Multimode Control Attacks on Elections

Piotr Faliszewski
Dept. of Computer Science
AGH University of Science and Technology, Kraków
Poland

Edith Hemaspaandra
Dept. of Computer Science
Rochester Institute of Technology
Rochester, NY 14623
USA

Lane A. Hemaspaandra
Dept. of Computer Science
University of Rochester
Rochester, NY 14627
USA

Abstract

In 1992, Bartholdi, Tovey, and Trick [1992] 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, and 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, Tovey, and Trick [1989; 1992] initiated a line of research whose goal is to protect elections from various attacking actions intended to skew their results. Bartholdi, 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 (e.g., [Bartholdi *et al.*, 1989; 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, Tovey, and Trick's work by providing new models, new results, and new perspectives. However, until now, nobody has considered a situation in which an attacker combines multiple standard attack types into a single attack—let us call it a *multipronged* (i.e., *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. (The framework of multiprong control can be extended to include manipulation and bribery. For example, one can define a new control type, "alter voters," allowing one to alter the votes of up to some input number of voters, and we could include that in our model below. This integrates unpriced bribery—which many recent papers simply refer to as "manipulation"—into the multiprong control model. However, due to space limitations the present paper focuses just on multiprong attacks whose prongs are all already existing control types.)

Given a type of multiprong control, we analyze whether it is possible to compute in polynomial time an optimal attack of this type or whether even recognizing a possibility 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 have not been able to find examples of that behavior in existing real-world election systems, but we have constructed artificial election systems that display precisely this behavior (see the open questions section). Our paper's core work studies the (b) case and shows that even attacks having multiple facets can in many cases be planned with perfect efficiency. (Such proofs 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 shoot beyond polynomial time.)

Related work. Since the seminal paper of Bartholdi, Tovey, and Trick [1992] much research has been dedicated to studying the complexity of control in elections. Bartholdi, Tovey, and Trick [1992] considered constructive control only (i.e., scenarios where the goal of the attacker is to ensure some candidate's victory) and Hemaspaandra, Hemaspaandra, and Rothe [2007a] extended their work to the destructive case (i.e., scenarios in which the goal is to prevent someone from winning). The Holy Grail of control research is finding a natural election system (with a polynomial-time winner algorithm) that is resistant to all the standard types of control, i.e., for which all the types of control are NP-hard. Hemaspaandra, Hemaspaandra, and Rothe [2007b] showed that there exist highly resistant artificial election systems, and Faliszewski et al. [2008] showed that Copeland voting is not too far from the goal mentioned above. Erdélyi, Rothe, and Nowak [2008] showed a system with even more resistances than Copeland, but in a slightly nonstandard voter model.

Going in a somewhat different direction, Meir et al. [2008] bridged the notions of constructive and destructive control via 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. Related to control, but in a quite different model, is the work of Zuckerman et al. [2008] on quota manipulation in weighted voting games.

There is a growing body of work on manipulation that regards frequency of (non)hardness of election problems (see, e.g., [Conitzer and Sandholm, 2006; Friedgut *et al.*, 2008; Dobzinski and Procaccia, 2008; Xia and Conitzer, 2008b; 2008a]). This work studies whether a given NP-hard election problem (to date only manipulation/winner problems have been studied) can be often solved in practice (assuming some distribution of votes). Such results are of course very relevant when one's goal is to protect elections from manipulative actions. However, in this paper we typically take the role of an attacker and design control algorithms that are fast on *all* instances.

Faliszewski et al. [to appear] provides an overview of some complexity-of-election issues.

Organization. In Section 2 we present the standard model of elections and describe relevant voting systems. In Section 3 we introduce multiprong control and provide initial results regarding multiprong control, and show how existing susceptibility, vulnerability, and resistance results interact with this model. In Section 4 we provide complexity analysis of candidate control in maximin elections, showing how multiprong control is useful in doing so. Section 5 provides conclusions and open problems.

2 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 via his or her preference list. 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 $c_1 > c_2 > c_3$. 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.

An election system is an algorithm that given an election (C,V) outputs a subset $W \subseteq C$, the winners of the election. In the nonunique-winner model each member of W *wins* the election, but in the unique-winner model a candidate has to be the only member of W to claim victory. We take the unique-winner model as the default in this paper as is most common in studies of control.

We consider the following five voting systems: plurality, Copeland, maximin, approval, and Condorcet. Except Condorcet, each of them assigns points to candidates and elects those that receive the most points. Let $E = (C,V)$ be an election, where $C = \{c_1,\ldots,c_m\}$ and $V = (v_1,\ldots,v_n)$. 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 has 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 of the second and third.

We use $\text{score}_E(c_i)$ 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 $N_E(c,c') > N_E(c',c)$. Clearly, each election has at most one Condorcet winner. (Not every election has a Condorcet winner, but our notion of an election system allows that.)

3 Control and Multiprong Control

Multiprong control model. In this section we introduce multiprong control, that is, control types that combine several standard types of control. We consider combinations of control via adding/deleting candidates/voters.

Definition 3.1. *Let $\mathcal{E}$ be an election system. In the unique-winner*[1] *constructive $\mathcal{E}$-AC+DC+AV+DV 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) four nonnegative integers, k_{AC}, k_{DC}, k_{AV}, and k_{DV}.

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:

[1] One can easily adapt the definition to the nonunique-winner model.

(e) *p is a unique winner of $\mathscr{E}$ election* $((C-C')\cup A',(V-V')\cup W')$,

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

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

In the unique-winner, destructive variant of the problem, we replace item (e) above with: "p is not *a unique winner of $\mathscr{E}$ election* $((C-C')\cup A',(V-V')\cup W')$." *(In addition, in the destructive variant we refer to p as "the despised candidate" rather than as "the preferred candidate.")*

The phrase AC+DC+AV+DV in the problem name corresponds to four of the standard types of control: adding candidates (AC), deleting candidates (DC), adding voters (AV), and deleting voters (DV); we will refer to these four as the *basic* types of control.

Almost always instead of considering all of AC, DC, AV, and DV we are interested in some subset of them, and thus we consider special cases of the AC+DC+AV+DV problem. For example, we write DC+AV (without the other acronyms) to refer to a variant of the AC+DC+AV+DV problem where only deleting candidates and adding voters is allowed (i.e., where we fix $k_{\mathrm{AC}} = k_{\mathrm{DV}} = 0$). If we name only a single type of control, we in effect obtain one of the standard control problems. We for historical reasons consider also a special case of the AC control type, denoted $\mathrm{AC_u}$ (and called control by adding an unlimited number of candidates), where there is no limit on the number of candidates to add, i.e., $k_{\mathrm{AC}} = \|A\|$.

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 $\|C'\| + \|D'\| + \|V'\| + \|W'\| \le k$). While one could make arguments as to which model is more appropriate, their computational complexity is related.

Theorem 3.2. *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 $\mathscr{E}$ be an election system. We will describe the idea of our proof on the example of the constructive $\mathscr{E}$-AC+AV problem. The idea easily 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 $\mathscr{E}$-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, $0 \le \ell \le k$ 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 $\mathscr{E}$-AC+AV problem. Clearly, this algorithm is correct and runs in polynomial time. □

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 *et al.*, 2006]. Analysis of such a priced control is beyond the scope of the current paper.

Susceptibility, Immunity, Vulnerability, and Resistance. As is standard in the election-control literature, we consider vulnerability, immunity, susceptibility, and resistance to control. Let $\mathscr{E}$ be an election system and let $\mathscr{C}$ be a type of control. We say that $\mathscr{E}$ is susceptible to constructive (destructive) $\mathscr{C}$ control if there is a scenario in which effectuating $\mathscr{C}$ makes someone become (stop being) the unique winner of some $\mathscr{E}$ election E. $\mathscr{E}$ is immune to constructive (destructive) $\mathscr{C}$ control if it is not susceptible to it. We say that $\mathscr{E}$ is vulnerable to constructive (destructive) $\mathscr{C}$ control if it is susceptible to $\mathscr{C}$ and there is a polynomial-time algorithm that decides the constructive (destructive) $\mathscr{E}$-$\mathscr{C}$ problem. (And this paper's algorithms—both those given and those omitted—go further and in fact will produce the successful control action.) $\mathscr{E}$ is resistant to constructive (destructive) $\mathscr{C}$ control if it is susceptible to it and the constructive (destructive) $\mathscr{E}$-$\mathscr{C}$ problem is NP-hard.

The next theorems describe how multiprong control problems can inherit susceptibility, immunity, and resistance from the basic control types that they are built from.

Theorem 3.3. *Let $\mathscr{E}$ be an election system and let $C_1 + \cdots + C_k$ be a variant of multiprong control (so $1 \le k \le 4$ and each C_i is a basic type of control). $\mathscr{E}$ is susceptible to constructive (destructive) $C_1 + \cdots + C_k$ control if and only if $\mathscr{E}$ 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 $\mathscr{E}$ 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. □

Theorem 3.4. *Let $\mathscr{E}$ be an election system and let $C_1 + \cdots + C_k$, $1 \le k \le 4$, be a variant of multiprong control. If for some i, $1 \le i \le k$, $\mathscr{E}$ is resistant to constructive (destructive) C_i control, then $\mathscr{E}$ is resistant to constructive (destructive) $C_1 + \cdots + C_k$ control.*

This theorem can be shown from Theorem 3.3 plus the fact that each individual prong's stand-alone control problem is essentially (give or take syntax) an embedded subproblem of each multiprong control problem that includes it.

Theorem 3.4 immediately yields many "free" resistance results based on the previous work on control. But we will focus on the more interesting issue of proving that even multi-

prong control is easy for some election systems whose control has already been studied (in the "combining vulnerabilities" section) and for candidate control in maximin (Section 4).

In general, we do not consider partition cases 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 partition the candidate set C into C_1 and C_2 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 either W_1 or W_2 contains more than one candidate, we replace it with an empty set (ties eliminate). The candidates who win election $(W_1 \cup W_2, V)$ are the winners of the whole 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. Clearly, it is impossible to make p a winner via 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 a 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.

Combining vulnerabilities. In the next theorem we show that for all election systems considered in [Bartholdi *et al.*, 1992], [Hemaspaandra *et al.*, 2007a], and [Faliszewski *et al.*, 2008], all constructive vulnerabilities to AC, DC, AV and DV combine, and all destructive vulnerabilities to AC, DC, AV, and DV combine. That is, for each election system studied in these three papers, if it is separately vulnerable to some basic control types $C_1,\ldots,C_k$, where each $C_i \in \{\text{AC}, \text{DC}, \text{AV}, \text{DV}\}$, it is also vulnerable to $C_1 + \cdots + C_k$.

Theorem 3.5. *(a) Plurality is vulnerable to both constructive AV+DV control and destructive AV+DV control. (b) Both Condorcet and approval are vulnerable to destructive AC+AV+DV control. (c) For each rational α, $0 \le \alpha \le 1$, Copeland$^\alpha$ is vulnerable to destructive AC+DC control.*

Proof. (a) Let us consider an instance I of constructive plurality-AV+DV 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 those voters who vote for candidates who still have more points than p. If p becomes a unique winner before we exceed the number of voters we can delete, then accept. Otherwise reject. We omit the easy proof for the destructive case.

(b) Let I be an instance of destructive Condorcet-AC+AV+DV, 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.
2. Delete as many voters who prefer p to c as possible.

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 easy 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).

We omit the analogous proof for the case of approval.

(c) The idea is to combine Copeland$^\alpha$ destructive-AC and destructive-DC algorithms [Faliszewski *et al.*, 2008], however we omit the proof due to space restrictions. □

We mention that by using techniques from plurality bribery models, such as regarding weighted voters/etc., we can also prove vulnerability for weighted/etc. plurality AV+DV control.

4 Candidate Control in Maximin

In this section we initiate the study of control in the maximin election system. Maximin is related to Copeland voting in that both are, loosely speaking, defined in terms of the pairwise head-to-head contests, and thus one might expect that both systems would show similar resistance to control. In fact, there are very interesting differences.

As is the case for Copeland$^\alpha$, $0 \le \alpha \le 1$, maximin is resistant to control by adding candidates.

Theorem 4.1. *Maximin is resistant to constructive AC control.*

Proof. It is easy to see that the problem is in NP. To show completeness, we give a reduction from the well-known NP-complete problem *exact cover by 3-sets* (X3C, see, e.g., [Garey and Johnson, 1979]). In X3C we are given a pair $(B,\mathscr{S})$, where $B = \{b_1,\ldots,b_{3k}\}$ is a set of $3k$ elements and $\mathscr{S} = \{S_1,\ldots,S_n\}$ is a set of 3-subsets of B. We ask whether there is a subset S' of exactly k elements of $\mathscr{S}$ such that their union is exactly B (we call such a set S' an exact cover of B).

Given an instance of X3C as described above, 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).) Before we describe preference lists of voters in V, let us describe a useful convention for specifying preference lists: Putting some set D of candidates as an item in a preference list means listing all the members of this set in some fixed, arbitrary order, and listing $\overleftarrow{D}$ means listing all the members of D, but in the reverse order.

Voters in V have the following preferences. For each $S_i \in \mathscr{S}$, 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 the unique winner of $(C \cup A', V)$ if and only if $(B, \mathscr{S})$ 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 $N_{E'}(b_i, b_j) = n+1$, $N_{E'}(p, b_i) = n+1$, 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 $\mathscr{S}$ 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 easy to see that p is the unique winner of E'' if and only if S'' is an exact cover of B. □

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, very interestingly, maximin is also vulnerable to DC control, and in fact even to AC$_\text{u}$+DC control. Intuitively, in constructive AC$_\text{u}$+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 his or her score from increasing). Studying constructive AC$_\text{u}$+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 4.2. *Maximin is vulnerable to constructive AC$_\text{u}$+DC control.*

Proof. We give a polynomial-time algorithm for constructive maximin-AC$_\text{u}$+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 are sets $A' \subseteq A$ and $C' \subseteq C$ such that (a) $\|C'\| \leq k_\text{DC}$ and (b) p is the unique winner of election $((C - C') \cup A', V)$. If $k_\text{DC} \geq \|C\| - 1$, we accept immediately because we can delete all candidates but p. Otherwise, we use the following algorithm.

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 $Q(k) = \{c_i \mid c_i \in C \cup A \wedge N_{E'}(p, c_i) < k\}$. Intuitively, $Q(k)$ is the set of candidates in E' that prevent p from having exactly k points.

For each value of $k \in P$, 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 the 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 easy 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 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 $\text{score}_{E''}(d) \geq k$ (break from the loop if no such candidate exists). (3) Add d to D and jump back to step (1). We accept if $C \cap D \leq k_\text{DC}$ and we proceed to the next value of k otherwise.[2] 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 easy 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 is by deleting those candidates. Also, clearly, 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. □

Maximin is also vulnerable to destructive AC+DC control. (We omit the proof—which is easier than the above proof—due to space limits.)

Theorem 4.3. *Maximin is vulnerable to destructive AC+DC control.*

The focus of this section is on candidate control, but we do mention that maximin is resistant to voter control. (We omit the proof due to space limits.)

[2] 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 $A - D$.

Theorem 4.4. *Maximin is resistant to constructive and destructive AV control and to constructive and destructive DV control.*

5 Conclusions and Open Problems

We have shown that combining various types of control into multiprong control attacks is a useful technique. It allows us to study more realistic control models, 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.

The research presented in this paper leads to several open questions. We are very pressingly interested in finding an example of an existing, natural election system that is resistant to some multipronged control $C_1 + \ldots + C_k$ even though it is vulnerable to each separate C_i. Although space does not allow the proof to be included here (it will be included in the full version), we have proven that there is an election system that is vulnerable to constructive AV control and to constructive AC control, yet is resistant to constructive AV+AC control. However, the system we construct to achieve this is a new, artificial system and not an existing, natural system such as those discussed in the rest of this paper.

As mentioned earlier, the framework of multiprong control can be extended to include manipulation/bribery. We'll discuss this further in the full version.

An involved, long-term research goal is to study multiple control attackers competing with each other. It is interesting to consider both game-theoretic scenarios and situations in which, for example, an attacker seeks a control action that succeeds irrespective of the action of the other attacker.

Acknowledgments

Supported in part by NSF grants CCF-0426761 and IIS-0713061, AGH-UST grant 11.11.120.777, the ESF's EUROCORES program LogICCC, and Friedrich Wilhelm Bessel Research Awards to Edith Hemaspaandra and Lane A. Hemaspaandra. We thank the anonymous IJCAI referees for helpful comments.

References

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

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

[Conitzer and Sandholm, 2006] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of AAAI-06*, pages 627–634. AAAI Press, July 2006.

[Dobzinski and Procaccia, 2008] S. Dobzinski and A. Procaccia. Frequent manipulability of elections: The case of two voters. In *Proceedings of WINE-08*, pages 653–664. Springer-Verlag *Lecture Notes in Computer Science #5385*, December 2008.

[Erdélyi *et al.*, 2008] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting broadly resists control. In *Proceedings of MFCS-08*, pages 311–322. Springer-Verlag *Lecture Notes in Computer Science #5162*, August 2008.

[Faliszewski *et al.*, 2006] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of bribery in elections. In *Proceedings of AAAI-06*, pages 641–646. AAAI Press, July 2006. Full version to appear in the *Journal of Artificial Intelligence Research*.

[Faliszewski *et al.*, 2008] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Copeland voting fully resists constructive control. In *Proceedings of AAIM-08*, pages 165–176. Springer-Verlag *Lecture Notes in Computer Science #5034*, June 2008. Full version to appear in the *Journal of Artificial Intelligence Research*.

[Faliszewski *et al.*, to appear] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer, to appear.

[Friedgut *et al.*, 2008] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *Proceedings of FOCS-08*, pages 243–249. IEEE Computer Society, October 2008.

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

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

[Hemaspaandra *et al.*, 2007b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *Proceedings of IJCAI-07*, pages 1308–1314. AAAI Press, January 2007.

[Meir *et al.*, 2008] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.

[Xia and Conitzer, 2008a] L. Xia and V. Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of ACM EC-08*, pages 109–118. ACM Press, July 2008.

[Xia and Conitzer, 2008b] L. Xia and V. Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of ACM EC-08*, pages 99–108. ACM Press, July 2008.

[Zuckerman *et al.*, 2008] M. Zuckerman, P. Faliszewski, Y. Bachrach, and E. Elkind. Manipulating quota value in weighted voting games. In *Proceedings of AAAI-08*, pages 215–220. AAAI Press, July 2008.