ECAI 2012
Luc De Raedt et al. (Eds.)

doi:10.3233/978-1-61499-098-7-396

Online Voter Control in Sequential Elections

Edith Hemaspaandra[1] and **Lane A. Hemaspaandra**[2] and **Jörg Rothe**[3]

Abstract. Previous work on voter control, which refers to situations where a chair seeks to change the outcome of an election by deleting, adding, or partitioning voters, takes for granted that the chair knows all the voters' preferences and that all votes are cast simultaneously. However, elections are often held sequentially and the chair thus knows only the previously cast votes and not the future ones, yet needs to decide instantaneously which control action to take. We introduce a framework that models *online voter control in sequential elections*. We show that the related problems can be much harder than in the standard (non-online) case: For certain election systems, even with efficient winner problems, online control by deleting, adding, or partitioning voters is PSPACE-complete, even if there are only two candidates. In addition, we obtain completeness for coNP in the deleting/adding cases with a bounded deletion/addition limit, and for NP in the partition cases with only one candidate. Finally, we show that for plurality, online control by deleting or adding voters is in P, and for partitioning voters is coNP-hard.

1 Introduction

The study of the computational properties of voting systems has been an exceedingly active area within computational social choice. In particular, various types of manipulation, electoral control, and bribery in voting have been classified in terms of their computational complexity (see [6]). This paper focuses on *voter control*, a model introduced by Bartholdi et al. [1], where a chair attempts to alter the outcome of an election via changing its structure by deleting, adding, or partition of voters.

To the best of our knowledge, all previous work on control (see, e.g., [1, 8, 7, 9, 4, 5]) takes for granted that the chair has full knowledge of all the voters' preferences and that all votes are cast simultaneously. However, in many settings voters vote sequentially and the chair's task in such a setting may often be quite different: Knowing only the already cast votes but not the future ones, the chair must decide *online* (i.e., in that moment) whether there exists a control action that guarantees success, no matter what votes will be cast later on. We introduce a framework to model *online voter control in sequential elections*. Our approach is inspired by the area of "online algorithms" [2], algorithms running and performing computational actions based only on the input data seen thus far.

In our framework of online voter control, the chair's task stated above is based on a "maxi-min" idea (although here, due to the time effects, that can involve more than two quantifiers), a typical online-algorithmic theme; in that framing of the chair's task we are following the approach used for online manipulation of [10]. Note that another central online-algorithmic theme, a strictly numerical ratio approach to so-called "competitive analysis," does not apply here, because voting is commonly based on an ordinal notion of preferences, which makes competitive ratios inappropriate in our setting. Sequential (or otherwise "dynamic") voting has been studied in other contexts as well, e.g., from a game-theoretic perspective as "Stackelberg voting games" [15] (see also [3]), or using an axiomatic approach [14] or Markov decision processes [13]. None of this work has considered the issue of voter control or has applied methods of online algorithms.

[1] Dept. of Computer Science, Rochester Institute of Technology, Rochester, NY 14623, USA, url: www.cs.rit.edu/∼ eh.
[2] Dept. of Computer Science, University of Rochester, Rochester, NY 14627, USA, url: www.cs.rochester.edu/∼ lane.
[3] Inst. für Informatik, Heinrich-Heine-Univ. Düsseldorf, 40225 Düsseldorf, Germany, email: rothe@cs.uni-duesseldorf.de.

2 Motivation

The coming sections will give our definitions, results, and some proofs. However, before that, the present section will very informally present some motivation and examples. In particular, we give example settings in which it is natural to study sequential action, in which the election's "chair" has a use-it-or-lose-it ability to do addition/deletion/partition-choice for each voter as the voter votes, and the chair knows the votes of the voters seen so far, but not of future voters. Of course, theoretical models don't capture the many interactions and subtleties of the real world, and so our models don't perfectly capture the full richness of even these sample situations. Nonetheless, we feel that for many cases, such as those we are about to mention, the theoretical models we develop in this paper are far closer to capturing the real-world situation than are existing models of simultaneous voting or even existing models where votes are sequential but all voters' preferences are known ahead of time.

As a concrete example (and let us for the moment not worry about what the particular election system is), consider a College faculty meeting at which, going right around the room, the faculty members hand their handwritten paper ballots to the Dean, who then passes them on to her administrative assistant, who quietly adds them to the totals he is keeping. But let us further assume that the Dean is a shifty person, and can, for a certain number of ballots, slip the piece of paper into her pocket after reading the vote, without that being noticed, and without the people in the room being likely to notice that there aren't quite enough votes in the totals (let's suppose it is a big college). And the question is whether the Dean can ensure that one of a few alternatives she favors can be made a winner of the election, using at most the given amount of vote-to-pocket shifting. This setting loosely corresponds to our sequential version of control by deleting voters. For vividness, our examples are about humans voting and a human chair (in the above, the Dean), and in the case just given, paper ballots. However, our model applies also to more electronically focused cases of preference aggregation, such as situations where the chair is a doctored voting machine, or perhaps is a web site and the voters are automated agents.

The above example is about deleting voters, but there are also natural examples for adding or partitioning voters. For partitioning voters, imagine that a school's undergraduate admissions office is going to use a panel, whose members will each be assigned to one of two faculty committees, to vet applying students (perhaps with the committees purportedly looking for different things, e.g., one is looking for traditional smartness and the other is looking for unusual levels of passion and creativity), with all applicants' folders given to both committees, and with each committee by voting selecting its favorite proposals, and then only the winners of those two vetting elections move on to a final election in which all the panel members vote. Suppose a particular admissions office staff member (who is the chair in this example), with all the faculty members lined up and coming into the room, as each faculty member steps to the doorway briefly chats with the faculty member well enough to discern the likely votes he or she will cast, and then right there assigns the person to either the smartness committee or the passion committee. If the admissions staff member does so with the goal of ensuring that at least one of a certain set of students (perhaps the students who are great football quarterbacks, or the students whose parents might fund a new admissions building) will be admitted, that very loosely put would be captured by our sequential version of control by partition of voters. For adding voters, a natural model might be a political candidate (who is the chair in this example) going door to door through her district in a preset order, and knowing from public records which voters are registered voters and which are not, and at each door meeting and learning the voter's preferences among the candidates, and then for those voters who are not registered deciding whether to use charisma to convince them to register and vote, with the limitation that the candidate has only so much charisma to use.

The above are a few very informal examples of settings where sequential action is natural, and one knows the votes cast so far but not those to be cast in the future (except who will be casting them and in which order). Let us finish this informal section by briefly giving a mini-example of the flavor of the goal we have for our chairs, and how that affects their actions. We are assuming that chairs are very pessimistic; what they want to know is whether there is some action they can take at the given moment so that one of their preferred candidates will win no matter what the value is of all the currently unknown-to-them future votes—but assuming that their own future decisions are similarly aimed at this same goal. To make this more concrete, let us discuss the most important real-world election system: plurality. In our addition-of-voters example above, suppose the candidate going door to door has only one preferred candidate in the election, namely, herself. Then it is quite clear and simple what she should do. Until she runs out of charisma, she should for each unregistered voter she meets for whom she is the favorite candidate expend her charisma to have that person become a registered voter. That is an "operational" approach that would work perfectly. But more must be said. The question our pessimistic candidate (and our decision problems) want answered at each point is whether, whatever the preferences still to come after the current point are, that candidate will win. And it is also clear how to judge that. The candidate, as she starts speaking with a given unregistered voter who likes our candidate the most (we can similarly describe how to reason in the other cases), reasons as follows: I need to assume that all future voters (whose preferences I don't currently know!) concentrate their votes on a candidate other than me who currently has the most votes (in the tally I have been building in my canvas so far), and that I use my charisma to (if it is not expended) add the current unregistered voter, and then I suppress those hypothetical against-me voters, and would that leave me a winner of this election? If the answer is yes, then the candidate should be very happy, as she knows she can guarantee herself victory as long as she doesn't later do anything overtly stupid with her charisma. The example we just gave is in effect explaining why it holds that (so-called constructive) control by adding voters is in polynomial time for sequential plurality elections. Now, one might assume that plurality is such a simple system that for all types of sequential control we will obtain polynomial-time algorithms. However, as Theorem 9 we will show that that is not the case (unless P = NP). The proof of that result is in effect giving an example—although admittedly a more complex one—in which a coNP-hard problem, namely the complement of Hitting Set, is transformed into an election control instance about sequentially partitioning voters (the control setting we described above in our example about college admissions).

3 Preliminaries

We assume familiarity with complexity-theoretic notions such as the complexity classes P, NP, coNP, and PSPACE, the polynomial-time many-one reducibility ($\leq_m^p$), and with $\leq_m^p$-hardness and $\leq_m^p$-completeness for a complexity class. A standard NP-complete problem is the satisfiability problem (SAT) from propositional logic, a standard coNP-complete problem is the tautology problem, and the quantified boolean formula problem (QBF) is a standard PSPACE-complete problem.

Voter Control Types in Simultaneous Elections

A pair (C,V) is called a *(standard or simultaneous) election* if C is a set of candidates and V a list of voters that all have cast their votes simultaneously. We assume that each vote in V has the form (v,p), where v is the name of this voter and p is v's (total) preference order over C. For example, if $C=\{c,d,e\}$ then $(\text{Bob}, d>e>c)\in V$ indicates that Bob (strictly) prefers d to e and e to c.

Bartholdi et al. [1] introduced the *standard types of (constructive) voter control in simultaneous elections* as follows. An election system is a mapping from votes/candidates to a winner set. Let $\mathscr{E}$ be a given election system. In *control by deleting voters* ($\mathscr{E}$-CCDV), given an election (C,V), a distinguished candidate $c\in C$, and a nonnegative integer $k\leq\|V\|$, we ask whether we can delete at most k voters from V such that c is an $\mathscr{E}$ winner of the resulting election. In *control by adding voters* ($\mathscr{E}$-CCAV), we are given a candidate set C, a list V of registered voters with preferences over C, a list V' of as yet unregistered voters with preferences over C, a distinguished candidate $c\in C$, and a nonnegative integer $k\leq\|V'\|$, and the question is whether we can add to V at most k voters from V' such that c is an $\mathscr{E}$ winner of the resulting election. Finally, in *control by partition of voters*, we are given an election (C,V) and a distinguished candidate $c\in C$, and we ask whether V can be partitioned into two sublists, V_1 and V_2, such that c is an $\mathscr{E}$ winner of the election $(W_1\cup W_2,V)$, where W_i for $i\in\{1,2\}$ is the (possibly empty) set of winners of subelection (C,V_i) that have survived the tie-handling rule used. Of the two tie-handling models introduced by Hemaspaandra et al. [8] we focus on the *ties-promote (TP)* model only, where all winners of a subelection proceed to the runoff, as that model fits more naturally the nonunique-winner model in which we will define our online control problems. The resulting problem is denoted by $\mathscr{E}$-CCPV.

The destructive variants of these three problems, denoted by $\mathscr{E}$-DCDV, $\mathscr{E}$-DCAV, and $\mathscr{E}$-DCPV, are obtained by requiring that the distinguished candidate c is *not* a winner of the election resulting from the control action at hand [8].

Online Voter Control in Sequential Elections

We study *online voter control in sequential elections*, where we assume that the voters vote in order, one after the other, each expressing preferences over all the candidates. If u is the current voter and C the given candidate set, an *election snapshot for C and u* is specified by a triple $V = (V_{<u}, u, V_{u<})$, where the voters in $V_{<u}$ have already cast their votes, each a preference order over C, now it is u's turn to cast a vote, and the future voters in $V_{u<}$ will cast their votes in the order listed. ($V_{<u}$ and u of course list the votes cast and who cast them, but $V_{u<}$ just gives the order of the voters following u.)

We now define our notions of online voter control for the standard voter control types stated above, and the related problems. They all will start from a basic *online voter control setting* (an *OVCS*, for short), augmented by appropriate additional information according to the control type at hand. A basic OVCS (C, u, V, σ, d) consists of a set C of candidates, the current voter u (which isn't strictly needed here, as u is clearly singled out within V anyway), an election snapshot for C and u, the chair's preference order σ on C, and a distinguished candidate $d \in C$. Let $\mathscr{E}$ be a given election system and let $W_{\mathscr{E}}(C,V)$ denote the $\mathscr{E}$ winner set of (standard) election (C,V). For each online voter control type we will define, the question the chair faces is: Is it possible to decide whether or not to exert the action of this control type to the current voter u (e.g., whether or not to delete u) such that, no matter what votes the remaining voters after u wish to cast, the chair's goal can be reached by the current decision regarding u and by using the chair's future decisions of this type (if any), each being made using the chair's then-in-hand knowledge about what votes will have been cast by then?[4] By *the chair's goal* we mean either to ensure $W_{\mathscr{E}}(C,V') \cap \{c \,|\, c \geq_\sigma d\} \neq \emptyset$ for each possible ultimate election (C,V') (i.e., each V' is a possible vote list resulting from the control type at hand after all decisions have been made by the chair and all voters have cast their votes) in the constructive case, or to ensure that $W_{\mathscr{E}}(C,V') \cap \{c \,|\, d \geq_\sigma c\} = \emptyset$ in the destructive case.[5] Note, in particular, that due to the conditions $W_{\mathscr{E}}(C,V') \cap \{c \,|\, c \geq_\sigma d\} \neq \emptyset$ and $W_{\mathscr{E}}(C,V') \cap \{c \,|\, d \geq_\sigma c\} = \emptyset$ that define the chair's goal, we adopt the *nonunique-winner* model in defining our problems. So, to formally define our problems, it remains to specify for each control type the information by which the basic OVCS is augmented. What kind of decisions the chair is to make in the course of a sequential election will always be clear from the control type at hand (e.g., whether or not to delete a voter in "online control by deleting voters").

Let $B = (C, u, V, \sigma, d)$ be a given basic OVCS. For *online control by deleting voters*, B is augmented by the following additional information: A nonnegative integer k (the deletion upper bound); for each voter v before u, there is a flag saying if v was deleted and the vote cast by v (if not deleted)—at most k voters can be marked as deleted for the input to be syntactically legal; a vote to cast for the current voter u (if u is not to be deleted). We denote these problems by online-$\mathscr{E}$-CCDV (constructive) and online-$\mathscr{E}$-DCDV (destructive). (We certainly could equivalently formulate the problem in a way that masks out all earlier deleted voters, and so removes the need for the flagging; but we prefer the above version as it allows the actual history of the voting situation to be part of the instance.)

For *online control by adding voters*, B is augmented by the following additional information: A nonnegative integer k (the addition upper bound); each voter v in V has a flag saying if v is unregistered (i.e., can be added) or registered—u must be unregistered; each voter v before u has another flag saying if v was added—at most k voters have that flag set in any syntactically legal input; the vote cast is given for each registered or added unregistered voter before u, and also u's potential vote (if u is to be added). We denote these problems by online-$\mathscr{E}$-CCAV (constructive) and online-$\mathscr{E}$-DCAV (destructive).

For *online control by partition of voters*, B is augmented by the following additional information: Each voter v before u has a flag saying which part of the partition v was assigned to ("left" or "right"), and the vote cast by v and also u's vote is given. We denote these problems by online-$\mathscr{E}$-CCPV (constructive) and online-$\mathscr{E}$-DCPV (destructive), as we focus on the ties-promote (TP) rule. This is the right choice for the nonunique-winner model, which itself is here more natural than the unique-winner model, since our online control problems are defined via *upper-cuts* ("make d or a better candidate win," in the constructive case) or *lower-cuts* ("make sure that neither d nor a worse candidate wins," in the destructive case).

A natural worry about our maxi-min approach to online voter control is that it is always possible that all the future voters are hostile to one's goals. And in that case, one may be, depending on the election system, powerless to reach one's goal in the worst case, and so the maxi-min outcome is easily seen to be failure to reach one's goal. Although this worry exists in a weaker form for online manipulation and online bribery, since for those if one is allowed almost no vote-changing one is in many cases obviously in trouble, at least in those settings one can do whatever one wants to those votes one does manipulate or bribe. In control, one doesn't get to set the value of a single vote, and that is pretty extreme.

This is a valid worry, but there are some things that keep it in perspective. Primarily, our paper is trying to find out the very greatest complexity that control can ever have (when restricted to election systems having polynomial-time winner problems). And so we can look at election systems that sidestep the above worry, due to their properties simply not matching the intuition above, which is that we are using an election system in which a lot of bad-for-us votes result in a bad-for-us output. In effect, we are seeking to understand the limits of behavior, in order to set a bounding box on the behaviors that can be realized. Of course, for many natural election systems, the effect mentioned in the previous paragraph will hold, and for many inputs that fact can be exploited to help achieve polynomial-time algorithms for the control problem; indeed, in this paper itself, we give examples of achieving polynomial-time algorithms for a natural system: plurality. Of course, problems may start with some votes already cast, and this may itself potentially make for interesting "endgame" decision issues, as may issues involving weighted votes. Also, we very much hope further studies will be conducted (by us or by others) employing a range of models, including ones beyond maxi-min.

Due to space limitations here, most proofs are omitted, including the rather difficult and novel proof of Theorem 4. *Omitted proofs can in general be found in our full technical report version [12].*

[4] Note that this maxi-min-inspired (but with more quantifiers) approach is really about alternating quantifiers. We are asking if there exists a current action of the chair, such that for all potential revealed vote values that come between now and the next time the chair has to decide on an action, there exists a next action by the chair, such that for all … … the chair reaches her goal.

[5] Why do we provide an ordering σ rather than just providing as a list the set of candidates who are good enough to count as reaching our goal? For the decision-problem version of online manipulation, which is our formulation here, providing such a set would be just as good. But by making σ a part of the input, we make the model compatible, for the future, with the interesting optimization problem of trying to find the most preferred candidate within σ for which the chair can ensure that there is among the winner set one of the candidates in the segment from that candidate to the top candidate in σ.

4 General Upper and Lower Bounds

Theorem 1 *For each election system $\mathscr{E}$ with a polynomial-time winner problem,*[6] online-$\mathscr{E}$-CCDV*,* online-$\mathscr{E}$-DCDV*,* online-$\mathscr{E}$-CCAV*,* online-$\mathscr{E}$-DCAV*,* online-$\mathscr{E}$-CCPV*, and* online-$\mathscr{E}$-DCPV *are each in* PSPACE*.*

Theorem 1 settles all general (i.e., regarding any voting system for which winner determination is easy) upper bounds for our online voter control problems. We now turn to their lower bounds.

4.1 Control by Deleting and by Adding Voters

Theorem 2 *There exist election systems $\mathscr{E}$ and $\mathscr{E}'$ with polynomial-time winner problems such that* online-$\mathscr{E}$-CCDV*,* online-$\mathscr{E}$-CCAV*,* online-$\mathscr{E}'$-DCDV*, and* online-$\mathscr{E}'$-DCAV *are* PSPACE*-complete, even if limited to only two candidates.*

PROOF. Let (C,V) be an election. We define election system $\mathscr{E}$ as follows. $\mathscr{E}$ interprets—in some fixed, natural encoding—the lexicographically least candidate name in C as a boolean formula, Φ, whose variable names must be the strings $x_1,x_2,\ldots,x_{2\ell}$ for some ℓ, where $x_{2\ell}$ actually appears in Φ (the other variables don't have to; no variables other than $x_1,x_2,\ldots,x_{2\ell}$ are allowed). If these syntactic requirements fail to hold, everyone loses in $\mathscr{E}$. Otherwise, if any two voters in V have the same name, everyone loses in $\mathscr{E}$. Otherwise, order the voters in V lexicographically by name of the voter, and let $v_1,v_2,\ldots,v_z$ be the voter names in this order. If $z<2\ell$ or if there are less than two candidates, everyone loses in $\mathscr{E}$. Otherwise, if for some odd i, $1\le i\le 2\ell-1$, the two lowest order bits of v_i are not 00 or 01, or if for some even i, $2\le i\le 2\ell$, the two lowest order bits of v_i are not 10 or 11, everyone loses in $\mathscr{E}$. Otherwise, assign the variables of $\Phi(x_1,x_2,\ldots,x_{2\ell})$ as follows. For each odd i, $1\le i\le 2\ell-1$, set x_i to *true* if the two lowest order bits of v_i are 01, and set x_i to *false* otherwise (i.e., the two lowest order bits of v_i are 00). For each even i, $2\le i\le 2\ell$, set x_i to *true* if the name of the least preferred candidate in the vote of v_i is lexicographically less than the name of the next to least preferred candidate in the vote of v_i, and set x_i to *false* otherwise. If this assignment satisfies Φ, everyone wins in $\mathscr{E}$, and otherwise everyone loses. This ends the specification of $\mathscr{E}$. Since a boolean formula whose variables have all been assigned can be evaluated in polynomial time, $\mathscr{E}$ has a polynomial-time winner problem.

By Theorem 1, online-$\mathscr{E}$-CCDV is in PSPACE. To show PSPACE-hardness of online-$\mathscr{E}$-CCDV, we $\le_m^p$-reduce the PSPACE-complete problem QBF′, a variant of QBF, to it. QBF′ is the set of boolean formulas of the form $F(x_1,x_2,\ldots,x_{2\ell})$, for some ℓ, such that the variable $x_{2\ell}$ appears in F, all variables appearing in F are from the variable name collection "x_1", "x_2", …, "$x_{2\ell}$", and $(\exists b_1)(\forall b_2)\cdots(\exists b_{2\ell-1})(\forall b_{2\ell})[F(x_1:=b_1,x_2:=b_2,\ldots,x_{2\ell}:=b_{2\ell})$ evaluates to *true*], where $b_i\in\{0,1\}$ and $x_i:=b_i$ means that variable x_i is set to *true* if $b_i=1$, and is set to *false* if $b_i=0$, for $1\le i\le 2\ell$.

Let $F(x_1,x_2,\ldots,x_{2\ell})$ be a given instance of QBF′, where $x_{2\ell}$ explicitly appears in F. (If our input is syntactically incorrect, we map it to a fixed no-instance of online-$\mathscr{E}$-CCDV.) We construct from F an instance of online-$\mathscr{E}$-CCDV, consisting of a basic OVCS (C,u,V,σ,d), augmented by the additional information of online control by deleting voters, as follows. Define $C=\{a,b\}$, where a encodes F (in our fixed, natural encoding of boolean formulas) and b is the string lexicographically immediately following a; the current voter is $u=v_1$; V will be specified below; the chair's preference order is $a>_\sigma b$; for specificity, we let $d=a$ be the distinguished candidate (though that does not matter, as all candidates win or all lose in $\mathscr{E}$); the deletion limit is $k=\ell$; and a vote $a>b$ to cast for u if not deleted (again, the vote doesn't matter, as $u=v_1$ will specify an assignment to x_1 by her name, not by her vote). There are $(3/2)\cdot 2\ell=3\ell$ voters in V such that the name of the ith voter, v_i, is the binary string u_iw_i, where u_i is the binary representation of i and $w_i=00$ if $i\equiv 1 \bmod 3$, $w_i=01$ if $i\equiv 2 \bmod 3$, and $w_i=10$ if $i\equiv 0 \bmod 3$, $1\le i\le 3\ell$. This completes the description of our $\le_m^p$-reduction from QBF′ to online-$\mathscr{E}$-CCDV, which clearly can be computed in polynomial time.

We claim that $F\in$ QBF′ if and only if the chair's goal can be reached by at most k deletions of voters. Why? By the definition of $\mathscr{E}$, everyone loses unless our $k=\ell$ deletions are used on *exactly one of* v_{3i-2} *and* v_{3i-1}, for each i, $1\le i\le\ell$. No v_{3i}, $1\le i\le\ell$, can be deleted if there is to be a winner. And the "exactly one of v_{3i-2} and v_{3i-1}" choices, $1\le i\le\ell$, specify an assignment of truth values to the odd-numbered variables: For each i, $1\le i\le\ell$, x_{2i-1} is set to *true* if v_{3i-2} is deleted and v_{3i-1} is not, and is set to *false* if v_{3i-1} is deleted and v_{3i-2} is not. On the other hand, for each i, $1\le i\le\ell$, the truth value of x_{2i} is specified by the *vote* of voter v_{3i}, since after these ℓ deletions, v_{3i} will be the $2i$th voter name in the lexicographic order. It follows that the chair's goal can be reached by at most k deletions of voters if and only if $(\exists b_1)(\forall b_2)\cdots(\exists b_{2\ell-1})(\forall b_{2\ell})[F(x_1:=b_1,x_2:=b_2,\ldots,x_{2\ell}:=b_{2\ell})$ evaluates to *true*], which is true if and only if $F\in$ QBF′.

PSPACE-hardness of online-$\mathscr{E}$-CCAV for the election system $\mathscr{E}$ defined above can be shown via essentially the same $\le_m^p$-reduction from QBF′. The only difference is that we now map the given QBF′ instance F to an instance of online-$\mathscr{E}$-CCAV, which is defined exactly as the online-$\mathscr{E}$-CCDV instance constructed above, except that all voters v_i with $i\equiv 0 \bmod 3$ are specified as registered voters, and all other voters are unregistered. The correctness argument is analogous.

The destructive cases can be shown analogously, by modifying the election system $\mathscr{E}$ defined above as follows, yielding our modified system $\mathscr{E}'$: Whenever everyone loses (wins) in $\mathscr{E}$, everyone wins (loses) in $\mathscr{E}'$. It follows from Theorem 1 and the above $\le_m^p$-reduction from QBF′ that online-$\mathscr{E}'$-DCDV and online-$\mathscr{E}'$-DCAV are both PSPACE-complete. ❑

For control by deleting or adding voters, the deletion or addition limit k is part of the problem instance. Now, we consider restrictions of these problems in which this limit is bounded by a constant. For a given election system $\mathscr{E}$ and a fixed k, let online-$\mathscr{E}$-CCDV$[k]$ be the restriction of online-$\mathscr{E}$-CCDV to those inputs whose deletion limit is at most k, and define the problem variant online-$\mathscr{E}$-CCAV$[k]$ analogously. We now show that this change in the definition brings the complexity of these problems from PSPACE down to coNP.

Theorem 3 *For each $k\ge 0$, the following hold:*

1. *For each election system $\mathscr{E}$ with a polynomial-time winner problem,* online-$\mathscr{E}$-CCDV$[k]$ *and* online-$\mathscr{E}$-CCAV$[k]$ *are in* coNP*.*
2. *There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that* online-$\mathscr{E}$-CCDV$[k]$ *and* online-$\mathscr{E}$-CCAV$[k]$ *are* coNP*-complete, even if limited to only two candidates.*

4.2 Control by Partition of Voters

Theorem 4 *There exist election systems $\mathscr{E}$ and $\mathscr{E}'$, whose winner problems can be solved in polynomial time, such that*

[6] The statement of Theorem 1 holds even for election systems whose winner problems are in PSPACE.

online-$\mathscr{E}$-CCPV *and* online-$\mathscr{E}'$-DCPV *are* PSPACE-*complete, even if limited to only two candidates.*

The above result establishes PSPACE-completeness for constructive and destructive online control by partition of voters in election systems with polynomial-time winner problems, even if there are at most two candidates. Can we make do with one? The following result shows that if we could, then PSPACE would equal NP $\cap$ coNP.[7]

Theorem 5 *1. For each election system $\mathscr{E}$ with a polynomial-time winner problem,* online-$\mathscr{E}$-CCPV *and* online-$\mathscr{E}$-DCPV*, when inputs are restricted to at most one candidate, are both in* NP.

2. *There exist election systems $\mathscr{E}$ and $\mathscr{E}'$ with polynomial-time winner problems such that* online-$\mathscr{E}$-CCPV *and* online-$\mathscr{E}'$-DCPV*, even when restricted to one candidate, are both* NP*-complete.*

Corollary 6 *The following three statements are equivalent:*

1. PSPACE = NP $\cap$ coNP.
2. *There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that* online-$\mathscr{E}$-DCPV *is* PSPACE*-complete even when restricted to one candidate.*
3. *There exists an election system $\mathscr{E}$ with a polynomial-time winner problem such that* online-$\mathscr{E}$-CCPV *is* PSPACE*-complete even when restricted to one candidate.*

The analogues of the destructive cases of both parts of Theorem 5 also hold when "online" is removed, i.e., for the problem $\mathscr{E}$-DCPV. In contrast, the *constructive non-online* analogue of Theorem 5's first part can be strengthened to a P upper bound. (Why can we get a P result here but not in Theorem 5? Our proof (available in our full version) that establishes the following result does not apply if some voters are already committed to sides of the partition—it is assuming (and truly using the fact) that we have full control of where *all* voters go. But in the online setting, the current voter u can be a voter who does *not* come first and so some voters may already be assigned to sides of the partition. And why do we get P for constructive but not destructive? The effect the proof (available in our full version) of the following theorem uses is specific to the constructive case.)

Theorem 7 *For each election system $\mathscr{E}$ with a polynomial-time winner problem, $\mathscr{E}$-*CCPV*, when inputs are restricted to at most one candidate, is in* P.

5 Online Control for Plurality

We have seen in the previous section that online control can be a very hard, namely a PSPACE-complete, problem, even for voting systems whose winners can be determined in polynomial time. In this section, we study online control for plurality voting. In this very simple, yet popular voting system, every voter gives one point to her most preferred candidate, and all candidates with the most points win. It is known that non-online control by adding and by deleting voters can be done in polynomial time, both in the constructive [1] and in the destructive [8] case. We now show that the corresponding types of online control are also easy.

Theorem 8 online-plurality-CCDV*,* online-plurality-CCAV*,* online-plurality-DCDV*, and* online-plurality-DCAV *are each in* P.

PROOF. For online-plurality-CCDV, let (C,u,V,σ,d) be a given basic OVCS, augmented by the additional information of online control by deleting voters: a deletion upper bound k, for each voter v before u a flag saying if v was deleted and the vote cast by v (if not deleted), where at most k voters can be marked as deleted, and a vote to cast for u (if u is not to be deleted). If d is the chair's bottom choice in σ, we are done, since the input then is trivially in online-plurality-CCDV (unless it is syntactically illegal). If exactly k voters have been marked as already deleted, we can do no more deletions, so u and all later voters go in, and we assume (as this is the most challenging case) that all later voters vote for one particular candidate in $\Lambda_d = \{c \in C \,|\, c <_\sigma d\}$ that among the candidates in Λ_d has the most first place votes after u is put in, and so we can easily answer the online control question. If less than k voters have been selected already for deletion, then delete u if and only if u's top choice is a highest scoring (with respect to the voters before u) candidate in $\{c \in C \,|\, c <_\sigma d\}$. Then assume that all later voters vote for one particular candidate in $\Lambda_d = \{c \in C \,|\, c <_\sigma d\}$ that among the candidates in Λ_d has the most first place votes after u is put in. And assume we delete as many of those as the deletion amount left (after u) allows. It is easy to see whether this results in "d or better" being a winner (in which case our algorithm answers "yes") or not (in which case our algorithm answers "no"). (One might comment that it would suffice, especially to just handle the decision version, to follow the very simple "operational" approach mentioned in Section 2. However, we have given a more dynamic description of the process both as we want to make clear how the chair can decide what action to take at each point and as the description above is also helping establish the correctness of the actions taken.)

For online-plurality-CCAV, let (C,u,V,σ,d) be a given basic OVCS, augmented by the additional information of online control by adding voters: an addition upper bound k, for each voter the information of whether she is registered or not, and for each unregistered voter before u the information of whether she has been added or not, the vote of each registered or added voter before u, and u's potential vote. Again, the question is trivial if d is the chair's bottom choice in σ. Otherwise, we can see what u's vote is and if k has yet been reached. If k has not been reached yet, we add u if and only if u's top choice belongs to $\{c \in C \,|\, c \geq_\sigma d\}$.[8] And in the worst case all future voters vote for the same member of $\{c \in C \,|\, c <_\sigma d\}$, which will be one that after u votes has the most first-place votes among those.

The two destructive cases can be handled analogously. The main differences are, in both cases, that the question now is trivial to decide if d is the chair's *top* choice in σ; in the deleting voters case, that u is to be deleted (provided the deletion limit k has not been reached yet) if and only if u's top choice is a highest scoring (with respect to the voters before u) candidate in $\{c \in C \,|\, c \leq_\sigma d\}$; and in the adding voters case, that u is to be added (provided the addition limit k has not been reached yet) if and only if u's top choice belongs to $\{c \in C \,|\, c >_\sigma d\}$. And, in both cases, we again assume that all future votes will belong to some particular member of $\{c \in C \,|\, c \leq_\sigma d\}$ that after u votes has the most first-place votes among those candidates. ❑

Non-online control by partition of voters, in the model (ties-promote) we feel is most natural and have adopted in this paper, is known to be NP-complete [8] in both the constructive and destructive cases. In contrast, the corresponding types of online control are

[7] Are elections with just one candidate even ever interesting in the real world? We feel they are. For example, a referendum is essentially an up-or-down vote on one "candidate." So is a vote on whether to recall an elected official, or to impeach a judge, or ratify a person who has been nominated for a sports hall of fame.

[8] Sure enough, u's top choice could be one of those candidates that end up having only few votes, so adding u could be a wasted addition that will block some future good addition in some vote sequences, but in the worst case all future voters put first a candidate disliked by the chair; so our action is fine within the quantifier structure of the problem.

both coNP-hard. This implies that these problems cannot be in NP, unless NP = coNP, which is considered to be highly unlikely. It remains open whether or not they are in coNP; we conjecture that they are not.

Theorem 9 online-plurality-CCPV *and* online-plurality-DCPV *are both* coNP-*hard.*

PROOF. We prove this by a reduction from the complement of the NP-complete problem Hitting Set: Given a set $B = \{b_1, \ldots, b_m\}$, a nonempty collection $\mathscr{S} = \{S_1, \ldots, S_n\}$ of subsets of B, and a positive integer $k \leq m$, does $\mathscr{S}$ have a hitting set of size at most k, i.e., does there exist a set $B' \subseteq B$ such that $\|B'\| \leq k$ and for all $S_i \in \mathscr{S}$, $S_i \cap B' \neq \emptyset$.

We turn an instance $(B, \mathscr{S}, k)$ of hitting set into the following instance of online partition of voters. The set of candidates is $\{c, w, b_1, \ldots, b_m\} \cup A$, where $A = \{a_i \mid 1 \leq i \leq 4mnk+1\}$. The current voter is u. The votes before u that are on the left side of the partition are exactly the same as the votes before u that are on the right side of the partition. Both sides of the partition consists of the following votes.

- $4nk$ votes $c > w > \cdots$, where $\cdots$ denotes that the remaining candidates follow in some arbitrary order.
- $4nk$ votes $w > c > \cdots$.
- For every i, $1 \leq i \leq n$, $2k$ votes $S_i > c > \cdots$, where S_i denotes the candidates in S_i in some arbitrary order.
- For every j, $1 \leq j \leq m$, as many votes $b_j > B - \{b_j\} > c > w > \cdots$ as needed to make the score of b_j equal to $4nk-1$ in this subelection.
- For every i, $1 \leq i \leq 4mnk$, one vote $a_i > c > \cdots$ and one vote $a_i > w > \cdots$.

Voter u votes $a_{4mnk+1} > w > \cdots$. And there are k voters after u. The chair's top choice is c and the chair's bottom choice is w, and the distinguished candidate is c in the constructive case (i.e., for online-plurality-CCPV) and w in the destructive case (i.e., for online-plurality-DCPV). A simple but crucial observation is that no candidate $a \in A$ will ever make it to the final round, since her score in the first round will be at most $2+k$. If c and w participate in the final round, c gets $8mnk$ points and w gets $8mnk+1$ points from voters whose top choice was in A. This will ensure that c and w are the only possible winners in the final round.

We will show that $\mathscr{S}$ does not have a hitting set of size at most k if and only if c can always be made a winner in the constructed election, and we will show that $\mathscr{S}$ does not have a hitting set of size k if and only if w can always be made to not be a winner in the constructed election. This will prove the theorem.

First suppose that $\mathscr{S}$ has a hitting set of size at most k. Let B' be a hitting set of size k. B' exists, since $k \leq m$. Let the k voters after u vote such that the top choice of the ith voter is the ith candidate in B'. Then, no matter how we partition the voters, the set of candidates that participate in the final round is $\{c, w\} \cup B'$. The scores in the final round are as follows: (a) $score(c) = 8nk + 8mnk$, (b) $score(w) = 8nk + 8mnk + 1$, and (c) for all $b \in B'$, $score(b) \leq 8mnk - m + k$. It follows that c is not a winner and that w is a winner.

For the converse, note that we can always make sure that the set of candidates in the final round is of the form $\{c, w\} \cup B'$, where $B' \subseteq B$ and $\|B'\| \leq k$, by putting all the voters after u in the same first-round election. If there is no hitting set of size at most k, then B' is not a hitting set. It follows that in the final election the following hold: (a) $score(c) \geq 8nk + 8mnk + 4k$, (b) $score(w) \leq 8nk + 8mnk + 1 + k$, and (c) for all $b \in B'$, $score(b) \leq 8mnk - m + k$. It follows that c is the unique winner of this election. ❑

6 Conclusions and Open Questions

Inspired by the maxi-min approach of online algorithms, we studied online voter control in sequential voting. We showed that for suitably constructed election systems with polynomial-time winner problems, the resulting problems can be extremely hard, namely PSPACE-complete, even for only two candidates. For plurality, however, online control by deleting or adding voters is easy, just as in the non-online case. In future work we will study online voter control also for other natural election systems. Can one obtain PSPACE-completeness results for highly natural, existing systems, for example? Another interesting task will be to investigate online control by a typical-case analysis. We have also studied online candidate control in sequential voting [11] and have already started to investigate online bribery.

Acknowledgments We are deeply grateful to the reviewers. This work was supported in part by grants NSF CCF-{0426761,0915792, 1101452,1101479}; DFG RO-1202/15-1; ARC DP110101792; an SFF grant from HHU; a DAAD PPP/PROCOPE grant; and AvH Foundation Bessel Awards to E. & L. Hemaspaandra.

REFERENCES

[1] J. Bartholdi, III, C. Tovey, and M. Trick, 'How hard is it to control an election?', *Math. and Computer Modeling*, **16**(8/9), 27–40, (1992).
[2] A. Borodin and R. El-Yaniv, *Online Computation and Competitive Analysis*, Cambridge University Press, 1998.
[3] Y. Desmedt and E. Elkind, 'Equilibria of plurality voting with abstentions', in *Proc. ACM-EC'10*, pp. 347–356. ACM Press, (2010).
[4] G. Erdélyi, M. Nowak, and J. Rothe, 'Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control', *Math. Logic Quarterly*, **55**(4), 425–443, (2009).
[5] G. Erdélyi, L. Piras, and J. Rothe, 'The complexity of voter partition in Bucklin and fallback voting: Solving three open problems', in *Proc. AAMAS'11*, pp. 837–844. IFAAMAS, (2011).
[6] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra, 'Using complexity to protect elections', *Comm. of the ACM*, **53**(11), 74–82, (2010).
[7] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'Llull and Copeland voting computationally resist bribery and constructive control', *JAIR*, **35**, 275–341, (2009).
[8] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'Anyone but him: The complexity of precluding an alternative', *Artificial Intelligence*, **171**(5–6), 255–285, (2007).
[9] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'Hybrid elections broaden complexity-theoretic resistance to control', *Math. Logic Quarterly*, **55**(4), 397–424, (2009).
[10] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'The complexity of online manipulation of sequential elections', Technical Report arXiv:1202.6655 [cs.GT], Computing Research Repository, arXiv.org/corr/, (February 2012).
[11] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'Controlling candidate-sequential elections', in *Proc. ECAI'12*. IOS Press, (2012). To appear.
[12] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'Online voter control of sequential elections', Technical Report arXiv:1203.0411 [cs.GT], Computing Research Repository, arXiv.org/corr/, (March 2012).
[13] D. Parkes and A. Procaccia, 'Dynamic social choice: Foundations and algorithms'. Working paper, May 2011.
[14] M. Tennenholtz, 'Transitive voting', in *Proc. ACM-EC'04*, pp. 230–231. ACM Press, (2004).
[15] L. Xia and V. Conitzer, 'Stackelberg voting games: Computational aspects and paradoxes', in *Proc. AAAI'10*, pp. 697–702. AAAI Press, (2010).