, and then c_3 would have preference list $c_1 > c_2 > c_3$.[3] Given an election $E = (C, V)$, by $N_E(c_i, c_j)$, where $c_i, c_j \in C$ and $i \neq j$, we denote the number of voters in V who prefer c_i to c_j. We adopt the following convention for specifying preference lists.

Convention 3.1. *Listing some set D of candidates as an item in a preference list means listing all the members of this set in increasing lexicographic order (with respect to the candidates' names), and listing $\overleftarrow{D}$ means listing all the members of D but in decreasing lexicographic order (with respect to the candidates' names).*

Example 3.2. *Let us give a quick example of the convention. If $C = \{$Bob, Carol, Ted, Alice$\}$ and $D = \{$Alice, Ted, Bob$\}$, then Carol $> D$ is shorthand for Carol $>$ Alice $>$ Bob $>$ Ted, and Carol $> \overleftarrow{D}$ is shorthand for Carol $>$ Ted $>$ Bob $>$ Alice.*[4]

Note that in the model used in this paper, we assume that the person trying to do the attack on the election knows what the votes, V, are. This has been the standard model in computational studies of attacks on elections ever since the seminal work of Bartholdi et al. (1989a, 1992) and Bartholdi and Orlin (1991). However, it is worth noting that it is an abstract model that has strains in its connection to the real world. Regarding proving lower bounds, such as NP-hardness results, results in this model are actually stronger: One is showing that even given full access to the votes, V, the attacker *still* has an NP-hard task. On the other hand, when we build polynomial-time attack algorithms in this model, those algorithms are benefiting from the model letting them know what the votes are. How natural this model is will vary by the situation, and one should always keep in mind that this is indeed an abstract model, not the real world itself. However, in many settings, it is not unreasonable to assume that the attacker might have strong information about the votes. That information might come from polls, or it might come from door-to-door or telephone canvassing, or it might come from voter registration or contribution records, or

2. We also assume that each voter has a unique name. However, all the election systems we consider here—except for the election system of Theorem 4.12—are oblivious to the particular voter names and the order of the votes.
3. Preference lists are also called preference orders, and in this paper we will use these two terms interchangeably.
4. In the constructions where we use this convention, we are using variable names for objects—such as $\{b_1, \ldots, b_{3k}\}$ and p and so on—that are used as candidate sets in elections that are being output by the reduction. However, since an election has as part of its input the set of candidates (which are named), our actual reductions will be assigning name strings to each of these objects, and so the ordering we have discussed is well-defined and can be easily carried out by the reductions. And in fact, in the context of those reductions, actual (string) values of the b_i's are probably already part of the input to the reduction. (We have chosen lexicographic order simply because polynomial-time reductions can sort things into it, and its reverse, without any problem.)

it might come (in intimate, human elections, such as votes on whether one should have a course-based or an exam-based M.S. degree in one's department) from great familiarity of the attacker with the voters, or it might come from the attacker being the vote collector.

An election system is a mapping that given an election (C, V) outputs a set W, satisfying $W \subseteq C$, called the winners of the election.[5]

We focus on the following five voting systems: plurality, Copeland, maximin, approval, and Condorcet. (However, in Section 6 and the appendix we take a detour through some other systems.) Each of plurality, Copeland, maximin, and approval assigns points to candidates and elects those that receive the most points. Let $E = (C, V)$ be an election, where $C = \{c_1, \dots, c_m\}$ and $V = (v_1, \dots, v_n)$. In plurality, each candidate receives a single point for each voter who ranks him or her first. In maximin, the score of a candidate c_i in E is defined as $\min_{c_j \in C - \{c_i\}} N_E(c_i, c_j)$. For each rational α, $0 \leq \alpha \leq 1$, in Copeland$^\alpha$ candidate c_i receives 1 point for each candidate c_j, $j \neq i$, such that $N_E(c_i, c_j) > N_E(c_j, c_i)$ and α points for each candidate c_j, $j \neq i$, such that $N_E(c_i, c_j) = N_E(c_j, c_i)$. That is, the parameter α describes the value of ties in head-to-head majority contests. In approval, instead of preference lists each voter's ballot is a 0-1 vector, where each entry denotes whether the voter approves of the corresponding candidate (gives the corresponding candidate a point). For example, vector $(1, 0, 0, 1)$ means that the voter approves of the first and fourth candidates, but not the second and third. We use $\mathrm{score}_E(c_i)$ to denote the score of candidate c_i in election E (the particular election system used will always be clear from context).

A candidate c is a Condorcet winner of an election $E = (C, V)$ if for each other candidate $c' \in C$ it holds that $N_E(c, c') > N_E(c', c)$. Condorcet voting is the election system in which the winner set is, by definition, exactly the set of Condorcet winners. It follows from the definition that each election has at most one Condorcet winner. Not every election has a Condorcet winner. However, as our notion of an election allows outcomes in which no one wins, electing the Condorcet winner when there is one and otherwise having no winner is a legal election system.

5. Readers with a social choice background may wonder why we do not forbid the case $W = \emptyset$, as is typically done in social choice framings of elections. Briefly put, allowing the possibility $W = \emptyset$ has been the standard model in the computational studies of elections, starting with the seminal papers of Bartholdi, Orlin, Tovey, and Trick. By retaining this model, our results can better be compared with the existing computational results on attacks on elections. In fact, the recent (admittedly computationally oriented) textbook of Shoham and Leyton-Brown (2009, Def. 9.2.2) treats the definition of a social choice correspondence as allowing any subset of the candidates, including the empty set (in contrast, in social choice papers, the notion of social choice correspondence routinely in its definition excludes the possibility of having no winners). Although we follow the model of allowing the empty outcome as it is the standard computational model and allows comparison with existing results, we mention in passing that we find this model, on its merits, the more attractive one, although this is certainly a matter of taste, familiarity, and comfort. This model avoids building in a special-case exception in the definition and allows one to discuss zero-candidates elections if for some reason one wants to. But more importantly, many natural election systems might have no winners. Examples include threshold election systems, including majority-rule elections and the election systems often used to see whether anyone—by exceeding a certain percentage of approval votes from a group of expert sports writers, for instance—merits induction into a sports Hall of Fame that year. Condorcet voting (to be defined later), which the seminal control-of-elections paper of Bartholdi et al. (1992) treated as an election system, also can have an empty winner set.

3.2 Computational Complexity

We use standard notions of complexity theory, as presented, e.g., in the textbook of Papadimitriou (1994). We assume that the reader is familiar with the complexity classes P and NP, polynomial-time many-one reductions, and the notions of NP-hardness and NP-completeness. $\mathbb{N}$ will denote $\{0, 1, 2, \ldots\}$.

Most of the NP-hardness proofs in this paper follow by a reduction from the well-known NP-complete problem *exact cover by 3-sets*, known for short as X3C (see, e.g., Garey & Johnson, 1979). In X3C we are given a pair $(B, \mathcal{S})$, where $B = \{b_1, \ldots, b_{3k}\}$ is a set of $3k$ elements and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a set of 3-subsets of B, and we ask whether there is a subset S' of exactly k elements of $\mathcal{S}$ such that their union is exactly B. We call such a set S' an exact cover of B.

In Section 6, we consider the fixed-parameter complexity of multiprong control. The idea of fixed-parameter complexity is to measure the complexity of a given decision problem with respect to both the instance size (as in the standard complexity theory) and some parameter of the input (in our case, the number of candidates involved). For a problem to be said to be fixed-parameter tractable, i.e., to belong to the complexity class FPT, we as is standard require that the problem can be solved by an algorithm running in time $f(j)n^{O(1)}$, where n is the size of the encoding of the given instance, j is the value of the parameter for this instance, and f is some function. Note that f does not have to be polynomially bounded or even computable. However, in all FPT claims in this paper, f is a computable function. That is, our algorithms actually achieve so-called strongly uniform fixed-parameter tractability. We point readers interested in parameterized complexity to, for example, the recent book by Niedermeier (2006).

4. Control and Multiprong Control

In this section we introduce multiprong control, that is, control types that combine several standard types of control. We first provide the definition, then proceed to analyzing general properties of multiprong control, then consider multiprong control for election systems for which the complexity of single-prong control has already been established, and finally give an example of an election system for which multiprong control becomes harder than any of its constituent prongs (assuming P $\neq$ NP). We conclude the section with a summary of its main contributions.

4.1 The Definition

We consider combinations of control by adding/deleting candidates/voters[6] and by bribing voters. Traditionally, bribery has not been considered a type of control but it fits the model very naturally and strengthens our results.

In discussing control problems, we must be very clear about whether the goal of the attacker is to make his or her preferred candidate *the only winner*, or is to make his or her preferred candidate *a winner*. To be clear on this, we as is standard will use the term "unique-winner model" for the model in which the goal is to make one's preferred

6. Other control types, defined by Bartholdi et al. (1992) and refined by Hemaspaandra et al. (2007), regard various types of partitioning candidates and voters.

candidate the one and only winner, and we will use the term "nonunique-winner model" for the approach in which the goal is to make one's preferred candidate be a winner. (Note that if exactly one person wins, he or she most certainly is considered to have satisfied the control action in the nonunique-winner model. The "nonunique" in the model name merely means we are not *requiring* that winners be unique.)

The destructive cases of each of these are, in the nonunique-winner model, blocking one's despised candidate from being a unique winner,[7] and in the unique-winner model, blocking one's despised candidate from being a winner. We take the unique-winner model as the default in this paper, as is the most common model in studies of control.

Definition 4.1. *Let $\mathcal{E}$ be an election system. In the unique-winner,*[8] *constructive $\mathcal{E}$-AC+DC+AV+DV+BV control problem we are given:*

(a) two disjoint sets of candidates, C and A,

(b) two disjoint collections of voters, V and W, containing voters with preference lists over $C \cup A$,

(c) a preferred candidate $p \in C$, and

(d) five nonnegative integers, k_{AC}, k_{DC}, k_{AV}, k_{DV}, and k_{BV}.

We ask whether it is possible to find two sets, $A' \subseteq A$ and $C' \subset C$, and two subcollections of voters, $V' \subseteq V$ and $W' \subseteq W$, such that:

(e) it is possible to ensure that p is a unique winner of $\mathcal{E}$ election $((C - C') \cup A', (V - V') \cup W')$ via changing preference orders of (i.e., bribing) at most k_{BV} voters in $(V - V') \cup W'$,

(f) $p \notin C'$, and

(g) $\|A'\| \leq k_{\mathrm{AC}}$, $\|C'\| \leq k_{\mathrm{DC}}$, $\|W'\| \leq k_{\mathrm{AV}}$, and $\|V'\| \leq k_{\mathrm{DV}}$.

In the unique-winner, destructive variant of the problem, we replace item (e) above with: "it is possible to ensure that p is not *a unique winner of $\mathcal{E}$ election $((C-C') \cup A', (V-V') \cup W')$ via changing preference orders of at most k_{BV} voters in $(V-V') \cup W'$." (In addition, in the destructive variant we refer to p as "the despised candidate" rather than as "the preferred candidate," and we often denote him or her by d.)*

Table 1 summarizes in informal English the notation used in this definition, so that this information is easily available for the reader to refer back to.

The phrase AC+DC+AV+DV+BV in the problem name corresponds to four of the standard types of control: adding candidates (AC), deleting candidates (DC), adding voters (AV), deleting voters (DV), and to (unpriced) bribery (BV); we will refer to these five types of control as the *basic* types of control. We again remind the reader that traditionally

7. We will often use the phrase "a unique winner," as we just did. The reason we write "a unique winner" rather than "the unique winner" is to avoid the impression that the election necessarily has some (unique) winner.
8. One can straightforwardly adapt the definition to the nonunique-winner model.

Notation	Meaning
AC	Control by adding candidates.
DC	Control by deleting candidates.
AV	Control by adding voters.
DV	Control by deleting voters.
BV	Control by bribing voters.
C	The set of initial candidates in an election.
A	The set of additional candidates that the control agent may introduce.
V	The collection of initial voters in an election.
W	The collection of additional voters that the control agent may introduce.
k_{AC}	The bound on the number of candidates that can be added in AC control.
k_{DC}	The bound on the number of candidates that can be deleted in DC control.
k_{AV}	The bound on the number of voters that can be added in AV control.
k_{DV}	The bound on the number of voters that can be deleted in DV control.
k_{BV}	The bound on the number of voters that can be bribed in BV control.
p	The preferred candidate (the constructive control goal is to ensure that p is a unique winner).
d	The despised candidate (the destructive control goal is to ensure that d is not a unique winner).

Table 1: Notations from Definition 4.1 that are used frequently elsewhere.

bribery is not a type of control but we will call it a basic type of control for the sake of uniformity and throughout the rest of the paper we will consider it as such.

As to why we choose these as the “basic” types, it is essentially because these are the collection on which we focus, and so the term is a handy one to use to indicate them. But we have focused on these particular ones largely because we find them highly attractive. The various “partition” control types that appeared in the original paper on control, while quite interesting, have always seemed less natural to us than adding/deleting voters/candidates. Bribery to us is also quite compellingly natural. The attack known as manipulation is not included by us among the “basic” types, but without a doubt is a natural and important type of attack on elections, and in the conclusion we discuss it briefly, and commend to the reader the issue of studying manipulation as an additional prong.

Instead of considering all of AC, DC, AV, DV, and BV, we often are interested in some subset of them and so we consider special cases of the AC+DC+AV+DV+BV problem. For example, we write DC+AV to refer to a variant of the AC+DC+AV+DV+BV problem where only deleting candidates and adding voters is allowed. As part of our model we assume that in each such variant, only the parameters relevant to the prongs are part of the input. So, for example, DC+AV would have k_{DC}, k_{AV}, C, V, W, and p as the (only) parts of its input. And the “missing” parts (e.g., for DC+AV, the missing parts are A, k_{AC}, k_{DV}, and k_{BV}) are treated in the obvious way in evaluating the formulas in Definition 4.1, namely, missing sets are treated as $\emptyset$ and missing constants are treated as 0. If we name only a single type of control, we in effect degenerate to one of the standard control problems.

The reader may naturally wonder in what order the prongs of a multi-prong attack occur. Note that part (e) of Definition 4.1 is quietly setting the order. However, almost all the order interactions are uninteresting (unless the attacker is idiotic). For example, the definition does not allow one to bribe voters that one will be deleting, but doing so would be pointless anyway, so this is not an interesting restriction on the attacker. Similarly, the definition does not allow one to add a voter and then immediately delete it, but again, that does not take even one successful attack away from the attacker. Indeed, the only interesting order interaction to consider is whether one can bribe added voters, or whether one can only bribe voters who were originally in the election. One could argue this either way, and if one wanted to avoid focusing on one or the other, one could analyze everything both ways. However, our definition embraces the model in which even the added voters can be bribed. Note that this is the model more favorable to the attacker. One referee commented that it would be unreasonable to give attackers the flexibility of multiple attacks but deny them the freedom to control the order of the attacks. In keeping with the spirit of that comment, our definition resolves the order issue in the way that is most favorable to the attacker, and so in no way is biased against the attacker. However, for completeness we should mention that it is possible that some—perhaps highly artificial—systems might have different attack complexities in the model where added voters cannot be bribed as compared to the model in which added voters can be bribed.

There is at least one more way in which we could define multiprong control. The model in the above definition can be called the *separate-resource model*, as the extent to which we can use each basic type of control is bounded separately. In the *shared-resource model* one pool of action allowances must be allocated among the allowed control types (so in the definition above we would replace $k_{\mathrm{AC}}, k_{\mathrm{DC}}, k_{\mathrm{AV}}$, and k_{DV} with a single value, k, and require that $\|C'\| + \|D'\| + \|V'\| + \|W'\| + \textit{the-number-of-bribed-voters} \leq k$). Although one could make various arguments about which model is more appropriate, their computational complexity is related.

Theorem 4.2. *If there is a polynomial-time algorithm for a given variant of multiprong control in the separate-resource model then there is one for the shared-resource model as well.*

Proof. Let $\mathcal{E}$ be an election system. We will describe the idea of our proof on the example of the constructive $\mathcal{E}$-AC+AV problem. The idea straightforwardly generalizes to any other set of allowed control actions (complexity-theory savvy readers will quickly see that we, in essence, give a disjunctive truth-table reduction).

We are given an instance I of the constructive $\mathcal{E}$-AC+AV problem in the shared-resource model, where k is the limit on the sum of the number of candidates and voters that we may add. Given a polynomial-time algorithm for the separate-resource variant of the problem, we solve I using the following method. (If $k > \|A\| + \|W\|$ then set $k = \|A\| + \|W\|$.) We form a sequence $I_0, \ldots, I_k$ of instances of the separate-resource variant of the problem, where each I_ℓ, $0 \leq \ell \leq k$, is identical to I, except that we are allowed to add at most ℓ candidates and at most $k - \ell$ voters. We accept if at least one of I_ℓ is a "yes" instance of the separate-resource, constructive $\mathcal{E}$-AC+AV problem. It is straightforward to see that this algorithm is correct and runs in polynomial time. $\square$

It would be interesting to consider a variant of the shared-resource model where various actions come at different costs (e.g., adding some candidate c' might be much more expensive—or difficult—than adding some other candidate c''). This approach would be close in spirit to priced bribery of Faliszewski, Hemaspaandra, and Hemaspaandra (2009a). Analysis of such priced control is beyond the scope of the current paper.

4.2 Susceptibility, Immunity, Vulnerability, and Resistance

As is standard in the election-control (and election-bribery) literature, we consider vulnerability, immunity, susceptibility, and resistance to control. Let $\mathcal{E}$ be an election system and let $\mathcal{C}$ be a type of control.

We say that $\mathcal{E}$ is susceptible to constructive $\mathcal{C}$ control if there is a scenario in which effectuating $\mathcal{C}$ makes someone become a unique winner of some $\mathcal{E}$ election E. We say that $\mathcal{E}$ is susceptible to destructive $\mathcal{C}$ control if there is a scenario in which effectuating $\mathcal{C}$ makes someone stop being a unique winner of some $\mathcal{E}$ election E.

$\mathcal{E}$ is immune to constructive (respectively, destructive) $\mathcal{C}$ control if $\mathcal{E}$ is not susceptible to constructive (respectively, destructive) $\mathcal{C}$ control.

We say that $\mathcal{E}$ is vulnerable to constructive (respectively, destructive) $\mathcal{C}$ control if $\mathcal{E}$ is susceptible to constructive (respectively, destructive) $\mathcal{C}$ control and there is a polynomial-time algorithm that decides the constructive (respectively, destructive) $\mathcal{E}$-$\mathcal{C}$ problem. Actually, this paper's vulnerability algorithms/proofs will each go further and will in polynomial time produce, or will make implicitly clear how to produce, the successful control action. So we in each case are even achieving the so-called certifiable vulnerability of Hemaspaandra et al. (2007).

$\mathcal{E}$ is resistant to constructive (respectively, destructive) $\mathcal{C}$ control if $\mathcal{E}$ is susceptible to (respectively, destructive) $\mathcal{C}$ control and the constructive (respectively, destructive) $\mathcal{E}$-$\mathcal{C}$ problem is NP-hard.

Before we move on to our theorems, we mention two important points to put these notions into context. Vulnerability is about (within susceptible settings) polynomial-time recognition of those inputs on which there are successful attacks (and on those, as noted above, we will actually in this paper produce the attacks), not about the proportion of inputs on which attacks succeed (however, see our comments and references earlier in this paper about work studying frequency of hardness issues). Also, for a type of multiprong control that, for example, has one on-its-own immune prong and some on-their-own vulnerable prongs, we are about to prove that immunity will not hold. (In some cases vulnerability will hold, and in some cases resistance might hold.) However, for that portion of the input universe that allows changes on the immune-on-its-own prong but not on the other prongs, one can quickly (assuming the winner problem for the given election system itself is a polynomial-time problem) both recognize that one is on that subspace of inputs and determine whether one can achieve one's goal (since due to the immunity we are on a "dead" prong—that prong on its own cannot raise us from failure to success).

The next few theorems describe how multiprong control problems can inherit susceptibility, immunity, vulnerability, and resistance from the basic control types that they are built from. (We will often write "(destructive)" rather than "(respectively, destructive)," when the "respectively" is clear from context.)

Theorem 4.3. *Let $\mathcal{E}$ be an election system and let $C_1+\cdots+C_k$ be a variant of multiprong control (so $1 \leq k \leq 5$ and each C_i is a basic control type). $\mathcal{E}$ is susceptible to constructive (destructive) $C_1+\cdots+C_k$ control if and only if $\mathcal{E}$ is susceptible to at least one of constructive (destructive) $C_1, \ldots, C_k$ control.*

Proof. The "if" direction is trivial: The attacker can always choose to use only the type of control to which $\mathcal{E}$ is susceptible. As to the "only if" direction, it is not hard to see that if there is some input election for which by a $C_1 + \cdots + C_k$ action we can achieve our desired change (of creating or removing unique-winnerhood for p, depending on the case), then there is *some* election (not necessarily our input election) for which one of those actions alone achieves our desired change. In essence, we can view a control action A of type $C_1+\cdots+C_k$ as a sequence of operations, each operation being of one of the $C_1, \ldots, C_k$ types, that—when executed in order—transform our input election into an election where our goal is satisfied. Thus there is a single operation within A—and this operation is of one of the types $C_1, \ldots, C_k$—that transforms some election E' where our goal is not satisfied to some election E'' where the goal is satisfied. □

We immediately have the following corollary.

Corollary 4.4. *Let $\mathcal{E}$ be an election system and let $C_1+\cdots+C_k$ be a variant of multiprong control (so $1 \leq k \leq 5$ and each C_i is a basic control type). $\mathcal{E}$ is immune to constructive (destructive) $C_1 + \cdots + C_k$ control if and only if for each i, $1 \leq i \leq k$, $\mathcal{E}$ is immune to constructive (destructive) C_i control.*

In the next theorem we show that if a given election system is vulnerable to some basic type of control and is immune to another basic type of control, then it is vulnerable to these two types of control combined. The proof of this theorem is straightforward, but we need to be particularly careful as vulnerabilities and immunities can behave quite unexpectedly. For example, it might seem that we can assume that if an election system is vulnerable to AV and DV then it should also be vulnerable to BV, because bribing a particular voter can be viewed as first deleting this voter and then adding—in his or her place—a voter with the preference order as required by the briber. (This assumes we have such a voter among the voters we can add, but when arguing susceptibility/immunity we can make this assumption.) However, there is a simple election system that is vulnerable to both AV and DV control, but that is immune to BV control. This system just says that in an election $E = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$, the winner is the candidate c_i such that $n \equiv i - 1 \pmod{m}$.[9]

Theorem 4.5. *Let $\mathcal{E}$ be an election system and let $C_1+\cdots+C_k+D_1+\cdots+D_\ell$ be a variant of multiprong control (so $1 \leq k \leq 5$, $1 \leq \ell \leq 5$, and each C_i and each D_i is a basic control type) such that $\mathcal{E}$ is vulnerable to constructive (destructive) $C_1 + \cdots + C_k$ control and for each i, $1 \leq i \leq \ell$, $\mathcal{E}$ is immune to constructive (destructive) D_i control.[10] $\mathcal{E}$ is vulnerable to $C_1 + \cdots + C_k + D_1 + \cdots + D_\ell$ control.*

9. Of course, this election system is not neutral; permuting the names of the candidates, evaluating the election's winners, and then running the winners through the inverse of the permutation can change the outcome of an election.
10. So we certainly have $k + \ell \leq 5$, since immunity and vulnerability are mutually exclusive.

Proof. We will give a proof for the constructive case only. The proof for the destructive case is analogous. Let $\mathcal{E}$ be an election system as in the statement of the theorem and let I be an instance of constructive $\mathcal{E}$-$C_1 + \cdots + C_k + D_1 + \cdots + D_\ell$ control, which contains election $E = (C, V)$, information about the specifics of control actions we can implement, and where the goal is to ensure that candidate p is a unique winner. Let us first consider the case where BV is not among $C_1, \ldots, C_k, D_1, \ldots D_\ell$.

Let us assume that there is a collection A of control actions of types $C_1, \ldots C_k, D_1, \ldots, D_\ell$, such that applying the actions from A to E is legal within I and results in an election E_{C+D} where p is the unique winner. (We take A to be empty if p is a unique winner of E.) We split A into two parts, A_C and A_D, such that A_C contains exactly the actions of types $C_1, \ldots, C_k$, and a A_D contains exactly the actions of types $D_1, \ldots, D_\ell$. Since BV is not among our control actions, it is straightforward to see that it is possible to apply actions A_C to election E to obtain some election E_C. (To see why it is important that we do not consider BV, assume that BV is among control types $C_1, \ldots, C_k$ and AV is among control types $D_1, \ldots, D_\ell$. In this case, A_C might include an action that bribes a voter that is added by an action from A_D.)

We claim that p is the unique winner of E_C. For the sake of contradiction, let us assume that this is not the case (note that this implies that p is not a unique winner of E). If we apply control actions A_D to E_C, we reach exactly election E_{C+D}, where p is the unique winner. Yet, this is a contradiction, because by Corollary 4.4 we have that $\mathcal{E}$ is immune to $D_1 + \cdots + D_\ell$. That is, there is no scenario where control actions of type $D_1 + \cdots + D_\ell$ make some candidate a unique winner if he or she was not a unique winner before.

Thus it is possible to ensure that p is a unique winner by actions of type $C_1 + \cdots + C_k$ alone. We chose I arbitrarily, and thus any instance of $\mathcal{E}$-$C_1 + \cdots + C_k + D_1 + \cdots + D_\ell$ control can be solved by an algorithm that considers control actions of type $C_1 + \cdots + C_k$ only. This proves that $\mathcal{E}$ is vulnerable to $C_1 + \cdots + C_k + D_1 + \cdots + D_\ell$ control because, as we have assumed, it is vulnerable to $C_1 + \cdots + C_k$ control.

It remains to prove the theorem for the case where BV *is* among our control actions. In the case where BV is among the control actions but AV is not, or if AV and BV are in the same group of actions (i.e., either both are among the C_i's or both are among the D_i's), it is straightforward to see that the above proof still works. Similarly, if BV is among the D_i's and AV is among the C_i's, the above proof works as well. The only remaining case is if our allowed control types include both BV and AV, where BV is among the C_i's and AV is among the D_i's.

In this last case, the proof also follows the general structure of the previous construction, except that we have to take care of one issue: It is possible that A_C includes bribery of voters that are to be added by actions from A_D. (We use the same notation as in the main construction.) Let V_{BV} be the collection of voters that A_C requires to bribe, but that are added in A_D. We form a collection A'_C of control actions that is identical to A_C, except that it includes adding the voters from V_{BV}, and we let A'_D be identical to A_D, except that it no longer includes adding the voters from V_{BV}. Using A'_C and A'_D instead of A_C and A_D, it is straightforward to show the following: If it is possible to ensure that p is a unique winner in instance I by a legal action of type $C_1 + \cdots + C_k + D_1 + \cdots + D_\ell$, then it is also possible to do so by a legal action of type $C_1 + \cdots + C_k + \mathrm{AV}$, where each added voter is also bribed. Thus given an instance I of $\mathcal{E}$-$C_1 + \cdots + C_k + D_1 + \cdots + D_\ell$ we can solve it using the following algorithm. Let W be the collection of voters that can be added within I and let k_{AV} be the limit on the number of voters that we can add.

1. Let t be $\min(k_{\mathrm{AV}}, \|W\|)$.
2. For each i in $\{0, 1, \ldots, t\}$ execute the next two substeps.
   (a) Form instance I' that is identical to I, except i (arbitrarily chosen) voters from W are added to the election.
   (b) Run the $\mathcal{E}$-$C_1 + \cdots + C_k$ algorithm on instance I' and accept if it does.
3. If the algorithm has not accepted yet, reject.

It is straightforward to see that this algorithm is correct and, since $\mathcal{E}$ is vulnerable to $C_1 + \cdots + C_k$, works in polynomial time. This completes the proof of the theorem. $\square$

Theorem 4.6. *Let $\mathcal{E}$ be an election system and let $C_1 + \cdots + C_k$ be a variant of multiprong control (so $1 \leq k \leq 5$ and each C_i is a basic control type). If for some i, $1 \leq i \leq k$, $\mathcal{E}$ is resistant to constructive (destructive) C_i control, then $\mathcal{E}$ is resistant to constructive (destructive) $C_1 + \cdots + C_k$ control.*

Proof. Let C_i be the control type to which $\mathcal{E}$ is resistant. Since $\mathcal{E}$ is susceptible to constructive (destructive) C_i control, it follows by Theorem 4.3 that $\mathcal{E}$ is susceptible to constructive (destructive) $C_1 + \cdots + C_k$ control. And since the $\mathcal{E}$-C_i constructive (destructive) control problem is essentially (give or take syntax) an embedded subproblem of the $\mathcal{E}$-$C_1 + \cdots + C_k$ control problem, it follows that $\mathcal{E}$ is resistant to $C_1 + \cdots + C_k$ control. $\square$

By combining the results obtained so far in Section 4.2, we obtain a simple tool that allows us to classify a large number of multiprong control problems based on the properties of their prongs. This theorem—along with the forthcoming "Classification Rule A," which shows how to apply this result to well-behaved election systems—is the central result of this paper.

Theorem 4.7. *Let $\mathcal{E}$ be an election system and let $C_1 + \cdots + C_k$ be a variant of multiprong control (so $1 \leq k \leq 5$ and each C_i is a basic control type), such that for each C_i, $1 \leq i \leq k$, $\mathcal{E}$ is resistant, vulnerable, or immune to constructive (destructive) C_i control. If there is an i, $1 \leq i \leq k$, such that $\mathcal{E}$ is resistant to constructive (destructive) C_i control then $\mathcal{E}$ is resistant to constructive (destructive) $C_1 + \cdots + C_k$ control. Otherwise, if for each i, $1 \leq i \leq k$, $\mathcal{E}$ is immune to constructive (destructive) C_i control ($1 \leq i \leq k$), then it is immune to constructive (destructive) $C_1 + \cdots + C_k$ control. Otherwise, if $\mathcal{E}$ is vulnerable to the constructive (destructive) multiprong control consisting of all the individual prongs among the C_i for which $\mathcal{E}$ is vulnerable to constructive (destructive) control, then $\mathcal{E}$ is vulnerable to constructive (destructive) $C_1 + \cdots + C_k$ control.*

To avoid any confusion, we stress that throughout one's reading of the above corollary, one must either always use the "constructive" case or must always use the "destructive" case—one cannot mix and match. Also, our third "otherwise" really is an otherwise; its claim assumes that the first case (that there is at least one resistance) did not hold.

So does the vulnerability/resistance/immunity information on the five individual prongs completely determine which of vulnerability/resistance/immunity holds for each of the 2^5-1 multiprong settings? It would be very satisfying if this were so. However, later results in this paper show that this is not the case, since we will prove that two vulnerable prongs can combine to yield resistance (Theorem 4.12) but also can combine to yield vulnerability (e.g., Theorem 4.10). (It is even plausible that there may exist cases where two vulnerable prongs combine to yield a multiprong case that, while certainly susceptible due to Theorem 4.3 (i.e., immunity is impossible in this case), is neither in P nor NP-hard, i.e., is neither vulnerable nor resistant.)

However, for every voting system that has a certain common, nice property, we *can* from the 5 individual prongs' vulnerability/resistance/immunity status mechanically read off all $2^5 - 1$ multiprong results. That nice property is the following.

Definition 4.8. *We say that an election system $\mathcal{E}$ is* constructive (destructive) vulnerability-combining *if for each variant $C_1+\cdots+C_k$ of multiprong control (so $1 \leq k \leq 5$ and each C_i is a basic control type) it holds that if for all i, $1 \leq i \leq k$, $\mathcal{E}$ is vulnerable to constructive (destructive) C_i control, then $\mathcal{E}$ is vulnerable to constructive (destructive) $C_1 + \cdots + C_k$ control.*

For systems that are constructive (destructive) vulnerability-combining, it is straightforward to see that Theorem 4.7 can be used to read off all $2^5 - 1$ multiprong cases, given just the status of the five underlying prongs.

But how can one in practice establish that a system is constructive (destructive) vulnerability-combining? One could try it by brute force, looking at each collection of vulnerable prongs. However, there is a better path to follow. One can look at all the vulnerable prongs together, and prove (if it happens to be the case) that they yield vulnerability. Note that doing so successfully implies immediately that vulnerability holds for every nonempty subset of those prongs. That this is true follows from the following claim.

Proposition 4.9. *Let $\mathcal{E}$ be an election system and let $C_1, \ldots, C_k$, $1 \leq k \leq 5$, be a collection of basic control types. If for each i, $1 \leq i \leq k$, $\mathcal{E}$ is susceptible to constructive (destructive) C_i control, and $\mathcal{E}$ is vulnerable to constructive (destructive) $C_1 + \cdots + C_k$ control, then for any nonempty subset K of $\{C_1, \ldots, C_k\}$, $\mathcal{E}$ is vulnerable to the constructive (destructive) multiprong control involving exactly the prongs from K.*

Proof. Let the notation be as in the statement of the proposition, and let K be some nonempty subset of $\{C_1, \ldots, C_k\}$. By Theorem 4.3, $\mathcal{E}$ is susceptible to constructive (destructive) multiprong control by the constructive (destructive) control type consisting of exactly the prongs from K. By $\mathcal{E}$'s vulnerability to constructive (destructive) $C_1 + \cdots + C_k$ control, we have a polynomial-time algorithm for constructive (destructive) multiprong control involving exactly the prongs from K: It suffices to use the algorithm for constructive (destructive) $C_1 + \cdots + C_k$ multiprong control, with the bounds on the extent to which nonoccurring prongs can be used set to 0. □

Motivated by the above discussion and by Proposition 4.9, in Sections 4.3 and 5 we will show that plurality, Condorcet, Copeland$^\alpha$ (for each rational α, $0 \leq \alpha \leq 1$), approval, and maximin are indeed constructive vulnerability-combining and destructive vulnerability-combining. Thus for these systems we will have analyzed all $2^5 - 1$ constructive cases and all $2^5 - 1$ destructive cases of multiprong control. Namely, for a constructive (destructive) vulnerability-combining election system $\mathcal{E}$ and for any variant $C_1 + \cdots + C_k$, $1 \leq k \leq 5$, of constructive (destructive) multiprong control such that for each i, $1 \leq i \leq k$, $\mathcal{E}$ is either resistant, vulnerable, or immune to constructive (destructive) C_i control, we can use the following simple rule (which we will refer back to as **Classification Rule A**) to classify constructive (destructive) $C_1 + \cdots + C_k$ multiprong control (note: the correctness of the third part of this classification is where the vulnerability-combining property of $\mathcal{E}$ is being relied on):

1. If $\mathcal{E}$ is resistant to at least one of the control prongs, then it is resistant to the whole attack.

2. If $\mathcal{E}$ is immune to each of the prongs, then it is immune to the whole attack.

3. If neither of the above holds, then $\mathcal{E}$ is vulnerable to the whole attack.

In general, we do not consider partition cases of control in this paper. However, we make an exception for the next example, which shows how even types of control to which a given election system is immune may prove useful in multiprong control. In constructive control by partition of candidates (reminder: this is not a basic control type) in the ties-eliminate model (PC-TE control type), we are given an election $E = (C, V)$ and a preferred candidate $p \in C$, and we ask whether it is possible to find a partition (C_1, C_2) of C (i.e., $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$) such that p is a winner of the following two-round election: We first find the winner sets, W_1 and W_2, of elections (C_1, V) and (C_2, V). If W_1 (W_2) contains more than one candidate, we set $W_1 = \emptyset$ ($W_2 = \emptyset$), since we are in the "ties eliminate" model. The candidates who win election $(W_1 \cup W_2, V)$ are the winners of the overall two-stage election.

Now, let us look at constructive approval-AC+PC-TE control, where (by definition, let us say) we first add new candidates and then perform the partition action. We consider an approval election with two candidates, p and c, where p has 50 approvals and c has 100. We are also allowed to add candidate c', who has 100 approvals. Note that it is impossible to make p a unique winner by adding c'. Exercising the partition action alone does not ensure p's victory either. However, combining both AC and PC-TE does the job. If we first add c' to the election and then partition candidates into $\{p\}$ and $\{c, c'\}$ then, due to the ties-eliminate rule, p becomes the unique winner. It is rather interesting that even though approval is immune to constructive AC control, there are cases where one has to apply AC control to open the possibility of effectively using other types of control.

The above example is perhaps surprising in light of Theorem 4.5. In essence, in the proof of that theorem we argue that if an election system is vulnerable to some basic control type C but is immune to some other basic control type D, then it is also vulnerable to control type $C + D$. We proved the theorem by showing that we can safely disregard the actions of type D (assuming C does not include BV control type). The above example shows that this proof approach would not work if we considered PC-TE in addition to the basic control types.

4.3 Combining Vulnerabilities

In the previous section we considered the case where separate prongs of a multiprong control problem have different computational properties, e.g., some are resistant, some are vulnerable, and some are immune. In this section we consider the case where an election system is vulnerable to each prong separately, and we show how such vulnerabilities combine within election systems for which control results were obtained in previous papers (see Table 5). In particular, in the next theorem we show that for all the election systems considered by Bartholdi et al. (1992), Hemaspaandra et al. (2007), and Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a), all constructive vulnerabilities to AC, DC, AV, DV, and BV combine to vulnerabilities, and all destructive vulnerabilities to AC, DC, AV, DV, BV combine to vulnerabilities.[11] Using this (and some additional discussion to correctly handle the possibility that P = NP), we will soon conclude that each election system studied in these three papers is both constructive vulnerability-combining and destructive vulnerability-combining.

Theorem 4.10. *(a) Plurality is vulnerable to both constructive AV+DV+BV control and destructive AV+DV+BV control. (b) Both Condorcet and approval are vulnerable to AC+AV+DV+BV destructive control. (c) For each rational α, $0 \leq \alpha \leq 1$, Copeland$^\alpha$ is vulnerable to destructive AC+DC control.*[12]

Proof. (a) Let us consider an instance I of constructive plurality-AV+DV+BV control where we want to ensure candidate p's victory: It is enough to add all the voters who vote for p (or as many as we are allowed) and then, in a loop, keep deleting voters who vote for a candidate other than p with the highest score, until p is the only candidate with the highest score or we have exceeded our limit of voters to delete. Finally, in a loop, keep bribing voters who vote for a candidate other than p with the highest score to vote for p, until p is the only candidate with the highest score or we have exceeded our limit of voters to bribe. If p becomes a unique winner via this procedure, then accept. Otherwise reject. We omit the straightforward proof for the destructive case.

(b) Let I be an instance of destructive Condorcet-AC+AV+DV+BV, where our goal is to prevent candidate p from being a Condorcet winner (we assume that p is a Condorcet winner before any control action is performed). It is enough to ensure that some candidate c wins a head-to-head contest with p. Our algorithm works as follows.

Let C be the set of candidates originally in the election and let A be the set of candidates that we can add (we take $A = \emptyset$ if we are not allowed to add any candidates). For each $c \in (C \cup A) - \{p\}$ we do the following:

1. Add as many voters who prefer c to p as possible.

11. Constructive bribery for plurality and constructive bribery for approval have been considered by Faliszewski, Hemaspaandra, and Hemaspaandra (2009a) and constructive and destructive bribery for Copeland has been studied by Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a). In Theorem 4.10 we—in effect—give polynomial-time algorithms for destructive bribery in plurality, approval, and Condorcet. Constructive Condorcet-BV is NP-complete and this is implicitly shown by Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a, Theorem 3.2).
12. Regarding the types among AC, DC, AV, DV, and BV not mentioned in each part of the theorem, plurality is resistant to constructive and destructive AC and DC, Condorcet and approval are immune to constructive AC and destructive DC, and for each rational α, $0 \leq \alpha \leq 1$, Copeland$^\alpha$ is resistant to all five constructive basic types of control and to destructive AV, DV, and BV (see Table 5 for references).

2. Delete as many voters who prefer p to c as possible.

3. Among the remaining voters who prefer p to c, bribe as many as possible to rank c first.

If after these actions c wins his or her head-to-head contest with p then we accept. If no $c \in (C \cup A) - \{p\}$ leads to acceptance, then we reject. It is straightforward to see that this algorithm is correct and runs in polynomial time. (We point out that it is enough to add only a single candidate, the candidate c that prevents p from winning, if he or she happens to be a member of A).

For the case of approval, our algorithm works similarly, except the following differences: We add voters who approve of c but not of p. We delete voters who approve of p but not of c. For each remaining voter v_i, if we still have not exceeded our bribing limit, if v_i approves of p but not of c, we bribe v_i to reverse approvals on p and c. (Note that if we do not exceed our bribing limit by this procedure, this means that each voter that approves of p also approves of c and thus p is not a unique winner.) If these actions lead to p not being a unique winner, we accept. If we do not accept for any $c \in (C \cup A) - \{p\}$, we reject.

(c) The idea is to combine Copeland$^\alpha$ destructive-AC and destructive-DC algorithms (Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009a). We give the full proof for the sake of completeness.

Let us fix a rational value α, $0 \leq \alpha \leq 1$. Given an election E and a candidate c in this election, we write $\mathrm{score}_E^\alpha(c)$ to denote Copeland$^\alpha$ score of c. Let I be an instance of destructive Copeland$^\alpha$-AC+DC control, with an election $E = (C, V)$, where we can add at most k_{AC} spoiler candidates from the set A, and where we can delete at most k_{DC} candidates. Our goal is to ensure that some despised candidate $d \in C$ is not a unique winner. Our algorithm is based on the following simple observation of Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a). For each candidate $c \in C$:

$$\mathrm{score}^\alpha_{(C,V)}(c) = \sum_{c' \in C - \{c\}} \mathrm{score}^\alpha_{(\{c,c'\},V)}(c).$$

Our goal is to prevent candidate d from being a unique winner. If d is not a unique winner, we immediately accept. Otherwise, we seek a candidate $c \in C \cup A$ such that we can ensure that c's score is at least as high as that of d. Thus for each $c \in C \cup A$ we do the following.

1. If $c \in A$, and $k_{\mathrm{AC}} > 0$, we add c to the election (and if $c \in A$ but $k_{\mathrm{AC}} = 0$, we proceed to the next c).

2. As long as we can still add more candidates, we keep executing the following operation: If there is a candidate $c' \in A$ such that value $a(c') = \mathrm{score}^\alpha_{(\{c,c'\},V)}(c) - \mathrm{score}^\alpha_{(\{d,c'\},V)}(d)$ is positive, we add a candidate $c'' \in A$, for whom $a(c'')$ is highest.

3. As long as we can still delete candidates, we keep executing the following operation: If there is a candidate $c' \in C$ such that value $r(c') = \mathrm{score}^\alpha_{(\{d,c'\},V)}(d) - \mathrm{score}^\alpha_{(\{c,c'\},V)}(c)$ is positive, we delete a candidate $c'' \in C$, for whom $r(c'')$ is highest.

4. If after these steps d is not a unique winner, we accept.

If we do not accept for any $c \in C \cup A$, we reject.

It is straightforward to see that we never delete a candidate that we have added. Also, it is straightforward to see that the algorithm works in polynomial time, and that it is correct. Correctness follows from the fact that (a) in the main loop of the algorithm, when dealing with candidate $c \in C \cup A$, each addition of a candidate and each deletion of a candidate increases the difference between the score of c and the score of d as much as is possible, and (b) the order of adding/deleting candidates is irrelevant. □

From the above theorem, the comments preceding it, and a bit of care regarding the possibility that P = NP, we obtain the following claim.

Theorem 4.11. *Plurality, Condorcet, Copeland$^\alpha$ (for each rational α, $0 \leq \alpha \leq 1$), and approval are both constructive vulnerability-combining and destructive vulnerability-combining.*

Proof. Immune prongs are never vulnerable. If P $\neq$ NP then resistant prongs cannot be vulnerable, and we are already done by the previous theorem. If P = NP, then it is possible that resistant prongs are also vulnerable. Indeed, for all the systems under discussion in this proof, all their resistant prongs under basic control types happen to be in NP and are vulnerable if P = NP. However, it is straightforward to see that if P = NP, then for these particular election systems it holds that all of their vulnerable constructive (destructive) basic prongs—which for these will in that case be all their nonimmune basic constructive (destructive) prongs—when combined yield a multiprong constructive (destructive) control type for which vulnerability holds. □

As established by Theorem 4.11 and the (soon-to-come) results of Section 5, all natural election systems that we have discussed so far in this paper are vulnerability-combining both in the constructive setting and in the destructive setting. It is natural to wonder whether this is a necessary consequence of our model of multiprong control or whether in fact there is an election system for which combining two control types to which the system is vulnerable yields a multipronged control problem to which the system is resistant. Theorem 4.12 shows that the latter is the case, even for a natural election system.

In the thirteenth century, Ramon Llull proposed an election system that could be used to choose popes and leaders of monastic orders (see Hägele & Pukelsheim, 2001; McLean & Lorrey, 2006). In his system, voters choose the winner from among themselves (so, the candidates are the same as the voters). Apart from that, Llull's voting system is basically Copeland^1, the version of Copeland that most richly rewards ties. Formally, we define the voting system OriginalLlull as follows: For an election $E = (C, V)$, if the set of names of V, which we will denote by names(V), is not equal to C, then there are no winners. Otherwise, a candidate $c \in C$ is a winner if and only if it is a Copeland^1 winner. Note that single-prong AC and AV control for OriginalLlull do not make all that much sense, and so it should come as no surprise that OriginalLlull is vulnerable to both constructive AC control and constructive AV control. In addition, we will show (by renaming and padding) that Copeland^1-AV can be be reduced to OriginalLlull^1-AC+AV. Since Copeland^1 is resistant to constructive control by adding voters (Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009a), this then leads to the following theorem.

Theorem 4.12. OriginalLlull *is vulnerable to both constructive AC control and constructive AV control but is resistant to constructive AC+AV control.*

Proof. It is immediate that OriginalLlull is susceptible to constructive AC, AV, and (by Theorem 4.3) AC+AV control. It is also straightforward to see that constructive OriginalLlull-AC (AV) control is in P: If possible add candidates (voters) such that the set of voter names is equal to the set of candidates, and then check if the preferred candidate is a unique Copeland^1 winner. If this is not possible, reject.

We will now show, via a reduction from constructive Copeland^1-AV control (which is NP-hard by Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009a) that constructive OriginalLlull-AC+AV control is NP-hard. Let C be a set of candidates, let V and W be two disjoint collections of voters with preference lists over C, let $p \in C$ be the preferred candidate, and k be an element of $\mathbb{N}$. The question is whether there exists a subcollection $W' \subseteq W$ of size at most k such that p is a unique Copeland^1 winner of $(C, V \cup W')$. Without loss of generality, we assume that V is not empty.

We will now show how to pad this election. For an OriginalLlull election to be nontrivial, we certainly need to have the same number of candidates as voters (later, we will also rename the voters so that they are the same as the candidates). If $\|V\| < \|C\|$, we want to add a collection of new dummy voters V' such that $\|V\| + \|V'\| = \|C\|$ and such that adding V' to an election does not change the relative Copeland^1 scores of the candidates. This can be accomplished by letting half of the voters in V' vote C (recall Convention 3.1) and half of the voters in V' vote $\overleftarrow{C}$. Of course, this can only be done if $\|V'\|$ is even.

So, we will do the following. If $\|V\| < \|C\|$, we add a collection of new voters V' such that $\|V'\| = \|C\| - \|V\|$ if $\|C\| - \|V\|$ is even, and $\|V'\| = \|C\| - \|V\| + 1$ if $\|C\| - \|V\|$ is odd. If $\|V\| \geq \|C\|$, we let $V' = \emptyset$. Half of the voters in V' vote C and half of the voters in V' vote $\overleftarrow{C}$. In addition, we introduce a set A of new candidates such that $\|C\| + \|A\| = \|V\| + \|V'\| + \|W\|$. Note that this is always possible, since $\|V\| + \|V'\| \geq \|C\|$. We extend the votes of the voters (in V, V', and W) to $C \cup A$ by taking their preference order on C and following this by the candidates in A in some fixed, arbitrary order. Note that this will have the effect that candidates in A will never be winners.

Let $W' \subseteq W$, $A' \subseteq A$, $E = (C, V \cup W')$, $E' = (C \cup A', V \cup V' \cup W')$. It is straightforward to see that the following hold (recall that V is not empty).

1. For all $d \in A'$, $\mathrm{score}^1_{E'}(d) \leq \|A'\| - 1$.
2. For all $c \in C$, $\mathrm{score}^1_{E'}(c) = \mathrm{score}^1_{E}(c) + \|A'\|$.
3. For all $c, c' \in C, c \neq c'$, $\mathrm{score}^1_{E}(c) - \mathrm{score}^1_{E}(c') = \mathrm{score}^1_{E'}(c) - \mathrm{score}^1_{E'}(c')$.
4. p is a unique Copeland^1 winner of E if and only if p is a unique Copeland^1 winner of E'.

We are now ready to define the reduction. Name the voters such that $\mathrm{names}(V \cup V') \supseteq C$ and $\mathrm{names}(V \cup V' \cup W) = C \cup A$. Then map (C, V, W, p, k) to $(C, A, V \cup V', W, p, \|A\|, k)$. We claim that p can be made a unique Copeland^1 winner of (C, V) by adding at most k voters from W if and only if p can be made a unique OriginalLlull winner of $(C, V \cup V')$ by adding (an unlimited number of) candidates from A and at most k voters from W.

First suppose that W' is a subcollection of W of size at most k such that p is the unique Copeland^1 winner of $(C, V \cup W')$. Let $A' \subseteq A$ be the set of candidates such that

$C \cup A' = \text{names}(V \cup V' \cup W')$. By item 4 above, p is the unique Copeland^1 winner of $(C \cup A', V \cup V' \cup W')$, and thus p is the unique OriginalLlull winner of $(C \cup A', V \cup V' \cup W')$.

For the converse, suppose that there exist $A' \subseteq A$ and $W' \subseteq W$ such that $\|W'\| \leq k$, and p is the unique OriginalLlull winner of $(C \cup A', V \cup V' \cup W')$. Then p is the unique Copeland^1 winner of $(C \cup A', V \cup V' \cup W')$, and, by item 4, p is the unique Copeland^1 winner of $(C, V \cup W')$.

Thus our reduction is correct and, since it can be computed in polynomial time, the proof is complete. □

We have the following corollary, since OriginalLlull is neutral (permuting the names of the candidates, evaluating the election's winners, and then running the winners through the inverse of the permutation does not affect the outcome of the election) and anonymous (permuting the names of the voters does not affect the outcome of the election).[13]

Corollary 4.13. *There exists a neutral and anonymous election system $\mathcal{E}$ such that $\mathcal{E}$ is vulnerable to both constructive AC control and constructive AV control but is resistant to constructive AC+AV control.*

Something might seem a bit strange about Theorem 4.12. After all, it says that a powerful chair—one who can *both* add candidates and add voters—faces a harder task than would be faced by a weaker chair—say, one who can just add candidates. The important thing to keep in mind, to understand why this is not strange, is that the different chairs are facing (correspondingly) different problems. The powerful type of chair is being asked to determine whether by specified amounts of adding candidates and voters a goal can be met, and the weaker type of chair is being asked whether by a specified amount of adding candidates a goal can be met. Thus it is not at all paradoxical for the former problem to have higher complexity than the latter. (After all, a "powerful" solution-finder—one allowed to use any assignment—seeking to find a satisfying assignment to input Boolean formulas is facing an NP-hard task, but a "weak" solution-finder—one only allowed to find solutions with at most 2011 variables assigned "False"—seeking to find for input Boolean formulas a satisfying assignment, when such exists, that has at most 2011 variables assigned "False" faces just a polynomial-time task, due to the natural brute-force approach.)

4.4 Summary

We now summarize the main contributions of Section 4. We will not try here to motivate or interpret our results, but rather will summarize our results "as they are," so the reader can easily jump back to this section for reference.

13. The notion of anonymity just stated is the standard one in the literature. To avoid any confusion, we mention that an earlier version (Faliszewski, Hemaspaandra, & Hemaspaandra, 2010a) of this paper used a much stronger definition of anonymity—one that required one to be able to not just permute the voter names but to one-to-one map them to any set of names and yet have the outcome not change. Let us call that notion voter-superanonymity. OriginalLlull does not satisfy that stronger notion. However, the earlier version of this paper—by sneakily building the preference orders of the voters into the names of the candidates—constructed a highly artificial system that was neutral, anonymous (in the sense of the present paper), and voter-superanonymous, and that had two vulnerable prongs that when combined yielded resistance (Faliszewski et al., 2010a). In contrast, Theorem 4.12/Corollary 4.13 provides such a jump from vulnerability to resistance for a preexisting, natural voting system that is neutral and anonymous.

C_1	C_2	$C_1 + C_2$
R	R	R
R	V	R
R	I	R
I	I	I
I	V	V
V	V	susceptible (i.e., not I)

Table 2: An example of applying Theorem 4.7. Let $\mathcal{E}$ be some election system. We consider two basic types of constructive (destructive) control, C_1 and C_2, to which $\mathcal{E}$ is either resistant (R), immune (I), or vulnerable (V). The table shows how $C_1 + C_2$ control for $\mathcal{E}$ inherits properties from its prongs. Note that Theorem 4.7 does not on its own give results for the case when $\mathcal{E}$ is vulnerable to both C_1 and C_2 (although from Theorem 4.3 susceptibility must hold). To see this, note that for vulnerability-combining systems putting together two vulnerable prongs leads to a two-pronged control problem to which the system is vulnerable. But there are examples of voting systems where combining two vulnerable prongs leads to a resistant two-pronged control problem (see Theorem 4.12).

In Section 4.1, we introduced our model of multiprong control, which allows the attacker to use several types of control jointly, either to try to make a given candidate a unique winner (constructive control), or to try to prevent a given candidate from being a unique winner (destructive control).

In Section 4.2 we focused on the following issue: Let $\mathcal{E}$ be an election system. Let $C_1 + \cdots + C_k$ be a variant of multiprong control (so $1 \leq k \leq 5$ and each C_i is a basic control type). Suppose we know, for each C_i, $1 \leq i \leq k$, whether $\mathcal{E}$ is resistant, immune, or vulnerable to constructive (destructive) control type C_i. To what extent can we from this alone tell whether $\mathcal{E}$ is resistant, immune, or vulnerable to constructive (destructive) $C_1 + \cdots + C_k$ multiprong control? Theorem 4.7 and the paragraphs following it provide a detailed answer. As an example of applying Theorem 4.7, in Table 2 we show how properties of two control prongs, C_1 and C_2, combine into properties of two-pronged $C_1 + C_2$ control.

Simply stated, Theorem 4.7 (for those systems where each prong happens to be immune or vulnerable or resistant—and that should include essentially all reasonable, natural election systems having polynomial-time winner problems) gives a read-off-the-answer classification for all the multiprong cases, except if the system has multiple vulnerable constructive prongs or has multiple vulnerable destructive prongs. And to handle that case, the case of systems (where each prong happens to be immune or vulnerable or resistant) having multiple vulnerable constructive (destructive) prongs, we defined the notion of constructive (destructive) vulnerability-combining election systems, and provided as Classification Rule A a simple rule that classifies *all* multiprong control cases. We also provided a handy tool, Proposition 4.9, for proving that an election system is vulnerability-combining. We conjecture that almost all natural systems are vulnerability-combining.

Finally, in Section 4.3 we considered several natural election systems for which control has already been studied (namely, plurality, Condorcet, Copeland$^\alpha$ (for each rational α,

Election system	Our results that combine into multiprong vulnerability all immune and all vulnerable entries in each column of Table 5	
	Constructive	Destructive
Plurality	AV+DV+BV	AV+DV+BV
Condorcet	AC+DC	AC+DC+AV+DV+BV
Copeland$^\alpha$, for each $0 \leq \alpha \leq 1$	(none)	AC+DC
Approval	AC+DC	AC+DC+AV+DV+BV
Maximin	DC	AC+DC

Table 3: Summary of our vulnerability results for multiprong control for plurality, Condorcet, Copeland$^\alpha$, and approval voting systems. Regarding the five basic control types, these summarize that for each column of Table 5 that has at least one vulnerable entry, we have established that the multiprong combination of all the immunity and vulnerability entries in that column remains vulnerable. These results are obtained by combining Theorem 4.10 and Theorem 4.7 (with Theorem 4.7 letting us add in basic control types to which a given system is immune). For the sake of completeness, we have also included the analogous results regarding maximin, from Section 5.

$0 \leq \alpha \leq 1$), and approval), and for each of these systems we proved that the system is both constructive vulnerability-combining and destructive vulnerability-combining (see Theorem 4.11 for the formal result; but Table 3 summarizes the actual combinations that are in play here unless P = NP). Nonetheless, in Theorem 4.12 we have shown that the ancient election system OriginalLlull has two vulnerable prongs that combine to yield resistance; unless P = NP, OriginalLlull is not vulnerability-combining.

5. Control in Maximin

In this section we initiate the study of control in the maximin election system. Maximin is loosely related to Copeland$^\alpha$ voting in the sense that both are defined in terms of the pairwise head-to-head contests. In addition, the unweighted coalitional manipulation problem for maximin and Copeland$^\alpha$ ($\alpha \neq 0.5$) exhibits the same behavior: It is in P for one manipulator and NP-complete for two or more manipulators (Xia, Zuckerman, Procaccia, Conitzer, & Rosenschein, 2009; Faliszewski, Hemaspaandra, & Schnoor, 2008, 2010). Thus one might wonder whether both systems will be similar with regard to their resistances to control. In fact, there are very interesting differences.

It is straightforward to see that maximin is susceptible to all basic types of constructive and destructive control. And so, by Theorem 4.3, to show vulnerability to constructive (destructive) $\mathcal{C}$ control it suffices to give a polynomial-time algorithm that decides the constructive (destructive) $\mathcal{E}$-$\mathcal{C}$ problem, and to show resistance to constructive (destructive) $\mathcal{C}$ control it suffices to show that the constructive (destructive) $\mathcal{E}$-$\mathcal{C}$ problem is NP-hard.

As a consequence of the analysis given in the following subsections, we have that maximin is both constructive vulnerability-combining and destructive vulnerability-combining.

Theorem 5.1. *Maximin is both constructive vulnerability-combining and destructive vulnerability-combining.*

Proof. The same discussion and analysis provided in the proof of Theorem 4.11 apply here, except relying on the underpinning work provided later in this section. The constructive case here is degenerate, as among basic prongs there is only one that is of interest (although we will briefly discuss a special "extra" prong later). Even for the destructive case, there are only two basic prongs that are not resistant. $\square$

5.1 Candidate Control in Maximin

Let us now focus on candidate control in maximin, that is, on the AC and the DC control types, both in the constructive and in the destructive setting. As is the case for Copeland$^\alpha$, $0 \leq \alpha \leq 1$, maximin is resistant to control by adding candidates.

Theorem 5.2. *Maximin is resistant to constructive AC control.*

Proof. We give a reduction from X3C. Let $(B, \mathcal{S})$, where $B = \{b_1, \ldots, b_{3k}\}$ is a set of $3k$ elements and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a set of 3-subsets of B, be our input X3C instance. We form an election $E = (C \cup A, V)$, where $C = B \cup \{p\}$, $A = \{a_1, \ldots, a_n\}$, and $V = (v_1, \ldots, v_{2n+2})$. (Candidates in A are the spoiler candidates, which the attacker has the ability to add to election (C, V).)

Voters in V have the following preferences. For each $S_i \in \mathcal{S}$, voter v_i reports preference list $p > B - S_i > a_i > S_i > A - \{a_i\}$ and voter v_{n+i} reports preference list $\overleftarrow{A - \{a_i\}} > a_i > \overleftarrow{S_i} > \overleftarrow{B - S_i} > p$. Voter v_{2n+1} reports $p > A > B$ and voter v_{2n+2} reports $\overleftarrow{B} > p > \overleftarrow{A}$.

We claim that there is a set $A' \subseteq A$ such that $\|A'\| \leq k$ and p is a unique winner of $(C \cup A', V)$ if and only if $(B, \mathcal{S})$ is a "yes" instance of X3C.

To show the claim, let $E' = (C, V)$. For each pair of distinct elements $b_i, b_j \in B$, we have that $N_{E'}(b_i, b_j) = n+1$, $N_{E'}(p, b_i) = n+1$, and $N_{E'}(b_i, p) = n+1$. That is, all candidates in E' tie. Now consider some set $A'' \subseteq A$, $\|A''\| \leq k$, and an election $E'' = (C \cup A'', V)$. Values of $N_{E''}$ and $N_{E'}$ are the same for each pair of candidates in $\{p\} \cup B$. For each pair of distinct elements $a_i, a_j \in A''$, we have $N_{E''}(p, a_i) = n + 2$, $N_{E''}(a_i, p) = n$, and $N_{E''}(a_i, a_j) = n + 1$. For each $b_i \in B$ and each $a_j \in A''$ we have that

$$N_{E''}(b_i, a_j) = \begin{cases} n & \text{if } b_i \in S_j, \\ n+1 & \text{if } b_i \notin S_j, \end{cases}$$

and, of course, $N_{E''}(a_j, b_i) = 2n + 2 - N_{E''}(b_i, a_j)$. Thus, by definition of maximin, we have the following scores in E'': (a) $\text{score}_{E''}(p) = n + 1$, (b) for each $a_j \in A''$, $\text{score}_{E''}(a_j) = n$, and (c) for each $b_i \in B$,

$$\text{score}_{E''}(b_i) = \begin{cases} n & \text{if } (\exists a_j \in A'')[b_i \in S_j], \\ n+1 & \text{otherwise.} \end{cases}$$

A'' corresponds to a family S'' of 3-sets from $\mathcal{S}$ such that for each j, $1 \leq j \leq n$, S'' contains set S_j if and only if A'' contains a_j. Since $\|A''\| \leq k$, it is straightforward to see that p is a unique winner of E'' if and only if S'' is an exact cover of B. $\square$

Copeland$^\alpha$, $0 \leq \alpha \leq 1$, is resistant to constructive AC control, but for $\alpha \in \{0, 1\}$, Copeland$^\alpha$ is vulnerable to constructive control by adding an unlimited number of candidates. It turns out that so is maximin. However, interestingly, in contrast to Copeland, maximin is also vulnerable to DC control.

Rather than just proving that, we will prove a bit more: We will just for this moment discuss a different control type, $\mathrm{AC_u}$, that we will *not* consider to be one of the five basic types. $\mathrm{AC_u}$ control (called control by adding an unlimited number of candidates) is just like control by adding candidates, except there (by definition) is no limit on the number of candidates to add—in effect, one requires $k_{\mathrm{AC}} = \|A\|$.[14] We will show that maximin is vulnerable not just to DC control but in fact even to $\mathrm{AC_u}$+DC control. Intuitively, in constructive $\mathrm{AC_u}$+DC control we should add as many candidates as possible (because adding a candidate generally decreases other candidates' scores, making our preferred candidate's way to victory easier) and then delete those candidates who stand in our candidate's way (i.e., those whose existence blocks the preferred candidate's score from increasing). Studying constructive $\mathrm{AC_u}$+DC control for maximin jointly leads to a compact, coherent algorithm. If we were to consider both control types separately, we would have to give two fairly similar algorithms while obtaining a weaker result.

Theorem 5.3. *Maximin is vulnerable to constructive AC_{u}+DC control.*

Proof. We give a polynomial-time algorithm for constructive maximin-$\mathrm{AC_u}$+DC control. The input contains an election $E = (C, V)$, a set of spoiler candidates A, a preferred candidate $p \in C$, and a nonnegative integer k_{DC}. Voters in V have preference lists over the candidates in $C \cup A$. We ask whether there exist sets $A' \subseteq A$ and $C' \subseteq C$ such that (a) $\|C'\| \leq k_{\mathrm{DC}}$ and (b) p is a unique winner of election $((C - C') \cup A', V)$. If $k_{\mathrm{DC}} \geq \|C\| - 1$, we accept immediately because we can delete all candidates but p. Otherwise, we use the following algorithm.

Preparation. We rename the candidates in C and A so that $C = \{p, c_1, \ldots, c_m\}$ and $A = \{c_{m+1}, \ldots, c_{m+m'}\}$. Let $E' = (C \cup A, V)$ and let $P = \{N_{E'}(p, c_i) \mid c_i \in C \cup A\}$. That is, P contains all the values that candidate p may obtain as scores upon deleting some candidates from E'. For each $k \in P$, let $Q(k) = \{c_i \mid c_i \in C \cup A - \{p\} \wedge N_{E'}(p, c_i) < k\}$. Intuitively, $Q(k)$ is the set of candidates in E' that prevent p from having at least k points.

14. The control type $\mathrm{AC_u}$ is of some historical interest as it was used as the control by adding candidates notion in the seminal paper on control. However, following a suggestion of Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2007), AC has been used in most work in recent years; it is the more natural choice since it is analogous to the other three add/delete voter/candidate control types. In addition to Theorem 5.3's result about the constructive case, we mention in passing that maximin's vulnerability to destructive AC (see Theorem 5.4 in light of Proposition 4.9) implies that it is also vulnerable to $\mathrm{AC_u}$. Readers wishing to know what results hold for the prong $\mathrm{AC_u}$ for each of the election systems covered in this paper can find that summarized in a table of Faliszewski et al. (2010a)—except for the $\alpha \neq 0.5$ cases of Copeland$^\alpha$, and for those cases Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a) can be referred to. Our entire set of tools in Section 4.2 is framed around the five basic control types, so we do not here try to weave in $\mathrm{AC_u}$ into the framework. We mention that tools apply well to this type also, and so it would prove no special hurdle to incorporate it into the framework. Still, we feel that AC (not $\mathrm{AC_u}$) is by far the more attractive way to frame control by adding candidates and so such an addition is not compelling.

Main loop. For each $k \in P$, our algorithm tests whether by deleting at most k_{DC} candidates from C and any number of candidates from A it is possible to ensure that p obtains exactly k points and becomes a unique winner of E'. Let us fix some value $k \in P$. We build a set D of candidates to delete. Initially, we set $D = Q(k)$. It is straightforward to see that deleting candidates in $Q(k)$ is a necessary and sufficient condition for p to have score k. However, deleting candidates in $Q(k)$ is not necessarily sufficient to ensure that p is a unique winner because candidates with scores greater or equal to k may exist. We execute the following loop (which we will call the *fixing loop*):

1. Set $E'' = ((C \cup A) - D, V)$.
2. Pick a candidate $d \in (C \cup A) - D$ such that $\mathrm{score}_{E''}(d) \geq k$ (break from the loop if no such candidate exists).
3. Add d to D and jump back to Step 1.

We accept if $C \cap D \leq k_{\mathrm{DC}}$ and we proceed to the next value of k otherwise.[15] If none of the values $k \in P$ leads to acceptance then we reject.

Let us now briefly explain why the above algorithm is correct. It is straightforward to see that in maximin adding some candidate c to an election does not increase other candidates' scores, and deleting some candidate d from an election does not decrease other candidates' scores. Thus, if after deleting candidates in $Q(k)$ there still are candidates other than p with k points or more, the only way to ensure p's victory—without explicitly trying to increase p's score—is by deleting those candidates. Also, note that the only way to ensure that p has exactly k points is by deleting candidates $Q(k)$.

Note that during the execution of the fixing loop, the score of p might increase to some value $k' > k$. If that happens, it means that it is impossible to ensure p's victory while keeping his or her score equal to k. However, we do not need to change k to k' in that iteration of the main loop as we will consider k' in a different iteration. $\square$

Maximin is also vulnerable to destructive AC+DC control. The proof relies on the fact that (a) if there is a way to prevent a despised candidate from winning a maximin election via adding some spoiler candidates then there is a way to do so by adding at most two candidates, (b) adding a candidate cannot increase the score of any candidate other than the added one, and (c) deleting a candidate cannot decrease the score of any candidate other than the deleted one. In essence, the algorithm performs a brute-force search for the candidates to add and then uses the constructive maximin-DC control algorithm from Theorem 5.3.

Theorem 5.4. *Maximin is vulnerable to destructive AC+DC control.*

Proof. We remind the reader that part of the definition of destructive control by deleting candidates is that one cannot simply delete one's despised candidate.

15. If we accept, D implicitly describes the control action that ensures p's victory: We should delete from C the candidates in $C \cap D$ and add from A the candidates in $A - D$.

We will first give an algorithm for destructive maximin-AC and then argue how it can be combined with the algorithm from Theorem 5.3 to solve destructive maximin-AC+DC in polynomial time.

Let us first focus on the destructive AC problem. Our input is an election $E = (C, V)$, where $C = \{d, c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$, a spoiler candidate set $A = \{c_{m+1}, \ldots, c_{m'}\}$, and a nonnegative integer k_{AC}. The voters have preference orders over $C \cup A$. The goal is to ensure that d is not a unique winner of E via adding at most k_{AC} candidates from A.

Let us assume that there exists a set $A' \subseteq A$ such that d is not a unique winner of election $E' = (C \cup A', V)$. Since d is not a unique winner of E', there exists some candidate $c' \in C \cup A'$ such that $\mathrm{score}_{E'}(c') \geq \mathrm{score}_{E'}(d)$. Also, by definition of maximin, there is some candidate $d' \in C \cup A'$ such that $\mathrm{score}_{E'}(d) = N_{E'}(d, d')$. As a consequence, d is not a unique winner of election $E'' = (C \cup \{c', d'\}, V)$. The reason is that $\mathrm{score}_{E''}(d) = \mathrm{score}_{E'}(d)$ (because both E' and E'' contain d') and $\mathrm{score}_{E''}(c') \geq \mathrm{score}_{E'}(c')$ (because adding the remaining $A' - \{c', d'\}$ candidates to E'' does not increase c''s score). Thus, to test whether it is possible to ensure that d is not a unique winner of E, it suffices to test whether there is a set $A'' \subseteq A$ such that $\|A''\| \leq \min(2, k_{\mathrm{AC}})$ and d is not a unique winner of $(C \cup A'', V)$. Note that this test can be carried out in polynomial time.

Let us now consider the AC+DC case. The input and the goal are the same as before, except that now we are also given a nonnegative integer k_{DC} and we are allowed to delete up to k_{DC} candidates. We now describe our algorithm. For each set $\{c', d'\}$ of up to two candidates, $\{c', d'\} \subseteq (C \cup A) - \{d\}$ we execute the following steps.

1. We check if $\|A \cap \{c', d'\}\| \leq k_{\mathrm{AC}}$ (and we proceed to the next $\{c', d'\}$ if this is not the case).

2. We compute a set $D \subseteq C - \{d, c', d'\}$, $\|D\| \leq k_{\mathrm{DC}}$, that maximizes $\mathrm{score}_{E'}(c')$, where $E' = ((C \cup \{c', d'\}) - D, V)$.

3. If d is not a unique winner of $E' = ((C \cup \{c', d'\}) - D, V)$, we accept.

We reject if we do not accept for any $\{c', d'\} \subseteq (C \cup A) - \{d\}$.

The intended role of d' is to lower the score of d and keep it at a fixed level, while, of course, the intended role of c' is to defeat d. By reasoning analogous to that for the AC case, we can see that there is no need to add more than two candidates. Thus, given $\{c', d'\}$, it remains to compute the appropriate set D. In essence, we can do so in the same manner as in the constructive AC+DC case.

Let k be some positive integer. We set $D(k) = \{c_i \in C - \{c', d', d\} \mid N_E(c', c_i) < k\}$ and we pick $D = D(i)$, where i is as large as possible (but no larger than $\|V\|$) and $\|D\| \leq k_{\mathrm{DC}}$. Deleting candidates in D maximizes the score of c', given that we cannot delete d and d' (we cannot delete d by definition of control by deleting candidates, and we cannot—or, more precisely, do not want to—delete d' because the role of d' in our algorithm is to keep the score of d in check). It is straightforward to see that this D can be computed in polynomial time. ☐

5.2 Control by Adding and Deleting Voters in Maximin

In this section we consider the complexity of constructive and destructive AV and DV control types. (We will consider bribery, BV, in the next section; recall that in this paper, bribery is a basic control type, though it is usually treated separately in the literature.) In the previous section we have seen that maximin is vulnerable to all basic types of constructive and destructive candidate control except for constructive control by adding candidates (constructive AC control). The situation regarding voter control is quite different: As shown in the next three theorems, maximin is resistant to all basic types of constructive and destructive voter control.

Theorem 5.5. *Maximin is resistant to constructive and destructive AV control.*

Proof. We will first give an NP-hardness proof for the constructive case and then we will describe how to modify it for the destructive case.

We now give a reduction of the X3C problem to the constructive maximin-AV problem. Our input X3C instance is $(B, \mathcal{S})$, where $B = \{b_1, \ldots, b_{3k}\}$ is a set of $3k$ distinct elements and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a family of n 3-element subsets of B. Without loss of generality, we assume $k \geq 1$. Our reduction outputs the following instance. We have an election $E = (C, V)$, where $C = B \cup \{p, d\}$ and $V = (v_1, \ldots, v_{4k})$. There are $2k$ voters with preference order $d > B > p$, k voters with preference order $p > B > d$, and k voters with preference order $p > d > B$. In addition, we have a collection $W = (w_1, \ldots, w_n)$ of voters who can be added, where the i'th voter, $1 \leq i \leq n$, has preference order

$$B - S_i > p > S_i > d.$$

We claim that there is a subcollection $W' \subseteq W$ such that $\|W'\| \leq k$ and p is a unique winner of election $(C, V \cup W')$ if and only if $(B, \mathcal{S})$ is a "yes" instance of X3C.

It is straightforward to verify that for each $b_i \in B$ it holds that $N_E(p, b_i) = 2k$, and that $N_E(p, d) = 2k$. Thus $\text{score}_E(p) = 2k$. Similarly, it is straightforward to verify that $\text{score}_E(d) = 2k$, and that for each $b_i \in B$, $\text{score}_E(b_i) \leq k$. Let W'' be a subcollection of W such that $\|W''\| \leq k$ and let $E'' = (C, V \cup W'')$. For each $b_i \in B$ it holds that $\text{score}_{E''}(b_i) \leq 2k$. Since each voter in W ranks d as the least desirable candidate, $\text{score}_{E''}(d) = 2k$. What is p's score in election E''? If there exists a candidate $b_i \in B$ such that there is no voter w_j in W'' that prefers p to b_i, then $\text{score}_{E''}(p) = 2k$ (because $N_{E''}(p, b_i) = 2k$). Otherwise, $\text{score}_{E''}(p) \geq 2k + 1$. Thus p is a unique winner of E'' if and only if W'' corresponds to an exact cover of B. This proves our claim and, as the reduction is straightforwardly seen to be computable in polynomial time, concludes the proof for the constructive maximin-AC case.

To show that destructive maximin-AC is NP-hard, we use the same reduction, except that we remove from V a single voter with preference list $p > B > d$, and we set the task to preventing d from being a unique winner. Removing a $p > B > d$ voter from V ensures that before we start adding candidates, d has score $2k$ (and this score cannot be changed), p has score $2k-1$ (and p needs to get one point extra over each other candidate to increase his or her score and prevent d from being a unique winner), and each $b_i \in B$ has score $k-1$ (thus no candidate in B can obtain score higher than $2k-1$ via adding no more than k candidates from W). The same reasoning as for the constructive case proves that the reduction correctly reduces X3C to destructive maximin-AV. ☐

Theorem 5.6. *Maximin is resistant to constructive and destructive DV control.*

Proof. We will first show NP-hardness for constructive maximin-DV control and then we will argue how to modify the construction to obtain the result for the destructive case.

Our reduction is from X3C. Let $(B, \mathcal{S})$ be our input X3C instance, where $B = \{b_1, \ldots, b_{3k}\}$, $\mathcal{S} = \{S_1, \ldots, S_n\}$, and for each i, $1 \leq i \leq n$, $\|S_i\| = 3$. Without loss of generality, we assume that $n \geq k \geq 3$ (if $n < k$ then $\mathcal{S}$ does not contain a cover of B, and if $k \leq 2$ we can solve the problem by brute force). We form an election $E = (C, V)$, where $C = B \cup \{p, d\}$ and where $V = V' \cup V''$, $V' = (v'_1, \ldots, v'_{2n})$, $V'' = (v''_1, \ldots, v''_{2n-k+2})$. For each i, $1 \leq i \leq n$, voter v'_i has preference order

$$d > B - S_i > p > S_i$$

and voter v'_{n+i} has preference order

$$d > \overleftarrow{S_i} > p > \overleftarrow{B - S_i}.$$

Among the voters in V'' we have: 2 voters with preference order $p > d > B$, $n - k$ voters with preference order $p > B > d$, and n voters with preference order $B > p > d$. We claim that it is possible to ensure that p is a unique winner of election E via deleting at most k voters if and only if $(B, \mathcal{S})$ is a "yes" instance of X3C.

Via routine calculation we see that candidates in election E have the following scores:

1. $\text{score}_E(d) = 2n$ (because $N_E(d, p) = 2n$ and for each $b_i \in B$, $N_E(d, b_i) = 2n + 2$),
2. $\text{score}_E(p) = 2n - k + 2$ (because $N_E(p, d) = 2n - k + 2$ and for each $b_i \in B$, $N_E(p, b_i) = 2n - k + 2$), and
3. for each $b_i \in B$, $\text{score}_E(b_i) \leq 2n - k$ (because $N_E(b_i, d) = 2n - k$).

Before any voters are deleted, d is the unique winner with $k - 2$ more points than p. Via deleting at most k voters it is possible to decrease d's score at most by k points. Let W be a collection of voters such that p is the unique winner of $E' = (C, V - W)$. We partition W into $W' \cup W''$, where W' contains those members of W that belong to V' and W'' contains those members of W that belong to V''. We claim that W'' is empty. For the sake of contradiction let us assume that $W'' \neq \emptyset$. Let $E'' = (C, V - W'')$. Since every voter in V'' prefers p to d, we have that $N_{E''}(p, d) = N_E(p, d) - \|W''\|$ and, as a result, $\text{score}_{E''}(p) \leq \text{score}_E(p) - \|W''\|$. In addition, assuming W'' is not empty, it is straightforward to observe that $\text{score}_{E''}(d) \geq \text{score}_E(d) - \|W''\| + 1$ (the reason for this is that deleting any *single* member of V'' does not decrease d's score). That is, we have that:

$$\begin{aligned} \text{score}_{E''}(p) &\leq 2n - k + 2 - \|W''\|, \\ \text{score}_{E''}(d) &\geq 2n + 1 - \|W''\|. \end{aligned}$$

So in E'', d has at least $k - 1$ more points than p. Since $\|W''\| \geq 1$, we can delete at most $k - 1$ voters W' from election E''. But then p will not be a unique winner of E', which is a contradiction.

Thus W contains members of V' only. Since d is ranked first in every vote in V', deleting voters from W decreases d's score by exactly $\|W\|$. Further, deleting voters W certainly decreases p's score by at least one point. Thus after deleting voters W we have:

1. $\text{score}_{E'}(d) = 2n - \|W\|$,
2. $\text{score}_{E'}(p) \leq 2n - k + 2 - 1 = 2n - k + 1$.

In consequence, the only possibility that p is a unique winner after deleting voters W is that $\|W\| = k$ and we have equality in item 2 above. It is straightforward to verify that this equality holds if and only if W contains k voters among $v'_1, \ldots, v'_n$ that correspond to an exact cover of B via sets from $\mathcal{S}$ (recall that $k \geq 3$). This proves that our reduction is correct, and since the reduction is straightforwardly seen to be computable in polynomial time, completes the proof of NP-hardness of constructive maximin-DV control.

Let us now consider the destructive case. Let $(B, \mathcal{S})$ be our input X3C instance (with B and $\mathcal{S}$ as in the constructive case). We form election $E = (C, V)$ which is identical to the one created in the constructive case, except that $V'' = (v''_1, \ldots, v''_{2n-k})$ and we set these voters' preference orders as follows: There is one voter with preference order $p > d > B$, $n - k$ voters with preference order $p > B > d$, and $n - 1$ voters with preference order $B > p > d$. (That is, compared to the constructive case, we remove one voter with preference order $p > d > B$ and one with preference order $B > p > d$.) It is straightforward to see that d is the unique winner of election E and we claim that he or she can be prevented from being a unique winner via deleting at most k voters if and only if there is an exact cover of B by k sets from $\mathcal{S}$.

Via routine calculation, it is straightforward to verify that $\text{score}_E(d) = 2n$, and that $\text{score}_E(p) = 2n - k$. The former holds because $N_E(d, p) = 2n$ and $N_E(d, b_i) = 2n + 1$ and the latter holds because $N_E(p, d) = 2n - k$ and for each candidate $b_i \in B$ we have $N_E(p, b_i) = 2n - k + 1$. In addition, each candidate $b_i \in B$ has score at most $2n - k - 1$. Thus it is possible to ensure that d is not a unique winner via deleting at most k voters if and only if there are exactly k voters the deletion of which would decrease the score of d by k points and would not decrease p's score. Let us assume that such a collection of voters exists and let W be such a collection. Since every voter in V'' prefers p to d, note that W does not contain any voter in V''. Thus W contains exactly k voters from V'. Since for each $b_i \in B$ we have $N_E(p, b_i) = 2n - k + 1$, for each $b_i \in B$ W contains at most one voter who prefers p to b_i. Since $\|B\| = 3k$ and $k \geq 3$, this implies that W contains exactly a collection of voters corresponding to some exact cover of B by sets in $\mathcal{S}$. This completes the proof for the destructive case. □

5.3 Bribery in Maximin

We now move on to bribery in maximin. Given the previous results, it is not surprising that maximin is resistant both to constructive bribery and to destructive bribery. Our proof is an application of the "UV technique" of Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a). Very informally, the idea is to build an election in a way that ensures that the briber is limited to bribing only those voters who rank two special candidates ahead of the preferred one.

Theorem 5.7. *Maximin is resistant to constructive and destructive BV control.*

Proof. Our proofs follow via reductions from X3C. The reduction for the constructive case is almost identical the one for the constructive case and thus we will consider both cases in parallel.

Our reductions work as follows. Let $(B, \mathcal{S})$ be an instance of X3C, where $B = \{b_1, \ldots, b_{3k}\}$ is a set of $3k$ distinct elements, and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a family of 3-element subsets of B. (Without loss of generality, we assume that $n > k > 1$. If this is not the case, it is trivial to verify if $(B, \mathcal{S})$ is a "yes" instance of X3C.) We construct a set of candidates $C = \{p, d, s\} \cup B$, where p is our preferred candidate (the goal in the constructive setting is to ensure p is a unique winner) and d is our despised candidate (the goal in the destructive setting is to prevent d from being a unique winner). We construct six collections of voters, $V^1, V^2, V^3, V^4, V^5, V^6$, as follows:

1. V^1 contains $2n$ voters, $v^1_1, \ldots, v^1_{2n}$. For each i, $1 \leq i \leq n$, voters v^1_i and v^1_{i+n} have the following preference orders:

$$\begin{aligned} v^1_i &: d > s > S_i > p > B - S_i \\ v^1_{n+i} &: \overleftarrow{B - S_i} > p > \overleftarrow{S_i} > d > s. \end{aligned}$$

2. V^2 contains $2k$ voters, $v^2_1, \ldots, v^2_{2k}$. For each i, $1 \leq i \leq k$, voters v^2_i and v^2_{i+k} have the following preference orders:

$$\begin{aligned} v^2_i &: s > d > p > B \\ v^2_{k+i} &: \overleftarrow{B} > d > p > s. \end{aligned}$$

3. V^3 contains $2k$ voters, $v^3_1, \ldots, v^3_{2k}$. For each i, $1 \leq i \leq k$, voters v^3_i and v^3_{i+k} have the following preference orders:

$$\begin{aligned} v^3_i &: d > s > p > B \\ v^3_{k+i} &: \overleftarrow{B} > s > p > d. \end{aligned}$$

4. V^4 contains $4k$ voters, $v^4_1, \ldots, v^4_{4k}$. For each i, $1 \leq i \leq 2k$, voters v^4_i and v^4_{i+2k} have the following preference orders:

$$\begin{aligned} v^4_i &: d > B > p > s \\ v^4_{2k+i} &: s > p > d > \overleftarrow{B}. \end{aligned}$$

5. V^5 contains 2 voters, v^5_1, v^5_2 with the following preference orders

$$\begin{aligned} v^5_1 &: s > B > p > d \\ v^5_2 &: d > \overleftarrow{B} > p > s. \end{aligned}$$

6. V^6 contains a single voter, v^6_1, with preference order $p > d > s > B$.

We form two elections, E_c and E_d, where $E_c = (C, V^1 \cup \cdots \cup V^6)$ and $E_d = (C, V^1 \cup \cdots \cup V^5)$; that is, E_c and E_d are identical except E_d does not contain the single voter from V^6. E_c contains $2n + 8k + 3$ voters and E_d contains $2n + 8k + 2$ voters. Values of N_{E_c} and N_{E_d} for each pair of candidates are given in Table 4.

(a) Values of $N_{E_c}(\cdot,\cdot)$.

	p	d	s	B
p	$-$	$n+3k+2$	$n+3k+2$	$n+4k+1$
d	$n+5k+1$	$-$	$2n+4k+2$	$n+6k+2$
s	$n+5k+1$	$4k+1$	$-$	$n+4k+2$
B	$n+4k+2$	$n+2k+1$	$n+4k+1$	$\leq n+4k+2$

(b) Values of $N_{E_d}(\cdot,\cdot)$.

	p	d	s	B
p	$-$	$n+3k+1$	$n+3k+1$	$n+4k$
d	$n+5k+1$	$-$	$2n+4k+1$	$n+6k+1$
s	$n+5k+1$	$4k+1$	$-$	$n+4k+1$
B	$n+4k+2$	$n+2k+1$	$n+4k+1$	$n+4k+1$

Table 4: Values of $N_{E_c}(\cdot,\cdot)$ and $N_{E_d}(\cdot,\cdot)$ for each pair of candidates. Let E be one of E_c, E_d. An entry in row $c' \in \{p,d,s\}$ and column $c'' \in \{p,d,s\}$, $c' \neq c''$, of the appropriate table above gives value $N_E(c',c'')$. For row B and for column B we adopt the following convention. For each $c \in \{p,d,s\}$ and for each $b_i \in B$, an entry in row B and column c is equal to $N_E(b_i,c)$. For each $c \in \{p,d,s\}$ and for each $b_i \in B$, an entry in row c and column B is equal to $N_E(c,b_i)$. For each two distinct $b_i, b_j \in B$, the entry in row B and column B is the upper bound on $N_E(b_i,b_j)$. (For E_d this entry is, in fact, exact.)

For the constructive case, we claim that it is possible to ensure that p is a unique winner of election E_c by bribing at most k voters if and only if $(B,\mathcal{S})$ is a "yes" instance of X3C. Let us now prove this claim. By inspecting Table 4, and recalling that $n > k > 1$, we see that $\text{score}_{E_c}(p) = n+3k+2$, $\text{score}_{E_c}(d) = n+5k+1$, $score_{E_c}(s) = 4k+1$, and for each $b_i \in B$, $\text{score}_{E_c}(b_i) \leq n+2k+1$. That is, prior to any bribing, d is the unique winner and p has the second highest score.

It is straightforward to see that by bribing $t \leq k$ voters, the briber can change each candidate's score by at most t points. Thus, for the bribery to be successful, the briber has to bribe exactly k voters in such a way that d's score decreases to $n+4k+1$ and p's score increases to $n+4k+2$. To achieve this, the briber has to find a collection V' of voters such that $\|V'\| = k$, and

1. each voter in V' ranks p below both d and s, and
2. for each $b_i \in B$, there is a voter in V' who ranks p below b_i.

The only voters that satisfy the first condition are $v_1^1, \ldots, v_n^1, v_1^2, \ldots, v_k^2, v_1^3, \ldots, v_k^3$. Further, among these voters only $v_1^1, \ldots, v_n^1$ rank p below some member of B and, in fact, for each i, $1 \leq i \leq n$, v_i^1 ranks p below exactly three members of B. Thus it is straightforward to see that each k voters from $v_1^1, \ldots, v_n^1, v_1^2, \ldots, v_k^2, v_1^3, \ldots, v_k^3$ that satisfy the second condition correspond naturally to a cover of B by sets from $\mathcal{S}$. (Note that it suffices that the briber bribes voters in V' to rank p first without changing the votes in any other way, and that changing the votes in any other way than ranking p first is not necessary.) As a result, if it is possible to ensure that p is a winner of E_c by bribing at most k voters then $(B, \mathcal{S})$ is a "yes" instance of X3C. For the other direction, it is straightforward to verify that if $(B, \mathcal{S})$ is a "yes" instance of X3C then bribing k voters from $v_1^1, \ldots, v_n^1$ that correspond to a cover of B to rank p first suffices to ensure that p is a unique winner. This completes the proof for the constructive case.

For the destructive case, we claim that it is possible to ensure that d is not a unique winner of E_d if and only if $(B, \mathcal{S})$ is a "yes" instance of X3C. The proof is analogous to the constructive case: It suffices to note that p is the only candidate that can possibly tie for victory with d. The rest of the proof proceeds as for the constructive case. □

6. Fixed-Parameter Tractability

In this section we consider the parameterized complexity of multipronged control, in particular, the case where we can assume that the number of candidates is a small constant. Elections with few candidates are very natural. For example, in many countries presidential elections involve only a handful of candidates.

The main result of this section is that for many natural election systems $\mathcal{E}$ (formally, for all election systems whose winner determination problem can be expressed via an integer linear program of a certain form), it holds that the $\mathcal{E}$-AC+DC+AV+DV+BV control problem is fixed-parameter tractable (is in the complexity class FPT) for the parameter "number of candidates," both in the constructive setting and in the destructive setting. This result combines and significantly enhances FPT results from the literature, in particular, from the papers of Faliszewski, Hemaspaandra, and Hemaspaandra (2009a) and Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a), which are the model for and inspiration of this section. We also make explicit an "automatic" path to such results that is implicit in the work of the two papers cited in the previous sentence. This path should be helpful in letting many future analyses be done as tool-application exercises, rather than being case-by-case challenges.

The present paper is not the only paper to pick up on the type of pattern in the earlier work just mentioned, and to present a definition and results building on that. Independently of the present paper, Dorn and Schlotter (2010) have done this in a different, bribery-related context.

In this section we focus exclusively on the number of candidates as our parameter. That is, our parameter is the number of candidates initially in the election plus the number of candidates (if any) in the set of potential additional candidates. That is, in terms of the variables we have been using to describe multiprong control the parameter is $\|C\| + \|A\|$.

We mention that researchers sometimes analyze other parameterizations. For example, Liu et al. (2009), Liu and Zhu (2010), and Betzler and Uhlmann (2009) consider as the parameter the amount of change that one is allowed to use (e.g., the number of candidates one can add), Bartholdi, Tovey, and Trick (1989b), Betzler and Uhlmann (2009), and Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a) study as the parameter the number of voters (and also sometimes the number of candidates). And other parameters are sometimes used when considering the so-called possible winner problem (see, e.g., Betzler & Dorn, 2009; Betzler, Hemmann, & Niedermeier, 2009). However, we view the parameter "number of candidates" as the most essential and the most natural one. We now proceed with our discussion of fixed-parameter tractability, with the number of candidates as the parameter.

Let us consider an election system $\mathcal{E}$ and a set $C = \{c_1, \ldots, c_m\}$ of candidates. There are exactly $m!$ preference orders over the candidates in C and we will refer to them as $o_1, \ldots, o_{m!}$. Let us assume that $\mathcal{E}$ is anonymous (i.e., the winners of each $\mathcal{E}$ election do not depend on the order of votes or the names of the voters, but only—for each preference order o_i—on the number of votes with that preference order). We define predicate $win_{\mathcal{E}}(c_j, n_1, \ldots, n_{m!})$ to be true if and only if c_i is a unique winner of $\mathcal{E}$ elections with $C = \{c_1, \ldots, c_m\}$, where for each i, $1 \leq i \leq m!$, there are exactly n_i voters with preference order o_i. For the rest of this section, our inequalities always use one of the four operators "$>$," "$\geq$," "$<$," and "$\leq$."[16]

Definition 6.1. *We say that an anonymous election system $\mathcal{E}$ is unique-winner (nonunique-winner) integer-linear-program implementable if for each set of candidates $C = \{c_1, \ldots, c_m\}$ and each candidate $c_j \in C$ there exists a set S of linear inequalities with variables $n_1, \ldots, n_{m!}$ such that:*

1. *If the integer assignment $n_1 = \hat{n}_1, \ldots, n_{m!} = \hat{n}_{m!}$ satisfies S, then each $\hat{n}_i$ belongs to $\mathbb{N}$,*[17]
2. *S can be computed (i.e., obtained) in time polynomial in $m!$,*[18] *and*
3. *for each $(\hat{n}_1, \ldots, \hat{n}_{m!}) \in \mathbb{N}^{m!}$, we have that (a) holds if and only if (b) holds, where (a) and (b) are as follows:*
   (a) *S is satisfied by the assignment $n_1 = \hat{n}_1, \ldots, n_{m!} = \hat{n}_{m!}$.*
   (b) *c_j is a unique winner (is a winner) of an $\mathcal{E}$ election in which for each i, $1 \leq i \leq m!$, there are exactly $\hat{n}_i$ voters with preference order o_i, where o_i is the i'th preference order over the set C.*

16. We allow both strict and nonstrict inequalities. Since we allow only integer solutions, it is straightforward to simulate strict inequalities with nonstrict ones and to simulate nonstrict inequalities with strict ones, in both cases simply by adding a "1" to the appropriate side of the inequality. So we could equally well have allowed just strict, or just nonstrict, inequalities.
17. It is straightforward to to put $m!$ inequalities into S enforcing this condition. And this condition will help us make the electoral part of our definition meaningful, i.e., it will avoid having problems from the restriction in the final part of this definition that lets us avoid discussing negative numbers of voters.
18. We mention in passing that if the $m!$ in this part of the definition were changed to any other computable function of m, e.g., $m^{m^{m^m}}$, we would still obtain FPT results, and still would have them hold even in the strengthened version of FPT in which the f of "$f(\text{parameter}) \cdot \text{Inputsize}^{O(1)}$" is required to be computable. However, due to $m!$ being the number of preference orders over m candidates, having S be obtainable in time polynomial in $m!$ will in practice be a particularly common case.

    We also mention in passing that the FPT-establishing framework in this section and the results it yields, similarly to the case in our work mentioned earlier (Faliszewski, Hemaspaandra, & Hemaspaandra, 2009a; Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009a), not only will apply in the model where votes are input as a list of ballots, one per person, but also will hold in the so-called "succinct" model (see Faliszewski, Hemaspaandra, & Hemaspaandra, 2009a; Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009a), in which we are given the votes not as individual ballots but as binary numbers providing the number of voters having each preference order (or having each occurring preference order).

In a slight abuse of notation, for integer-linear-program implementable election systems $\mathcal{E}$ we will simply refer to the set S of linear inequalities from Definition 6.1 as $win_{\mathcal{E}}(c_j, n_1, \ldots, n_{m!})$. The particular set of candidates will always be clear from context. Naturally, it is straightforward to adapt Definition 6.1 to apply to approval voting, but for the sake of brevity we will not do so.

We are not aware of any natural systems that are integer-linear-program unique-winner implementable yet not integer-linear-program nonunique-winner implementable, or vice versa. In this paper we focus on the unique winner model so the reader may wonder why we defined the nonunique winner variant of integer-linear-program implementability. The answer is that, as we will see later in this section, it is a useful notion when dealing with destructive control.

The class of election systems that are integer-linear-program implementable is remarkably broad. For example, it is variously implicit in or a consequence of results of Faliszewski, Hemaspaandra, and Hemaspaandra (2009a) that plurality, veto, Borda, Dodgson, and each polynomial-time computable (in the number of candidates) family of scoring protocols are integer-linear-program implementable.[19] For many other election systems—e.g., Kemeny voting (Kemeny, 1959; Young & Levenglick, 1978) and Copeland voting—it is not clear whether they are integer-linear-program implementable, but there are similar approaches that will be as useful for us. We will return to this issue at the end of this section.

Theorem 6.2. *Let $\mathcal{E}$ be an integer-linear-program unique-winner implementable election system. For number of candidates as the parameter, constructive $\mathcal{E}$-AC+DC+AV+DV+BV is in FPT.*

Proof. Let $(C, A, V, W, p, k_{\mathrm{AC}}, k_{\mathrm{DC}}, k_{\mathrm{AV}}, k_{\mathrm{DV}}, k_{\mathrm{BV}})$ be our input instance of the constructive $\mathcal{E}$-AC+DC+AV+DV+BV control problem, as described in Definition 4.1. Let $C = \{p, c_1, \ldots, c_{m'}\}$ and $A = \{a_1, \ldots, a_{m''}\}$. Our parameter, the total number of candidates, is $m = m' + m'' + 1$. For each subset K of $C \cup A$ we let $o_1^K, \ldots, o_{\|K\|!}^K$ mean the $\|K\|!$ preference orders over K.

The idea of our algorithm is to perform an exhaustive search through all the subsets of candidates K, $K \subseteq C \cup A$, and for each K check whether (a) it is possible to obtain K from C by deleting at most k_{DC} candidates and adding at most k_{AC} candidates from A, and (b) it is possible to ensure that p is a unique winner of election (K, V) by deleting at most k_{DV} voters, adding at most k_{AV} voters from W, and bribing at most k_{BV} voters. Given K, step (a) can straightforwardly be implemented in polynomial time. To implement step (b), we introduce a linear integer program $P(K)$, which is satisfiable if and only if step (b) holds. Let us now fix $K \subseteq C \cup A$ and describe the integer linear program $P(K)$.

We assume that $p \in K$ as it is not legal to delete p (and it would be pointless, given that we want to ensure his or her victory). We interpret preference orders of voters in V and W as limited to the candidate set K. We use the following *constants* in our program.

19. Let m be the number of candidates. A scoring protocol is a vector of m nonnegative integers satisfying $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. Each candidate receives α_i points for each vote that ranks him or her in the i'th position, and the candidate(s) with most points win. Many election systems can be viewed as families of scoring protocols. For example, plurality is defined by scoring protocols of the form $(1, 0, \ldots, 0)$, veto is defined by scoring protocols of the form $(1, \ldots, 1, 0)$, and Borda is defined by scoring protocols of the form $(m-1, m-2, \ldots, 0)$, where m is the number of candidates.

For each i, $1 \leq i \leq \|K\|!$, we let n_i^V be the number of voters in V with preference order o_i^K, and we let n_i^W be the number of voters in W with preference order o_i^K. $P(K)$ contains the following variables (described together with their intended interpretation):

Variables $av_1, \dots, av_{\|K\|!}$. For each i, $1 \leq i \leq \|K\|!$, we interpret av_i as the number of voters with preference o_i^K that we add from W.

Variables $dv_1, \dots, dv_{\|K\|!}$. For each i, $1 \leq i \leq \|K\|!$, we interpret dv_i as the number of voters with preference o_i^K that we delete from V.

Variables $bv_{1,1}, bv_{1,2}, \dots, bv_{1,\|K\|!}, bv_{2,1}, \dots, bv_{\|K\|!,\|K\|!}$. For each i, j, $1 \leq i, j \leq \|K\|!$, we interpret $bv_{i,j}$ as the number of voters with preference o_i^K that, in case $i \neq j$, we bribe to switch to preference order o_j^K, or, in case $i = j$, we leave unbribed.

$P(K)$ contains the following constraints.

1. All the variables have nonnegative values.

2. For each variable av_i, $1 \leq i \leq \|K\|!$, there are enough voters in W with preference order o_i^K to be added. That is, for each i, $1 \leq i \leq \|K\|!$, we have a constraint $av_i \leq n_i^W$. Altogether, we can add at most k_{AV} voters so we have a constraint $\sum_{i=1}^{\|K\|!} av_i \leq k_{\mathrm{AV}}$.

3. For each variable dv_i, $1 \leq i \leq \|K\|!$, there are enough voters in V with preference order o_i^K to be deleted. That is, for each i, $1 \leq i \leq \|K\|!$, we have a constraint $dv_i \leq n_i^V$. Altogether, we can delete at most k_{DV} voters so we have a constraint $\sum_{i=1}^{\|K\|!} dv_i \leq k_{\mathrm{DV}}$.

4. For each variable $bv_{i,j}$, $1 \leq i, j \leq \|K\|!$, there are enough voters with preference o_i^K to be bribed. That is, for each i, $1 \leq i \leq \|K\|!$, we have a constraint $\sum_{j=1}^{\|K\|!} bv_{i,j} = n_i^V + av_i - dv_i$ (the equality comes from the fact that for each i, $1 \leq i \leq \|K\|!$, $bv_{i,i}$ is the number of voters with preference o_i^K that we do not bribe). Altogether, we can bribe at most k_{BV} voters so we also have a constraint
$$\left(\sum_{i=1}^{\|K\|!} \sum_{j=1}^{\|K\|!} bv_{i,j}\right) - \sum_{i=1}^{\|K\|!} bv_{i,i} \leq k_{\mathrm{BV}}.$$

5. Candidate p is the unique winner of the election after we have executed all the adding, deleting, and bribing of voters. Using the fact that $\mathcal{E}$ is integer-linear-program unique-winner implementable, we can express this as $win_{\mathcal{E}}(p, \ell_1, \dots, \ell_{\|K\|!})$, where we substitute each ℓ_j, $1 \leq j \leq \|K\|!$, by $\sum_{i=1}^{\|K\|!} bv_{i,j}$ (note that, by previous constraints, variables describing bribery already take into account adding and deleting voters). This is a legal integer-linear-program constraint as $win_{\mathcal{E}}(p, \ell_1, \dots, \ell_{\|K\|!})$ is simply a conjunction of linear inequalities over $\ell_1, \dots, \ell_{\|K\|!}$.

The number of variables and the number of inequalities in $P(K)$ are each polynomially bounded in $m!$. Keeping in mind Definitions 4.1 and 6.1, it is straightforward to see that program $P(K)$ does exactly what we expect it to. And testing whether $P(K)$ is satisfiable (i.e., has an integer solution, as we are in the framework of an integer linear program) is in FPT, with respect to the number of candidates being our parametrization, by using Lenstra's (1983) algorithm. Thus our complete FPT algorithm for the $\mathcal{E}$-AC+DC+AV+DV+BV problem works as follows. For each subset K of $C \cup A$ that includes p we execute the following two steps:

1. Check whether it is possible to obtain K from C by deleting at most k_{DC} candidates and by adding at most k_{AC} candidates from A.

2. Form linear program $P(K)$ and check whether it has any integral solutions using the algorithm of Lenstra (1983). Accept if so.

If after trying all sets K we have not accepted, then reject.

From the previous discussion, this algorithm is correct. Also, since (a) there are exactly 2^{m-1} sets K to try, (b) executing the first step above can be done in time polynomial in m, and (c) the second step is in FPT (given that m is the parameter), constructive $\mathcal{E}$-AC+DC+AV+DV+BV is in FPT for parameter m. $\square$

Theorem 6.2 deals with constructive control only. However, using its proof, it is straightforward to prove a destructive variant of the result. We say that an election system is strongly voiced (Hemaspaandra et al., 2007) if it holds that whenever there is at least one candidate, there is at least one winner.[20]

Corollary 6.3. *Let $\mathcal{E}$ be a strongly voiced, integer-linear-program nonunique-winner implementable election system. Destructive $\mathcal{E}$-AC+DC+AV+DV+BV is in FPT for the parameter number of candidates.*

To see that the corollary holds, it is enough to note that for strongly voiced election systems a candidate can be prevented from being a unique winner if and only if some other candidate can be made a (possibly nonunique) winner (see, e.g., Footnote 5 of Hemaspaandra et al., 2007, for a relevant discussion). Thus to prove Corollary 6.3, we can simply use an algorithm that for each candidate other than the despised one sees whether that candidate can be made a (perhaps nonunique) winner, and if any can be made a (perhaps nonunique) winner, declares destructive control achievable. (And the precise integer linear programming feasibility problem solution given by Lenstra's algorithm will reveal what action achieves the control.) This can be done in FPT using the algorithm from the proof of Theorem 6.2, adapted to work for the nonunique-winner problem (this is trivial given that Corollary 6.3 assumes that $\mathcal{E}$ is integer-linear-program nonunique-winner implementable).

Let us now go back to the issue that some election systems may not be integer-linear-program implementable. As an example, let us consider maximin. Let $E = (C, V)$ be an election, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$. As before, by $o_1, \ldots, o_{m!}$ we mean the $m!$ possible preference orders over C, and for each i, $1 \leq i \leq m!$, by n_i we

20. Please recall Footnote 5. In this paper's model of elections, which is the one that matches that of the previous papers studying control, the notion of election does allow an election system to have on some inputs no winner. However, we mention that in the social choice world, formalizations of elections typically build into their definition an exclusion of the possibility of having no winners, and so in that world, "strongly voiced" would seem a strange concept to explicitly define and require, as it is built into the general definition from the start.

mean the number of voters in V that report preference order o_i. For each c_i and c_j in C, $c_i \neq c_j$, we let $O(c_i, c_j)$ be the set of preference orders over C where c_i is preferred to c_j. Let $k = (k_1, \ldots, k_m)$ be a vector of nonnegative integers such that for each i, $1 \leq i \leq m$, it holds that $1 \leq k_i \leq m$. For such a vector k and a candidate $c_\ell \in C$ we define $M(c_\ell, k_1, \ldots, k_m)$ to be the following set of linear integer inequalities:

1. For each candidate c_i, his or her maximin score is equal to $N_E(c_i, c_{k_i})$. That is, for each i, j, $1 \leq i, j \leq m$, $i \neq j$, we have constraint $\sum_{o_k \in O(c_i, c_{k_i})} n_k \leq \sum_{o_k \in O(c_i, c_j)} n_k$
2. c_ℓ has the highest maximin score in election E and thus is the unique winner of E. That is, for each i, $1 \leq i \leq m$, $i \neq \ell$, we have constraint $\sum_{o_k \in O(c_\ell, c_{k_\ell})} n_k > \sum_{o_k \in O(c_i, c_{k_i})} n_k$.

It is straightforward to see that c_ℓ is a unique maximin winner of E if and only if there is a vector $k = (k_1, \ldots, k_m)$ such that all inequalities of $M(c_\ell, k_1, \ldots, k_m)$ are satisfied. It is also straightforward how to modify the above construction to handle the nonunique winner case. Since there are only $O(m^m)$ vectors k to try and each $M(c_\ell, k_1, \ldots, k_m)$ contains $O(m^2)$ inequalities, it is straightforward to modify the proof of Theorem 6.2 to work for maximin: Assuming that one is interested in ensuring candidate c_ℓ's victory, one simply has to replace program $P(K)$ in the proof of Theorem 6.2 with a family of programs that each include a different $M(c_\ell, k_1, \ldots, k_m)$ for testing if c_ℓ had won. And one would accept if any of these were satisfiable. Thus we have the following result.

Corollary 6.4. *Constructive AC+DC+AV+DV+BV control and destructive AC+DC+AV+DV+BV control are both in FPT for maximin for the parameter number of candidates.*

The above construction for the winner problem in maximin can be viewed as, in effect, a disjunction of a set of integer linear programs. Such constructions for the winner problem have already been obtained for Kemeny elections[21] by Faliszewski, Hemaspaandra, and Hemaspaandra (2009a) and for Copeland elections by Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a). Thus we have the following theorem.

21. Our definition is one that is for the notion of a Kemeny voting system where voting is by preference orders (so no ties within a voter's preferences are allowed) and the allowed values for a Kemeny consensus (soon to be defined) are also limited to preference orders. This is not the notion from the original papers (Kemeny, 1959; Young & Levenglick, 1978), but rather is the notion of Kemeny elections used by Faliszewski, Hemaspaandra, and Hemaspaandra (2009a), who themselves mention they are following the Saari and Merlin (2000) notion of Kemeny elections; interestingly, Hemaspaandra, Spakowski, and Vogel (2005) ensure that their own main result, which is about the complexity of the winner problem for Kemeny elections, holds for both cases, but they have to use real care to make that hold. So, all that being said, Kemeny elections are (as used in this paper) defined as follows. Let $E = (C, V)$ be an election, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$. For each preference order r and each voter v_i, $1 \leq i \leq n$, let $d(r, v_i)$ be the number of inversions between r and the preference order of v_i (i.e., the number of pairs of candidates $\{c_i, c_j\}$ ranked by v_i and r in the opposite order). The Kemeny score of a preference order r is defined as $\sum_{i=1}^{n} d(r, v_i)$. A preference order r is said to be a Kemeny consensus if its Kemeny score is lowest (tied for lowest is fine—the lowest does not have to be unique) among all preference orders over C. A candidate c_i is a Kemeny winner if c_i is ranked first in some Kemeny consensus.

Corollary 6.5. *With number of candidates as the parameter, constructive AC+DC+AV+DV+BV control and destructive AC+DC+AV+DV+BV control are in FPT for Kemeny and, for each rational α, $0 \leq \alpha \leq 1$, for Copeland$^{\alpha}$.*

We conclude with an important caveat. The FPT algorithms of this section are very broad in their coverage, but in practice they would be difficult to use as their running time depends on (the fixed-value parameter) m in a very fast-growing way and as Lenstra's algorithm has a large multiplicative constant in its polynomial running time. Thus the results of this section should best be interpreted as indicating that, for multipronged control in our setting, it is impossible to prove non-FPT-ness (and so it is impossible to prove fixed-parameter *hardness* in terms of the levels of the so-called "W" hierarchy of fixed-parameter complexity, unless that hierarchy collapses to FPT). If one is interested in truly practically implementing a multipronged control attack, one should probably devise a problem-specific algorithm rather than using our very generally applicable FPT construction.

7. Conclusions

This paper was motivated by the desire to move the study of control a step in the direction of better capturing real-life scenarios. In particular, attackers will not tend to artificially limit themselves to one type of attack, but rather may well pull out and employ every trick in their playbook. And so we have studied control attacks (including even (unpriced) bribery, for the first time, within the framework of control) in which multiple attack prongs can be pursued by the attacker.

This paper has shown that that approach—combining various types of control into multi-prong control attacks—is useful. For example, it allows us to express control vulnerability results and proofs in a compact way, and to obtain vulnerability results that are stronger than would be obtained for single prongs alone (and that immediately imply prior results, which were just for single prongs). The central contribution of this paper is that it provides a broad set of tools to allow one, from the immunity/vulnerability/resistance of the basic control prongs, to almost always determine the complexity of combinations of those prongs. For systems where each basic prong is immune, vulnerable, or resistant, the only blind spot in our machinery has to do with cases where there are multiple vulnerable prongs, as there we showed that the underlying prongs simply do not on their own determine the complexity of their multiprong combination. Even for that case, for all the systems discussed in the paper, and indeed for a very broad class of systems ("vulnerability-combining" systems) that we expect will include nearly all natural systems, we show how to classify even in the face of this difficulty. We provide a useful tool to help researchers prove that additional systems are vulnerability-combining. Section 4.4 summarizes all that. And Table 5 summarizes our results regarding the five election systems we have focused on in this paper; see also Tables 2 and 3.

However, we have also seen that there exists a natural election system, OriginalLlull, that unless P = NP is not constructive vulnerability-combining. That system is vulnerable to both constructive AC control and constructive AV control yet is resistant to constructive AC+AV control.

Control type	plurality		Condorcet		Copeland$^\alpha$, for each $0 \leq \alpha \leq 1$		approval		maximin	
	Con.	Des.	Con.	Des.	Con.	Des.	Con.	Des.	Con.	Des.
AC	R	R	I	V	R	V	I	V	***R***	***V***
DC	R	R	V	I	R	V	V	I	***V***	***V***
AV	V	V	R	V	R	R	R	V	***R***	***R***
DV	V	V	R	V	R	R	R	V	***R***	***R***
BV	V	***V***	R	***V***	R	R	R	***V***	***R***	***R***

Table 5: Resistance to basic control types for the five main election systems studied in this paper. In the table, I means the system is immune to the given control type, R means resistance, and V means vulnerability. Constructive results for AC, DC, AV, and DV for plurality and Condorcet are due to Bartholdi et al. (1992) and the corresponding destructive results are due to Hemaspaandra et al. (2007). All results for AC, DC, AV, and DV for approval are due to Hemaspaandra et al. (2007). All results regarding Copeland are due to Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a). Constructive bribery results for plurality and approval are due to Faliszewski, Hemaspaandra, and Hemaspaandra (2009a), and the constructive bribery result for Condorcet is implicit in the work of Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe (2009a). All the remaining entries (i.e., all results regarding maximin, and destructive bribery results for plurality, approval, and Condorcet) are due to this paper (and are bold-italic in the table). The main contribution of this paper, however, is the analysis of multiprong control types. In particular, all constructive (or destructive) collections of prongs for the systems here combine as specified by Classification Rule A of Section 4.2. This holds due to Theorem 4.7, the paragraphs immediately following it, and the fact that we prove all five systems in this table to be both constructive vulnerability-combining and destructive vulnerability-combining (Theorems 4.11 and 5.1).

We have also shown that as far as fixed-parameter tractability goes, at least with respect to the parameter "number of candidates," a very broad class of election systems is vulnerable to the most full attack over basic prongs, namely, a AC+DC+AV+DV+BV control attack.

Finally, in the appendix, we prove that no candidate whose Dodgson score is more than $\|C\|^2$ times the Dodgson winner's score can be a maximin winner.

This paper studies multipronged control where the prongs may include various standard types of control or bribery. However, it is straightforward to see that our framework can be naturally extended to include manipulation, and we commend that direction to any interested reader (and mention in passing that Section 4 of Faliszewski, Hemaspaandra, & Hemaspaandra, 2009a, does find a connection between bribery and manipulation). To do so, one would have to allow some of the voters—the manipulators—to have blank preference orders and, if such voters were to be included in the election, the controlling agent would have to decide on how to fill them in. (That is, the controlling agent in this model would control both the control aspects and the manipulation aspects.) It is interesting that in this model the controlling agent might be able to add manipulative voters (if there were manipulators among the voters that can be added) or even choose to delete them (it may seem that deleting manipulators is never useful but Zuckerman, Procaccia, & Rosenschein, 2009, give an example where deleting a manipulator is necessary to make one's favorite candidate a winner of a Copeland election).

We mention as a natural but involved open direction the study of multipronged control in the setting where there are multiple controlling agents, each with a different goal, each controlling a different prong. In such a setting, it is interesting to consider game-theoretic scenarios as well as situations in which, for example, one of the controlling agents is seeking an action that will succeed regardless of the action of the other attacker. Another important direction is the following. Section 6 for the very specific (namely, fixed-parameter) case it studies sets up a quite flexible framework for classifying a broad range of control-attack problems as being in polynomial time. It would be interesting to see what highly flexible schemes and matching results one can find to broadly classify the control complexity of elections in the general case (for motivation, see for example the work broadly classifying the manipulation complexity of scoring protocols, Hemaspaandra & Hemaspaandra, 2007; Conitzer, Sandholm, & Lang, 2007; Procaccia & Rosenschein, 2007).

Acknowledgments

Supported in part by NSF grants CCF-0426761, IIS-0713061, and CCF-0915792, Polish Ministry of Science and Higher Education grant N-N206-378637, the Foundation for Polish Science's Homing/Powroty program, AGH University of Science and Technology grant 11.11.120.865, the ESF's EUROCORES program LogICCC, and Friedrich Wilhelm Bessel Research Awards to Edith Hemaspaandra and Lane A. Hemaspaandra. A preliminary version (Faliszewski, Hemaspaandra, & Hemaspaandra, 2009b) of this paper appeared in the proceedings of the 21st International Joint Conference on Artificial Intelligence, July 2009. For their very helpful comments and suggestions, we are deeply indebted to JAIR handling editor Vincent Conitzer, Edith Elkind, and the anonymous IJCAI and JAIR referees.

Appendix A. A Connection of Maximin Voting to Dodgson Voting

Section 5 focused on control in maximin voting. In this appendix, we focus on a different facet of maximin voting, namely, we show a connection between maximin and the famous voting rule (i.e., election system) of Dodgson.

Dodgson voting, proposed in the 19th century by Charles Lutwidge Dodgson (1876),[22] works as follows. Let $E = (C, V)$ be an election, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$. For a candidate $c_i \in C$, the Dodgson score of c_i, denoted $\text{score}_E^D(c_i)$, is the smallest number of sequential swaps of adjacent candidates on the preference lists of voters in V needed to make c_i become the Condorcet winner. The candidates with the lowest score are the Dodgson election's winners. That is, Dodgson defined his system to elect those candidates that are closest to being Condorcet winners in the sense of adjacent-swaps distance. Although Dodgson's eighteenth-century election system was directly defined in terms of distance, there remains ongoing interest in understanding the classes of voting rules that can be captured in various distance-based frameworks (see, e.g., Meskanen & Nurmi, 2008; Elkind, Faliszewski, & Slinko, 2009, 2010c, 2010b).

22. Dodgson is better known as Lewis Carroll, the renowned author of "Alice's Adventures in Wonderland."

Unfortunately, it is known that deciding whether a given candidate is a winner according to Dodgson's rule is quite complex. In fact, Hemaspaandra, Hemaspaandra, and Rothe (1997), strengthening an NP-hardness result of Bartholdi et al. (1989b), showed that this problem is complete for parallelized access to NP. That is, it is complete for the Θ_2^{p} level of the polynomial hierarchy. Nonetheless, many researchers have sought efficient ways of computing Dodgson winners, for example by using frequently correct heuristics (Homan & Hemaspaandra, 2009; McCabe-Dansted, Pritchard, & Slinko, 2008), fixed-parameter tractability (see Bartholdi et al., 1989b; Faliszewski, Hemaspaandra, & Hemaspaandra, 2009a; Betzler, Guo, & Niedermeier, 2010, and the discussion in Footnote 17 of Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009a), and approximation algorithms for Dodgson scores (Caragiannis, Covey, Feldman, Homan, Kaklamanis, Karanikolas, Procaccia, & Rosenschein, 2009).

In addition to its high computational cost in determining winners, Dodgson's rule is often criticized for not having basic properties one would expect a good voting rule to have. For example, Dodgson's rule is not "weak-Condorcet consistent" (Brandt et al., 2010)—equivalently, it does not satisfy Fishburn's "strict Condorcet principle"—and does not satisfy homogeneity and monotonicity (see Brandt, 2009, which surveys a number of defects of Dodgson's rule). We provide definitions for the latter two notions (in the former case just for the case we need here, namely, anonymous rules), as they will be relevant to this section.

Homogeneity. We say that an anonymous voting rule $\mathcal{R}$ is homogeneous if for each positive integer k and each election $E = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$, it holds that $\mathcal{R}$ has the same winner set on E as on $E' = (C, V')$, where $V' = (\underbrace{v_1, \ldots, v_1}_{k}, \underbrace{v_2, \ldots, v_2}_{k}, \ldots, \underbrace{v_n, \ldots, v_n}_{k})$.

Monotonicity. We say that a voting rule $\mathcal{R}$ is monotone if for each election $E = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$, it holds that if some candidate $c_i \in C$ is a winner of E then c_i is also a winner of an election E' that is identical to E except that some voters rank c_i higher (without changing the relative order of all the remaining candidates).

Continuing the Caragiannis et al. (2009) line of research on approximately computing Dodgson scores, Caragiannis, Kaklamanis, Karanikolas, and Procaccia (2010) devised an approximation algorithm for computing Dodgson scores that, given an election $E = (C, V)$, where $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$ and a candidate c_i in C, computes in polynomial time a nonnegative integer $\mathrm{sc}_E(c_i)$ such that $\mathrm{score}_E^D(c_i) \leq \mathrm{sc}_E(c_i)$ and $\mathrm{sc}_E(c_i) = O(m \log m) \cdot \mathrm{score}_E^D(c_i)$. That is, the algorithm given by Caragiannis et al. (2010) is, in a natural sense, an $O(m \log m)$-approximation of the Dodgson score.[23] This algorithm

23. Throughout this section, we use the notion "$f(m)$-approximation of g" in the sense it is typically used when dealing with minimization problems. That is, we mean that the approximation outputs a value that is at least g and at most $f(m) \cdot g$. We are slightly informal (i.e., sloppy) regarding the interplay between this notation and Big-Oh notation; however, the sloppiness is of a quite standard type that will not cause confusion. We assume that the type of the input to g and type of the input to the approximation is clear from context; in this paper, in both cases, the arguments are an election E and a candidate c_i.

has additional properties: If one defines a voting rule to elect those candidates that have lowest scores according to the algorithm, then that voting rule is Condorcet consistent (i.e., when a Condorcet winner exists, he or she is the one and only winner under the voting rule), homogeneous, and monotone.

The mentioned result of Caragiannis et al. (2010) is very interesting. In our next theorem we show that maximin—which like the Caragiannis et al. rule is Condorcet-consistent, homogeneous, and monotone—also elects candidates that are, in a certain different yet precise sense, "close" to being Dodgson winners. This is interesting as maximin is often considered to be quite different from Dodgson's rule. Our proof is inspired by that of Caragiannis et al. (2010).

Theorem A.1. *Let $E = (C, V)$ be an election and let $W \subseteq C$ be a set of candidates that win in E according to the maximin rule. Let $m = \|C\|$ and let $s = \min_{c_i \in C} \mathrm{score}_E^D(c_i)$. For each $c_i \in W$ it holds that $s \leq \mathrm{score}_E^D(c_i) \leq m^2 s$.*

Proof. Let us fix an election $E = (C, V)$ with $C = \{c_1, \ldots c_m\}$ and $V = (v_1, \ldots, v_n)$. For each two candidates $c_i, c_j \in C$ we define $\mathrm{df}_E(c_i, c_j)$ to be the smallest number k such that if k voters in V changed their preference order to rank c_i ahead of c_j, then c_i would be preferred to c_j by more than half of the voters. Note that if for some $c_i, c_j \in C$ we have $\mathrm{df}_E(c_i, c_j) > 0$ then

$$N_E(c_i, c_j) + \mathrm{df}_E(c_i, c_j) = \left\lfloor \frac{n}{2} \right\rfloor + 1.$$

For each candidate $c_i \in C$ we define $\mathrm{sc}'_E(c_i)$ to be

$$\mathrm{sc}'_E(c_i) = m^2 \max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\}.$$

We now prove that sc' is an m^2-approximation of the Dodgson score.

Lemma A.2. *For each $c_i \in C$ it holds that $\mathrm{score}_E^D(c_i) \leq \mathrm{sc}'_E(c_i) \leq m^2 \mathrm{score}_E^D(c_i)$.*

Proof. Let us fix some $c_i \in C$. To see that the second inequality in the lemma statement holds, note that $\max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\} \leq \sum_{c_j \in C - \{c_i\}} \mathrm{df}_E(c_i, c_j) \leq \mathrm{score}_E^D(c_i)$ because for each candidate c_k we, at least, have to perform $\mathrm{df}_E(c_i, c_k)$ swaps to ensure that c_i defeats c_k in their majority head-to-head contest. Thus, after multiplying by m^2, we have

$$m^2 \max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\} \leq m^2 \mathrm{score}_E^D(c_i).$$

Let us now consider the first inequality. Let c_k be some candidate in $C - \{c_i\}$. To make sure that c_i is ranked higher than c_k by more than half of the voters, we can shift c_i to the first position in the preference lists of $\max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\} \geq \mathrm{df}_E(c_i, c_k)$ voters (or, all the remaining voters if less than $\max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\}$ voters do not rank c_i as their top choice). This requires at most m adjacent swaps per voter. Since there are $m - 1$ candidates in $C - \{c_i\}$, $m^2 \max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\}$ adjacent swaps are certainly sufficient to make c_i a Condorcet winner. (Lemma A.2) □

It remains to show that if some candidate c_i is a maximin winner in E then $\mathrm{sc}'_E(c_i)$ is minimal. Fortunately, this is straightforward to see. If some candidate c_i is a Condorcet winner of E then he or she is the unique maximin winner and he or she is the unique candidate c_i with $\mathrm{sc}'_E(c_i) = 0$. Let us assume that there is no Condorcet winner of E. Let us fix some candidate $c_i \in C$ and let $c_k \in C - \{c_i\}$ be a candidate such that $\mathrm{sc}'_E(c_i) = m^2 \mathrm{df}_E(c_i, c_k)$. That is, $\mathrm{df}_E(c_i, c_k) = \max\{\mathrm{df}_E(c_i, c_j) \mid c_j \in C - \{c_i\}\}$ and $\mathrm{df}_E(c_i, c_k) > 0$. Due to this last fact and our choice of c_k, we have $\mathrm{df}_E(c_i, c_k) = \lfloor \frac{n}{2} \rfloor + 1 - N_E(c_i, c_k)$ and so

$$N_E(c_i, c_k) = \left\lfloor \frac{n}{2} \right\rfloor + 1 - \mathrm{df}_E(c_i, c_k) = \min_{c_j \in C - \{c_i\}} N_E(c_i, c_j) = \mathrm{score}_E(c_i),$$

where $\mathrm{score}_E(c_i)$ is the maximin score of c_i in E. Thus each candidate c_i with the lowest value $\mathrm{sc}'_E(c_i)$ also has the highest maximin score. □

Theorem A.1 says that every maximin winner's Dodgson score is no less than the Dodgson score of the Dodgson winner(s) (that fact of course holds trivially), and is no more than m^2 times the Dodgson score of the Dodgson winner(s). That is, we have proven that *no candidate whose Dodgson score is more than m^2 times that of the Dodgson winner(s) can be a maximin winner.*

References

Bartholdi, III, J., & Orlin, J. (1991). Single transferable vote resists strategic voting. *Social Choice and Welfare*, *8*(4), 341–354.

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

Bartholdi, III, J., Tovey, C., & Trick, M. (1989b). Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, *6*(2), 157–165.

Bartholdi, III, J., Tovey, C., & Trick, M. (1992). How hard is it to control an election? *Mathematical and Computer Modeling*, *16*(8/9), 27–40.

Baumeister, D., Erdélyi, G., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2010). Computational aspects of approval voting. In Laslier, J., & Sanver, M. (Eds.), *Handbook on Approval Voting*, pp. 199–251. Springer.

Betzler, N., & Dorn, B. (2009). Towards a dichotomy of finding possible winners in elections based on scoring rules. In *Proceedings of the 34th International Symposium on Mathematical Foundations of Computer Science*, pp. 124–136. Springer-Verlag *Lecture Notes in Computer Science #5734*.

Betzler, N., Guo, J., & Niedermeier, R. (2010). Parameterized computational complexity of Dodgson and Young elections. *Information and Computation*, *208*(2), 165–177.

Betzler, N., Hemmann, S., & Niedermeier, R. (2009). A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 53–58. AAAI Press.

Betzler, N., & Uhlmann, J. (2009). Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, *410*(52), 43–53.

Black, D. (1958). *The Theory of Committees and Elections*. Cambridge University Press.

Brandt, F. (2009). Some remarks on Dodgson's voting rule. *Mathematical Logic Quarterly*, *55*(4), 460–463.

Brandt, F., Brill, M., Hemaspaandra, E., & Hemaspaandra, L. (2010). Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 715–722. AAAI Press.

Caragiannis, I., Covey, J., Feldman, M., Homan, C., Kaklamanis, C., Karanikolas, N., Procaccia, A., & Rosenschein, J. (2009). On the approximability of Dodgson and Young elections. In *Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 1058–1067. Society for Industrial and Applied Mathematics.

Caragiannis, I., Kaklamanis, C., Karanikolas, N., & Procaccia, A. (2010). Socially desirable approximations for Dodgson's voting rule. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pp. 253–262. ACM Press.

Conitzer, V., & Sandholm, T. (2006). Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pp. 627–634. AAAI Press.

Conitzer, V., Sandholm, T., & Lang, J. (2007). When are elections with few candidates hard to manipulate? *Journal of the ACM*, *54*(3), Article 14.

Dobzinski, S., & Procaccia, A. (2008). Frequent manipulability of elections: The case of two voters. In *Proceedings of the 4th International Workshop On Internet And Network Economics*, pp. 653–664. Springer-Verlag *Lecture Notes in Computer Science #5385*.

Dodgson, C. (1876). 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 McLean & Urken, 1995, and Black, 1958, both of which reprint this paper).

Dorn, B., & Schlotter, I. (2010). Multivariate complexity analysis of swap bribery. In *Proceedings of the 5th International Symposium on Parameterized and Exact Computation*, pp. 107–122. Springer-Verlag *Lecture Notes in Computer Science #6478*.

Elkind, E., Faliszewski, P., & Slinko, A. (2009). On distance rationalizability of some voting rules. In *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge*, pp. 108–117. ACM Press.

Elkind, E., Faliszewski, P., & Slinko, A. (2010a). Cloning in elections. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 768–773. AAAI Press.

Elkind, E., Faliszewski, P., & Slinko, A. (2010b). Good rationalizations of voting rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 774–779. AAAI Press.

Elkind, E., Faliszewski, P., & Slinko, A. (2010c). On the role of distances in defining voting rules. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 375–382. International Foundation for Autonomous Agents and Multiagent Systems.

Erdélyi, G., Nowak, M., & Rothe, J. (2009). Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, *55*(4), 425–443.

Erdélyi, G., Piras, L., & Rothe, J. (2010a). Bucklin voting is broadly resistant to control. Tech. rep. arXiv:1005.4115 [cs.GT], arXiv.org.

Erdélyi, G., Piras, L., & Rothe, J. (2010b). Control complexity in fallback voting. Tech. rep. arXiv:1004.3398 [cs.GT], arXiv.org.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2009a). How hard is bribery in elections? *Journal of Artificial Intelligence Research*, *35*, 485–532.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2009b). Multimode attacks on elections. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 128–133. AAAI Press.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2010a). Multimode control attacks on elections. Tech. rep. arXiv:1007.1800 [cs.GT], Computing Research Repository, http://arXiv.org/corr/.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2010b). Using complexity to protect elections. *Communications of the ACM*, *53*(11), 74–82.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2007). Llull and Copeland voting broadly resist bribery and control. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pp. 724–730. AAAI Press.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009a). Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, *35*, 275–341.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009b). A richer understanding of the complexity of election systems. In Ravi, S., & Shukla, S. (Eds.), *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pp. 375–406. Springer.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2011). The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, *209*(2), 89–107.

Faliszewski, P., Hemaspaandra, E., & Schnoor, H. (2008). Copeland voting: Ties matter. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pp. 983–990. International Foundation for Autonomous Agents and Multiagent Systems.

Faliszewski, P., Hemaspaandra, E., & Schnoor, H. (2010). Manipulation of Copeland elections. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, pp. 367–374. International Foundation for Autonomous Agents and Multiagent Systems.

Friedgut, E., Kalai, G., & Nisan, N. (2008). Elections can be manipulated often. In *Proceedings of the 49th IEEE Symposium on Foundations of Computer Science*, pp. 243–249. IEEE Computer Society.

Garey, M., & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

Hägele, G., & Pukelsheim, F. (2001). The electoral writings of Ramon Llull. *Studia Lulliana*, *41*(97), 3–38.

Hemaspaandra, E., & Hemaspaandra, L. (2007). Dichotomy for voting systems. *Journal of Computer and System Sciences*, *73*(1), 73–83.

Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (1997). Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, *44*(6), 806–825.

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

Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009). Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, *55*(4), 397–424.

Hemaspaandra, E., Spakowski, H., & Vogel, J. (2005). The complexity of Kemeny elections. *Theoretical Computer Science*, *349*(3), 382–391.

Homan, C., & Hemaspaandra, L. (2009). Guarantees for the success frequency of an algorithm for finding Dodgson-election winners. *Journal of Heuristics*, *15*(4), 403–423.

Isaksson, M., Kindler, G., & Mossel, E. (2010). The geometry of manipulation—A quantitative proof of the Gibbard–Satterthwaite Theorem. In *Proceedings of the 51st IEEE Symposium on Foundations of Computer Science*, pp. 319–328. IEEE Computer Society Press.

Kemeny, J. (1959). Mathematics without numbers. *Daedalus*, *88*, 577–591.

Lenstra, Jr., H. (1983). Integer programming with a fixed number of variables. *Mathematics of Operations Research*, *8*(4), 538–548.

Liu, H., Feng, H., Zhu, D., & Luan, J. (2009). Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, *410*(27–29), 2746–2753.

Liu, H., & Zhu, D. (2010). Parameterized complexity of control problems in maximin election. *Information Processing Letters*, *110*(10), 383–388.

Maudet, N., Lang, J., Chevaleyre, Y., & Monnot, J. (2010). Possible winners when new candidates are added: The case of scoring rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pp. 762–767. AAAI Press.

McCabe-Dansted, J., Pritchard, G., & Slinko, A. (2008). Approximability of Dodgson's rule. *Social Choice and Welfare*, *31*(2), 311–330.

McLean, I., & Lorrey, H. (2006). Voting in the medieval papacy and religious orders. Report 2006-W12, Nuffield College Working Papers in Politics, Oxford, Great Britain.

McLean, I., & Urken, A. (1995). *Classics of Social Choice*. University of Michigan Press.

Meir, R., Procaccia, A., Rosenschein, J., & Zohar, A. (2008). The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, *33*, 149–178.

Menton, C. (2010). Normalized range voting broadly resists control. Tech. rep. arXiv:1005.5698 [cs.GT], arXiv.org.

Meskanen, T., & Nurmi, H. (2008). Closeness counts in social choice. In Braham, M., & Steffen, F. (Eds.), *Power, Freedom, and Voting*. Springer-Verlag.

Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.

Papadimitriou, C. (1994). *Computational Complexity*. Addison-Wesley.

Procaccia, A., & Rosenschein, J. (2007). Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, *28*, 157–181.

Saari, D., & Merlin, V. (2000). A geometric examination of Kemeny's rule. *Social Choice and Welfare*, *17*(3), 403–438.

Schöning, U. (1986). Complete sets and closeness to complexity classes. *Mathematical Systems Theory*, *19*(1), 29–42.

Shoham, Y., & Leyton-Brown, K. (2009). *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press.

Walsh, T. (2009). Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 324–329. AAAI Press.

Xia, L., & Conitzer, V. (2008a). Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pp. 109–118. ACM Press.

Xia, L., & Conitzer, V. (2008b). A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the 9th ACM Conference on Electronic Commerce*, pp. 99–108. ACM Press.

Xia, L., Zuckerman, M., Procaccia, A., Conitzer, V., & Rosenschein, J. (2009). Complexity of unweighted manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 348–353. AAAI Press.

Young, H., & Levenglick, A. (1978). A consistent extension of Condorcet's election principle. *SIAM Journal on Applied Mathematics*, *35*(2), 285–300.

Zuckerman, M., Procaccia, A., & Rosenschein, J. (2009). Algorithms for the coalitional manipulation problem. *Artificial Intelligence*, *173*(2), 392–412.