Control in the Presence of Manipulators: Cooperative and Competitive Cases[1]

Zack Fitzsimmons
GCCIS
RIT
Rochester, NY 14623

Edith Hemaspaandra
Dept. of Comp. Science
RIT
Rochester, NY 14623

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

Abstract

Control and manipulation are two of the most studied types of attacks on elections. In this paper, we study the complexity of control attacks on elections in which there are manipulators. We study both the case where the "chair" who is seeking to control the election is allied with the manipulators, and the case where the manipulators seek to thwart the chair. In the latter case, we see that the order of play substantially influences the complexity. We prove upper bounds, holding over every election system with a polynomial-time winner problem, for all standard control cases, and some of these bounds are at the second or third level of the polynomial hierarchy, and we provide matching lower bounds to prove these tight. Nonetheless, for important natural systems the complexity can be much lower. We prove that for approval and plurality elections, the complexity of even competitive clashes between a controller and manipulators falls far below those high bounds, even as low as polynomial time. Yet for a Borda-voting case we show that such clashes raise the complexity unless NP = coNP.

1 Introduction

Elections are an important tool in reaching decisions, in both human and online settings. With the increasing importance of the online world and multiagent systems, the use of elections in computer-based settings will but increase. Unfortunately, given the relentless growth in the power of computers, it is natural to worry that computers will also be increasingly brought to bear in planning manipulative attacks on elections.

The two most computationally studied types of attacks on elections are known as "control" and "manipulation." Both were introduced by Bartholdi, Tovey, and Trick [BTT89,BTT92]. In control, an agent, usually referred to as "the chair," tries to make a given candidate win by adding/deleting/partitioning voters or candidates. In manipulation, some nonmanipulative voters and a coalition of manipulative voters vote under some election system, and the manipulative voters seek to make a given candidate win. There is a broad literature on the computational complexity of control, and on the computational complexity of manipulation.

This paper considers control attacks against elections that contain manipulators. We consider both the cooperative and the competitive cases.

In the cooperative case, the chair is allied with the manipulative coalition. For example, perhaps during a CS department's hiring, the department chair, who happens to also be the senior member of the systems group, is mounting a control by partition of voters attack (in which he or she is dividing the faculty into two subcommittees, one to decide which candidates are strong enough teachers to

[1]Supported in part by NSF grants CCF-0915792, CCF-1101452, and CCF-1101479. An earlier version of this paper [FHH13a] was presented at the Twenty-Third International Joint Conference on Artificial Intelligence (IJCAI-2013).

merit further consideration, and one to decide which candidates are strong enough researchers to merit further consideration), and also is able to directly control the votes of every one of his or her fellow members of the department's systems faculty. The chair's goal is to make some particular candidate, perhaps Dr. I. M. Systems, be the one chosen for hiring.

In the even more interesting competitive case, which can be thought of in a certain sense as *control versus manipulation*, we'll assume that the manipulative coalition's goal is to keep the chair from achieving the chair's goal. For the competitive case, we'll look at the case where the chair acts before the manipulators, and at the case where the manipulators act before the chair. For control attacks by so-called partition, in which there is a two-round election, we'll consider the case where the manipulators can change their votes in the second round, and the case where the manipulators cannot change their votes in the second round.

Our main contributions are as follows.

Building on the existing notions of control and manipulation, we give natural definitions that capture our cooperative and competitive notions as problems whose computational complexity can be studied. We note how existing hardness results for control and manipulation are, or are not, inherited by our problems. We prove upper bounds on our problems. For the competitive case, some of these are as high as $\mathrm{NP}^{\mathrm{NP}}$, $\mathrm{coNP}^{\mathrm{NP}}$, and $\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$.

Those are very high levels of complexity. Yet they are tight: We show that there are election systems (having p-time winner problems) for which most of those high bounds have matching lower bounds, yielding completeness for those classes.

The fact that the upper bounds are tight does not mean that those high complexity levels hold for every election system having a p-time winner problem. Indeed, we show that that is not the case. In particular, for the important election systems approval and plurality, we show that the complexity of control in the presence of manipulators, whether cooperative or competitive, falls far below those upper bounds, even falling as low as polynomial time.

We also obtain cases, for veto (Theorem 4.4) and Borda (Theorem 4.5) elections, where competitive control-plus-manipulation is variously easier or harder than one might expect from the separate control and manipulation cases.

2 Related Work

The idea of enhancing control with manipulative voters has been mentioned in the literature, namely, in a paragraph of [FHH11]. That paper cooperatively integrated with control, to a certain extent, a different attack type known as bribery [FHH09]. In that paper's conclusions and open directions, there is a paragraph suggesting that manipulation could and should also be integrated into that paper's "multiprong setting," and commending such future study to interested readers. That paragraph was certainly influential in our choice of this direction. However, it is speaking just of the cooperative case, and provides no results on this since it is suggesting a direction for study.

The lovely line of work about "possible winners" [KL05] in the context of adding candidates might at first seem to be merging manipulation and control. We refer to the line of work explored in [CLMM10,BRR11,XLM11,CLM+12]. That work considers an election with an initial set of candidates, over which all the voters have complete preferences, and a set of additional candidates over which the voters initially have no preferences, and asks whether, if the entire set of additional candidates is added, there is some way of extending the initial linear orders to now be over all the candidates, in such a way that a particular initial candidate becomes a winner of the election. Although on its surface this might feel like a cross between manipulation and control by adding candidates, in fact, in this interesting problem there is no actual choice regarding the addition of candidates; all are simply added. Thus this problem is a generalization of manipulation (as the papers note), that happens to be done in a setting that involves adding candidates. It is not a generalization of control by adding, or even so-called unlimited adding, of candidates, as in those the chair must choose what collection of candidates to add. In short, unlike control and unlike this paper, there is no existentially quantified action by a chair. (An interesting recent paper of Baumeister et al. [BRR$^+$12] uses the term possible winner in a new, different way, to speak of weights rather than preferences initially being partially unset. That particular paper's question, as that paper notes, can be seen as a generalization of control by adding and deleting voters. However, their notion is not a generalization of manipulation.)

The present paper does combine control and manipulation, with both those playing active—and sometimes opposing—roles. Manipulation alone has been extensively studied in a huge number of papers, starting with the seminal paper of [BTT89] (see also [BO91]), which covered the constructive case. The destructive cases (i.e., those where the goal is to keep a particular candidate from winning) are due to [CSL07]. Control alone has been extensively studied in many papers, with the seminal paper being [BTT92], which covered the constructive case. The destructive cases were first studied in [HHR07]. There has been quite a bit of work on finding systems for which conducting various types of manipulation is hard, or for which conducting most types of control attacks is hard, see, e.g., [ER10,EPR11,ENR09,FHHR09a,HHR09,Men13,MS13,PX12] or the surveys [FHHR09b,FHH10].

In the present paper, we'll see that who goes first, the chair or the manipulators, is important in determining what complexity upper bounds apply. Order has also been seen to be important in the study of so-called online control attacks [HHR12b,HHR12a], and of online manipulation attacks [HHR14]. However, the papers just mentioned are separately about control, and about manipulation. In contrast we are mostly interested in when both are occurring, and especially when the two attacks are in conflict with each other.

The present paper also looks at how revoting affects the complexity of elections that involve both control and manipulation. It is important to mention that, for the case of just manipulation, [NW12, NW13] (see also [FHH13c]) have recently discussed revoting, and give an example that shows that revoting can sometimes be a valuable tool for the manipulator.

3 Preliminaries

An election (a.k.a. a social choice correspondence) maps from a finite candidate set C and a finite vote collection V to a set, $W \subseteq C$, called the winner(s) [SL09]. Voters come without names, and the votes are input as a list, i.e., as ballots. For approval elections, each ballot is a length-$\|C\|$ 0-1 vector indicating whether each candidate is disapproved or approved. The candidate getting the most approvals is the winner (or winners if candidates tie for most). For all other systems we discuss, each ballot is a tie-free linear ordering of the candidates. For plurality elections, each voter gives one point to his or her top choice and zero to the rest. For veto elections, each voter gives zero points to his or her bottom choice and one to the rest. For Borda elections, each voter gives zero points to his or her bottom choice, one point to his or her next to bottom choice, and so on through giving $\|C\| - 1$ points to his or her top choice. In the three systems just mentioned, the winner is the candidate(s) who receives the most points. In a Condorcet election—[BTT92] recast the notion of a Condorcet winner [Con85] into an election system of sorts, in this way, and used it as one of their focus cases in their seminal control study—a candidate p is a winner exactly if for each other candidate b it holds that strictly more than half the votes cast prefer p to b. Unlike the systems from earlier in this paragraph, Condorcet elections on some inputs may have no winners.

An election system $\mathcal{E}$ is said to have a p-time winner problem if there is a polynomial-time algorithm that on input C, V, and $p \in C$, determines whether p is a winner under $\mathcal{E}$ of the election over C with the votes being V.

We assume the reader is aware of the NP, $\mathrm{coNP}^{\mathrm{NP}}$, $\mathrm{NP}^{\mathrm{NP}}$, and $\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$ levels of the polynomial hierarchy (the "exponentiation" notation denotes oracle class, informally put, having unit-cost access to a set of one's choice from the given class) [MS72,Sto76] and with many-one reductions (which here always means many-one polynomial-time reductions). As is standard, we use $\leq_{\mathrm{m}}^{\mathrm{p}}$ to denote many-one reductions. There are far fewer completeness results for levels of the hierarchy beyond NP, such as the abovementioned ones, than there are for NP; a collection of and discussion of such results can be found in [SU02a,SU02b]. Completeness and hardness here are always with respect to many-one reductions.

Our hardness results are worst-case results. However, it is known that if there exists even one set that is hard for NP (and note that all sets hard for $\mathrm{coNP}^{\mathrm{NP}}$, $\mathrm{NP}^{\mathrm{NP}}$, or $\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$ are hard for NP) and has a (deterministic) heuristic algorithm whose asymptotic error rate is subexponential, then the polynomial hierarchy collapses. See [HW12] for a discussion of that, and an attempt to reconcile that with the fact that in practice heuristics often do seem to do well, including for some cases related to elections, see, e.g., [Wal09].

For a complete description of all the many control types touched on in this paper, we refer the reader to the detailed definitions given in [FHHR09a]. But we briefly define here the underlying control types we use in our sample proofs. Given as input an election, (C, V), a distinguished candidate $p \in C$, and an integer $k \geq 0$, the constructive (respectively, destructive) control by deleting voters—for short CCDV (respectively, DCDV)—problem for an election system $\mathcal{E}$ asks whether there is some choice of at most k votes such that if they are removed, p is a winner (respectively, is not a winner) of the given election under $\mathcal{E}$. We are in the so-called nonunique-winner model, and so we ask about making p "a winner" rather than "the one and only winner," which is the so-called unique-winner model.[2] Each of those problems has an adding voters (AV) analogue, in which one has a collection of registered votes that definitely are cast, and has a collection of "unregistered" votes, and the question is whether there is some choice of at most k unregistered votes such that if they are added, the goal is met. These types of control are motivated by issues ranging from voter suppression to targeted phone calls to get-out-the-vote drives. There are the natural analogous types for adding and deleting candidates, AC and DC (note: in the destructive control by deleting candidates case—DCDC—deleting p is not allowed [BTT92]). The partition types are called runoff partition of candidates (RPC), partition of candidates (PC), and partition of voters (PV). In PV, the input is just C, V, and p, and the constructive (destructive) question is whether there exists a partition of the voters into V_1 and V_2 such that if the candidates who survive at least one of the elections (C, V_1) and (C, V_2) move on to a final election among just them with the vote set V, p is (is not) a winner. Here, there are two models for what "survive" means. In the ties eliminate (TE) model, to move forward one must uniquely win a first-round election; in the ties promote (TP) model, it suffices to be *a* winner of a first-round election. One of our sample proofs will be about CCPV-TE.

As to manipulation, the constructive (destructive) unweighted coalitional manipulation CUCM (DUCM) problem under election system $\mathcal{E}$ has as input C, $p \in C$, a collection of nonmanipulative voters each specified by his or her vote (linear order or approval vector, as appropriate), and a collection of manipulator voters (coming in as blank slates, e.g., input as a string 1^j to indicate there are j of them), and the question is whether there is some way of setting the votes of the manipulative coalition so that p is (is not) a winner of the resulting election under system $\mathcal{E}$ with

[2] Many of our results also hold in the other model, but the nonunique-winner model is probably the better, more natural model on which to focus in general.

those votes and the nonmanipulative votes both being cast.

Our model of allowing control in the presence of manipulators varies the standard control definitions to allow some of the voters to be manipulators, and thus to come in as blank slates. We mention that for AV, it is legal to have manipulators among the registered and/or the unregistered votes. For the cooperative cases, the question is whether the chair can choose preferences for the manipulators such that, along with using his or her legal control-decision ability for that control type, p can be made (precluded from being) a winner. We denote these types by adding in an "M+," e.g., plurality-M+CCAV. For the competitive cases, we can look at the case where the manipulative coalition sets its votes and then the chair chooses a control action, and we call that MF for "manipulators first." Or we can have the chair control first and then the manipulators set their votes, which we call CF for "chair first." Since the manipulators seek to thwart the chair, the case Borda-CCAV-MF, for example, asks whether under Borda, no matter how the manipulative voters, moving first, set their votes, there will exist some choice of at most k unregistered voters that the chair can add so that p is a winner. For partition cases, we add the string "-revoting" to indicate that after the first-round elections occur, the manipulators can change their votes in the final election.

To allow many things to be spoken of compactly, we use "stacked" notation to indicate every possible string one gets by reading across and taking one choice from each bracket one encounters on one's path across the expression. So, for example, $\mathrm{CC}\begin{bmatrix}\mathrm{A}\\ \mathrm{D}\end{bmatrix}\mathrm{V}\text{-}\begin{bmatrix}\mathrm{CF}\\ \mathrm{MF}\end{bmatrix}$ refers to four control types, not just two, and $\begin{bmatrix}\mathrm{C}\\ \mathrm{D}\end{bmatrix}\mathrm{C}\begin{bmatrix}\begin{bmatrix}\mathrm{A}\\ \mathrm{D}\end{bmatrix}\begin{bmatrix}\mathrm{C}\\ \mathrm{V}\end{bmatrix}\\ \begin{bmatrix}\mathrm{PC}\\ \mathrm{RPC}\\ \mathrm{PV}\end{bmatrix}\text{-}\begin{bmatrix}\mathrm{TE}\\ \mathrm{TP}\end{bmatrix}\end{bmatrix}$ refers to $2 \times (2 \times 2 + 3 \times 2) = 20$ control types.

Notice that for our competitive setting, we seem to be asymmetrically focusing on things from the perspective of the chair. That is, regardless of whether the chair moves first or whether the manipulators move first, our problems are always posed in terms of the chair's constructive or destructive goal regarding the candidate p. It would be natural to ask—and indeed, a conference referee asked us to address the issue of—whether one can interestingly study the competitive problem from the perspective of the manipulators rather than that of the chair. That is, in the MF case for example, one would ask whether the manipulators can act so as to achieve or block victory for p, regardless of the actions of the chair that follow. And one could similarly look at the CF case from the manipulators' perspective. After all, in many real-world settings, what one cares about may well be the perspective of the manipulators. Thus being able to address this issue would itself be an additional motivation for our paper. Fortunately, in the competitive case—and this holds in both the nonunique-winner model and the unique-winner model, and holds for all types of constructive and destructive attacks discussed here—the chair achieving his or her goal in the model where we view things from the perspective of the chair is precisely the same as the manipulators failing to meet their goal in the model where we view things from the perspective of the manipulators. This follows from the definitions. Thus, this paper is implicitly handling the case of the manipulators' perspective: For all our competitive cases, studying a constructive (respectively, destructive) attack problem from the perspective of the manipulators is exactly the same as studying the *destructive* (respectively, *constructive*) version of the same problem in the model of this paper, that is, from the perspective of the chair. (We caution that the above discussion should not be interpreted as saying that the constructive and destructive problems are each other's opposites. That is not true, although there is a partial connection between these cases, see the discussion in footnote 5 of [HHR07].)

4 Results

4.1 Inheritance

Each control type many-one reduces to each of its cooperative and to each of its competitive control-plus-manipulation variants, because for those variants the zero-manipulator cases degenerate to the pure control case. For example, $\mathcal{E}$-CCDV $\leq_{\mathrm{m}}^{\mathrm{p}}$ $\mathcal{E}$-M+CCDV and $\mathcal{E}$-CCDV $\leq_{\mathrm{m}}^{\mathrm{p}}$ $\mathcal{E}$-CCDV-MF. In particular, NP-hardness results for control inherit upward to each related cooperative and competitive case.

For manipulation, the inheritance behavior is not as broad, since partition control cannot necessarily be "canceled out" by setting a parameter to zero, as partition doesn't even have a numerical parameter. Nonpartition control types do display inheritance, but for the competitive cases there is some "flipping" of the type of control and the set involved. For each constructive (respectively, destructive) control type regarding adding or deleting candidates or voters, destructive (respectively, constructive) manipulation many-one reduces to the complement of the set capturing the competitive case of the constructive (respectively, destructive) control type combined with manipulation. For example, $\mathcal{E}$-CUCM $\leq_{\mathrm{m}}^{\mathrm{p}}$ $\overline{\mathcal{E}\text{-DCAC-CF}}$ and $\mathcal{E}$-DUCM $\leq_{\mathrm{m}}^{\mathrm{p}}$ $\overline{\mathcal{E}\text{-CCDV-MF}}$. For the cooperative cases there is no "flipping." For each constructive or destructive control type regarding adding or deleting candidates or voters, manipulation many-one reduces to the cooperative case of that control type combined with manipulation. For example, $\mathcal{E}$-CUCM $\leq_{\mathrm{m}}^{\mathrm{p}}$ $\mathcal{E}$-M+CCAC and $\mathcal{E}$-DUCM $\leq_{\mathrm{m}}^{\mathrm{p}}$ $\mathcal{E}$-M+DCAC.

4.2 General Upper Bounds and Matching Lower Bounds

Problem	CF	CF-revoting	MF	MF-revoting
$\mathcal{E}$ - $\left[\begin{smallmatrix}\mathrm{C}\\ \mathrm{D}\end{smallmatrix}\right]\mathrm{C}\left[\begin{smallmatrix}\mathrm{A}\\ \mathrm{D}\end{smallmatrix}\right]\left[\begin{smallmatrix}\mathrm{C}\\ \mathrm{V}\end{smallmatrix}\right]$	$\mathrm{NP}^{\mathrm{NP}}$	N/A	$\mathrm{coNP}^{\mathrm{NP}}$	N/A
$\mathcal{E}$ - $\left[\begin{smallmatrix}\mathrm{C}\\ \mathrm{D}\end{smallmatrix}\right]\mathrm{C}\left[\begin{smallmatrix}\mathrm{PC}\\ \mathrm{RPC}\\ \mathrm{PV}\end{smallmatrix}\right]$ - $\left[\begin{smallmatrix}\mathrm{TE}\\ \mathrm{TP}\end{smallmatrix}\right]$	$\mathrm{NP}^{\mathrm{NP}}$	$\mathrm{NP}^{\mathrm{NP}}$	$\mathrm{coNP}^{\mathrm{NP}}$	$\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$

Table 1: Upper Bounds. (N/A means not applicable.)

For elections systems with p-time winner problems, all the cooperative cases clearly have NP upper bounds. But the upper bounds for the competitive cases are far higher, falling in the second and third levels of the polynomial hierarchy, as described by the following theorem.

Theorem 4.1 *For each election system $\mathcal{E}$ having a p-time winner problem, the bounds of Table 1 hold.*[3]

Although the table's upper bounds clearly follow from the structure of the problems, the bounds are very high. Can they be improved by some cleverer approach? Or are there systems with p-time winner problems that show the bounds to be tight? The following result establishes that the latter holds; each of the cells in the table is tight for at least some cases.

[3]Where the table says N/A—not applicable—the nonrevoting bounds just to the left of the box technically still hold; we say N/A simply to be clear that revoting cannot even take place in nonpartition cases, since there is no second round.

Theorem 4.2 *1. For each of the eight problems on the top line of Table 1, and each of the columns on that line, there exists an election system $\mathcal{E}$, which has a p-time winner problem, for which the named problem is complete for the named complexity class.*

2. *For each of* CCPV-TP *and* CCPV-TE*, and each of the* CF, CF-revoting*, and* MF *columns of Table 1, there exists an election system $\mathcal{E}$, which has a p-time winner problem, for which the named problem is complete for the named complexity class.*

3. *There exists an election system $\mathcal{E}$, which has a p-time winner problem, for which* CCPV-TP-MF-revoting *is* $\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$*-complete.*

The above result says that the upper bounds are not needlessly high. They are truly needed, at least for some systems. However, the constructions proving the lower bounds are artificial and the construction involving the third level of the polynomial hierarchy is lengthy and difficult.[4] In particular, this leaves completely open the possibility that for particular, important real-world systems, even the competitive cases may be far simpler than those bounds suggest. In the coming section, we will see that indeed for some of the most important real-world systems, even in the presence of manipulators, the control problem is just as computationally easy as when there are no manipulators.

4.3 Specific Systems

Plurality is certainly the most important of election systems, and approval is also an important system. For plurality, approval, and Condorcet elections, we note that all the "M+," "CF," and "MF" cases whose control type is classified as P (i.e., as a "V" or "I" in the notation of that the 2007 table we're about to mention) in the with-no-manipulators table of complexities for that election system in [HHR07] are in P for all our cooperative and competitive cases.

Theorem 4.3 *Each problem contained in*

- $\begin{bmatrix}\text{approval}\\ \text{Condorcet}\\ \text{plurality}\end{bmatrix}\text{-M+}\begin{bmatrix}\text{C}\\ \text{D}\end{bmatrix}\text{C}\left[\begin{matrix}\begin{bmatrix}\text{A}\\ \text{D}\end{bmatrix}\begin{bmatrix}\text{C}\\ \text{V}\end{bmatrix}\\ \begin{bmatrix}\text{PC}\\ \text{RPC}\\ \text{PV}\end{bmatrix}\text{-}\begin{bmatrix}\text{TE}\\ \text{TP}\end{bmatrix}\end{matrix}\right]$,
- $\begin{bmatrix}\text{approval}\\ \text{Condorcet}\\ \text{plurality}\end{bmatrix}\text{-}\begin{bmatrix}\text{C}\\ \text{D}\end{bmatrix}\text{C}\left[\begin{matrix}\begin{bmatrix}\text{A}\\ \text{D}\end{bmatrix}\begin{bmatrix}\text{C}\\ \text{V}\end{bmatrix}\\ \begin{bmatrix}\text{PC}\\ \text{RPC}\\ \text{PV}\end{bmatrix}\text{-}\begin{bmatrix}\text{TE}\\ \text{TP}\end{bmatrix}\end{matrix}\right]\text{-CF}$, *or*
- $\begin{bmatrix}\text{approval}\\ \text{Condorcet}\\ \text{plurality}\end{bmatrix}\text{-}\begin{bmatrix}\text{C}\\ \text{D}\end{bmatrix}\text{C}\left[\begin{matrix}\begin{bmatrix}\text{A}\\ \text{D}\end{bmatrix}\begin{bmatrix}\text{C}\\ \text{V}\end{bmatrix}\\ \begin{bmatrix}\text{PC}\\ \text{RPC}\\ \text{PV}\end{bmatrix}\text{-}\begin{bmatrix}\text{TE}\\ \text{TP}\end{bmatrix}\end{matrix}\right]\text{-MF}$,

[4] The third-level case has to overcome the specific, and as far as we know new, worry that in the second round, the first-round vote of the manipulators is no longer available. Yet in a "∀∃∀" context (which is the quantifier structure that models $\mathrm{coNP}^{\mathrm{NP}^{\mathrm{NP}}}$), a particular existential choice has to handle only a particular value of the first ∀. So to make the construction work, we need to in some sense have the first-round votes, which are no longer available, still cast a clear and usable shadow forward into the second round, at least in certain cases in the image of the reduction. We achieve this, in particular by shaping the election system itself carefully to help realize this unusual effect. Otherwise, we would not be capturing the right quantifier structure.

whose corresponding control type is in P *in Table 1 of page 258 of [HHR07],*[5] *is in* P.

As an illustration, we include a proof of plurality-M+CCPV-TE $\in$ P.

Proof. Note that it is not the case that the manipulators can always simply vote for p, no matter what the chair does. For example, if the chair partitions the voters such that one of the subelections contains a voter voting $p > a > b$, and the other subelection contains 100 voters voting $a > b > p$, 101 voters voting $b > a > p$, and one manipulator, the manipulator should vote for a, so that a and b are tied in the second subelection and neither goes through to the second round. Still, we will show that if a partition of the voters and a manipulation of the manipulators exist such that p wins the election, then there exists a way for p to win when all manipulators vote for p. It follows that we can check if p can be made a winner by first having all manipulators vote for p and then running the polynomial-time algorithm for plurality-CCPV-TE from [HHR07].

So, suppose that a manipulation and a partition (V_1, V_2) exist such that p is a winner of the election. Without loss of generality, suppose p is the unique winner of (C, V_1). Then p is also the unique winner of (C, V_1) if all manipulators in V_1 vote for p. Now consider (C, V_2). As explained above, simply changing the manipulators' votes to p could have bad effects. Instead, we do the following. For each manipulator in V_2, if the manipulator votes for $a \neq p$, we change the manipulator's vote to p, move this manipulator to V_1 and for each candidate $b \neq a$, we move a voter who votes for b (if one exists) from V_2 to V_1. (If this voter is a manipulator, we also change its vote to p.) Since we added a vote for p to V_1, p will still be the unique winner of (C, V_1). Since we keep the relative scores of the candidates with positive scores constant, we never turn winners into nonwinners in (C, V_2). This implies that if (C, V_2) has a unique winner $c \neq p$, then c does not beat p in the runoff. It follows that p remains a winner of the runoff. (There was a slight problem in the argument used in this paragraph in our previous version [FHH13a,FHH13b]; that problem is now fixed here.) ❑

We now give results for veto and Borda, including, for the latter, an interesting increase in complexity.

CWCM and DWCM are the same as CUCM and DUCM, except in such weighted cases every voter has a positive integer weight, and a voter with weight w counts as w voters. Consider the case of 3-candidate weighted veto elections. The known results on this are that CWCM is NP-complete, DWCM is in P, and CCAV and CCDV are both in P. The following results, whose second part may be surprising, shows that for this system CC$\begin{bmatrix}\text{A}\\\text{D}\end{bmatrix}$V-$\begin{bmatrix}\text{CF}\\\text{MF}\end{bmatrix}$ are all in P—not NP-complete.

Theorem 4.4 *For 3-candidate weighted veto elections, the following hold.*

1. M+CC$\begin{bmatrix}\text{A}\\\text{D}\end{bmatrix}$V *are both* NP-*complete.*

2. CC$\begin{bmatrix}\text{A}\\\text{D}\end{bmatrix}$V-$\begin{bmatrix}\text{CF}\\\text{MF}\end{bmatrix}$ *are each in* P.

Proof. The first case follows directly from the fact that constructive manipulation is NP-complete [CSL07] and the inheritance observations from Section 4.1 (as the relevant result there holds even for the weighted case).

[5]This claim holds despite the fact that that table is focused on the unique-winner case, and despite the fact that the "AC" line of that table refers to so-called unlimited adding and in this paper (as is now standard) we use "AC" to refer to (limited) adding. The reason we have looked at only the P cases of the 2007 table is that due to our inheritance results, for the NP cases getting a P result will be impossible, at least if the unique-winner results of the table carry over to our nonunique-winner setting—which typically is the case and has been checked for many, but not all, cases in the table. We will cover this issue in more detail in the full version of this paper.

For the competitive cases, note that the only action that makes sense for the manipulators is to veto p. This holds regardless of whether the manipulators or the chair goes first. So, we let the manipulators veto p and then run the polynomial-time algorithm for CCAV and CCDV. ❑

3-candidate weighted Borda elections show a true increase in complexity. The underlying CWCM, DWCM, CCAV, and CCDV complexities are known to be respectively NP-complete, P, NP-complete, and NP-complete, and thus all are in NP. Yet we show that CCAV-MF is coNP-hard, and so cannot be in NP unless the polynomial hierarchy collapses to NP $\cap$ coNP.

Theorem 4.5 *For 3-candidate weighted Borda elections, the following hold.*

1. $\mathrm{M{+}CC}\left[\begin{smallmatrix}\mathrm{A}\\ \mathrm{D}\end{smallmatrix}\right]\mathrm{V}$ *are both* NP-*complete.*
2. $\mathrm{CC}\left[\begin{smallmatrix}\text{AV-CF}\\ \text{DV-}\left[\begin{smallmatrix}\mathrm{CF}\\ \mathrm{MF}\end{smallmatrix}\right]\end{smallmatrix}\right]$ *are each* NP-*hard.*
3. CCAV-MF *is* NP-*hard and* coNP-*hard.*
4. $\mathrm{CC}\left[\begin{smallmatrix}\mathrm{A}\\ \mathrm{D}\end{smallmatrix}\right]\text{V-CF}$ *is* NP-*complete.*

Proof. The first case follows directly from the fact that manipulation is NP-complete [CSL07] and the inheritance observations from Section 4.1.

The remaining NP-hardness results follow from the NP-completeness of CCAV and CCDV and the inheritance observations from Section 4.1.

To show that CCAV-CF is in NP, guess a set of voters to add, and then check that the manipulators can't make p not win. We do this by setting all manipulators to $a > b > p$, checking that p is a winner, and then setting all manipulators to $b > a > p$, and checking that p is a winner. A similar argument shows that CCDV-CF is in NP.

It remains to show that CCAV-MF is coNP-hard, i.e., that the complement of CCAV-MF is NP-hard. We will reduce from Partition. Given a nonempty sequence of positive integers $k_1, \ldots, k_t$ that sums to $2K$, we will construct an election such that there is a partition (i.e., a subsequence of $k_1, \ldots, k_t$ that sums to K) if and only if the manipulators can vote in such a way that the chair won't be able to make p a winner.

We construct the following election: We have manipulators with weights $k_1, \ldots, k_t$. The manipulators are registered voters. We have two unregistered voters, both with weight $3K - 1$. One of these voters votes $p > a > b$ and one votes $p > b > a$. We have addition limit one, i.e., the chair can add at most one voter.

If there is a partition, then the manipulators vote so that a total of K vote weight casts the vote $a > b > p$ and a total of K vote weight casts the vote $b > a > p$. So, the scores of p, a, and b are 0, $3K$, and $3K$. There is no way for the chair to make p a winner by adding at most one voter. If the chair adds the weight $3K - 1$ voter voting $p > a > b$, the score of p is $6K - 2$ and the score of a is $3K + (3K - 1) = 6K - 1$ and so p is not a winner. Adding the other voter gives a score of $6K - 2$ for p and a score of $6K - 1$ for b and again p is not a winner.

Now consider the case that there is no partition. Look at the scores of the candidates after the manipulators have voted. Without loss of generality, assume that $\mathit{score}(a) \leq \mathit{score}(b)$. Then $\mathit{score}(a) \leq 3K - 1$ (since there is no partition) and $\mathit{score}(b) \leq 4K$. Now the chair adds the weight $3K - 1$ voter voting $p > a > b$. After adding that voter, p's score is $6K - 2$, a's score is at most $(3K - 1) + (3K - 2)$ and b's score is at most $4K$. It follows that p is a winner. ❑

5 Conclusions and Open Directions

We have established general inheritance results and complexity upper bounds for control in the presence of manipulators, for both cooperative and competitive settings. We for the upper bounds provided matching lower bounds, but also showed that for many natural systems the complexity is far lower than the general upper bounds.

Many open directions remain. For example, regarding 3-candidate weighted Borda elections, we have shown that CCAV-MF is NP-hard and coNP-hard, and although our upper-bound theorem is not explicitly about weighted cases, clearly this problem, for exactly the same reason as in our upper-bound theorem, is in $\mathrm{coNP}^{\mathrm{NP}}$. But precisely where within that range does it fall? Also, what happens for real-world election systems that themselves are complex to manipulate and/or control, such as Llull, Copeland, fallback, sincere-preference approval, and Schulze elections? Do some of these systems themselves provide natural systems that might for our competitive cases be complete for some of the high complexity classes given in Table 1?

Acknowledgments

We are grateful to the anonymous IJCAI-2013 and COMSOC-2014 referees for helpful comments and suggestions.

References

[BO91] J. Bartholdi, III and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[BRR11] D. Baumeister, M. Roos, and J. Rothe. Computational complexity of two variants of the possible winner problem. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 853–860. International Foundation for Autonomous Agents and Multiagent Systems, May 2011.

[BRR+12] D. Baumeister, M. Roos, J. Rothe, L. Schend, and L. Xia. The possible winner problem with uncertain weights. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 133–138. IOS Press, August 2012.

[BTT89] J. Bartholdi, III, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[BTT92] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[CLM+12] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 64(1):74–88, 2012.

[CLMM10] Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible winners when new candidates are added: The case of scoring rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 762–767. AAAI Press, July 2010.

[Con85] J.-A.-N. de Caritat, Marquis de Condorcet. *Essai sur l'Application de L'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix.* 1785. Facsimile reprint of original published in Paris, 1972, by the Imprimerie Royale.

[CSL07] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[ENR09] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.

[EPR11] G. Erdélyi, L. Piras, and J. Rothe. The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 837–844. International Foundation for Autonomous Agents and Multiagent Systems, May 2011.

[ER10] G. Erdélyi and J. Rothe. Control complexity in fallback voting. In *Proceedings the 16th Australasian Theory Symposium*, pages 39–48. Australian Computer Society, January 2010.

[FHH09] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[FHH10] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[FHH11] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[FHH13a] Z. Fitzsimmons, E. Hemaspaandra, and L. Hemaspaandra. Control in the presence of manipulators: Cooperative and competitive cases. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 113–119. AAAI Press, August 2013.

[FHH13b] Z. Fitzsimmons, E. Hemaspaandra, and L. Hemaspaandra. Control in the presence of manipulators: Cooperative and competitive cases. Technical Report arXiv:1308.0544v1 [cs.GT], Computing Research Repository, arXiv.org/corr/, August 2013.

[FHH13c] Z. Fitzsimmons, E. Hemaspaandra, and L. Hemaspaandra. X THEN X: Manipulation of same-system runoff elections. Technical Report arXiv:1301.6118 [cs.GT], Computing Research Repository, arXiv.org/corr/, January 2013.

[FHHR09a] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[FHHR09b] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*, pages 375–406. Springer, 2009.

[HHR07] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[HHR09] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[HHR12a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Controlling candidate-sequential elections. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 905–906. IOS Press, August 2012.

[HHR12b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Online voter control in sequential elections. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 396–401. IOS Press, August 2012.

[HHR14] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online manipulation of sequential elections. *Journal of Computer and System Sciences*, 80(4):697–710, 2014.

[HW12] L. Hemaspaandra and R. Williams. An atypical survey of typical-case heuristic algorithms. *SIGACT News*, 43(4):71–89, December 2012.

[KL05] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[Men13] C. Menton. Normalized range voting broadly resists control. *Theory of Computing Systems*, 53(4):507–531, 2013.

[MS72] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129. IEEE Press, October 1972.

[MS13] C. Menton and P. Singh. Control complexity of Schulze voting. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pages 286–292. AAAI Press, August 2013.

[NW12] N. Narodytska and T. Walsh. Manipulating two stage voting rules. In *Proceedings (Workshop Notes) of the 4th International Workshop on Computational Social Choice*, pages 323–334, September 2012.

[NW13] N. Narodytska and T. Walsh. Manipulating two stage voting rules. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pages 423–430. International Foundation for Autonomous Agents and Multiagent Systems, May 2013.

[PX12] D. Parkes and L. Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1429–1435. AAAI Press, August 2012.

[SL09] Y. Shoham and K. Leyton-Brown. *Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2009.

[Sto76] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[SU02a] M. Schaefer and C. Umans. Completeness in the polynomial-time hierarchy: Part I: A compendium. *SIGACT News*, 33(3):32–49, 2002.

[SU02b] M. Schaefer and C. Umans. Completeness in the polynomial-time hierarchy: Part II. *SIGACT News*, 33(4), 2002.

[Wal09] T. Walsh. 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*, pages 324–329. AAAI Press, July 2009.

[XLM11] L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: New results coming up! In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 829–836. International Foundation for Autonomous Agents and Multiagent Systems, May 2011.