Auton Agent Multi-Agent Syst (2017) 31:606–627
DOI 10.1007/s10458-016-9339-3

How hard is control in single-crossing elections?

Krzysztof Magiera[1] · **Piotr Faliszewski**[1]

Published online: 21 June 2016

Abstract Election control problems model situations where some entity (traditionally called the election chair) wants to ensure some candidate's victory by either adding or deleting candidates or voters. The complexity of deciding if such control actions can be successful is well-studied for many typical voting rules and, usually, such control problems are NP-complete. However, Faliszewski et al. (Inf Comput 209(2):89–107, 2011) have shown that many control problems become polynomial-time solvable when we consider single-peaked elections. In this paper we show that a similar phenomenon applies to the case of single-crossing elections. Specifically, we consider the complexity of control by adding/deleting candidates/voters under plurality, Condorcet, and approval voting. For each of these control types and each of the rules, we show that if the control type is NP-complete in general, it becomes polynomial-time solvable for single-crossing elections.

Keywords Elections · Control · Complexity · Single-crossing · Approval · Condorcet · Plurality

1 Introduction

The goal of this paper is to study the complexity of election control, for the case where voters' preferences are single-crossing. While in general, without any restrictions on voters' preferences, the complexity of control problems tends to be high, Faliszewski et al. [25] have shown that it can drop significantly if we assume preferences to be single-peaked. We complement their results by showing the same phenomenon under the single-crossing assumption.

A preliminary version of this paper was presented at the ECAI-2014 conference.

✉ Piotr Faliszewski
faliszew@agh.edu.pl

1 AGH University of Science and Technology, Kraków, Poland

On the intuitive level, election control problems model settings where some entity (traditionally referred to as the *election chair*) is interested in affecting the result of a given election by changing its structure. Typical cases include, e.g., control by deleting candidates or control by adding voters. The former may correspond, e.g., to a situation where an election organizer prevents some candidates from registering, whereas the latter may correspond, e.g., to campaigns aimed at convincing members of the society to cast their votes (of course, a political party that runs such a campaign is most interested in convincing the people that support the party).

Election control problems arise not only in standard political elections, but also (or, perhaps, mostly) in multiagent systems. Ephrati and Rosenschein [17] proposed voting-based solutions for multiagent planning problems, and there control by adding (deleting) candidates can correspond to a particular agent proposing (or arguing for dismissal of) particular variants of the constructed plan. Voting is a natural tool for recommendation systems [27,31], and there adding/deleting voters can correspond to a situation where an insincere user of a recommendation algorithm tries to manipulate its results by limiting/extending the available preference data. (For example, this insincere user may want to manipulate the results if he or she is trying to influence some other entity's business decisions and these decisions are based on the output of the recommendation system.) Finally, in the well-known scenario where one uses a voting-based meta-search engine for the Web [13] (i.e., an algorithm that treats other web search engines as voters and aggregates their results using voting mechanisms), control by adding/deleting candidates/voters corresponds to a particular selection of the web pages and search engines to use. (Again, the owner of such a meta-search engine might want to skew its results in some way.)

Election control typically corresponds to negative actions (though, we should stress that this is not always the case) and, so, we would like control to be as hard as possible.[1] Indeed, this idea was at the heart of the research line started by Bartholdi et al. [2], who introduced the study of the complexity of election control problems and who have shown that candidate control is NP-hard for the plurality rule and that voter control is hard for the Condorcet rule (we point the reader to Sect. 2 for detailed definitions). Bartholdi et al. focused on constructive control, where the manipulating entity tries to ensure some given candidate's victory. Later, Hemaspaandra et al. [28] extended the control model to include *destructive control*, where the goal is to prevent a particular candidate from winning.

1.1 Election control in restricted domains

We quickly review some research regarding the complexity of election control at the end of this section. However, briefly put, we have that "for most of the typical voting rules, most of the typical (constructive) control problems are NP-hard." (Note that we do not want to neglect the value of studying control for various voting rules and we think that such research is important; our point is that control problems tend to have high computational complexity.) In light of the previous discussion, we should take it as good news. However, this is so only on the surface. NP-hardness is a worst-case notion and many of the election-control NP-hardness proofs rely on building very contrived elections which are unlikely to appear in real life. Indeed, if in some political elections a voter thinks that a left-wing candidate is the

[1] One could even say that it would be ideal if control were not possible at all. This, however, is neither feasible nor desirable. After all, if sufficiently many voters with particular preferences joined the election, we *would* want them to be able to sway the result in their favor. Similarly, if a particularly desirable candidate joined the race, it *would* be natural that he or she should win. Indeed, for candidate control there even are results suggesting that candidate control is unavoidable in principle [12] (see also recent work on the topic [29,38]).

best one, then we would not expect this voter to consider an extreme right-wing candidate to be the second best. In real life, we would expect voters' preference orders to have a natural structure and we certainly would not expect to see all the possible orders.

Social choice theory offers several models that capture such more structured preferences of the voters. Among these models, the two most popular are the single-peaked assumption introduced by Black [3] and the single-crossing assumption, introduced by Mirrlees [34] and Roberts [39]. In the former, the idea is that there is some global order of the candidates (e.g., modeling the left-to-right political spectrum of opinions) and for each integer k and each voter v, v's k most preferred candidates form a consecutive block with respect to this global order. The idea of the single-crossing assumption is that we can order the voters in such a way that the following holds: If a and b are two candidates, then the voters that prefer a to b form a consecutive block on one end of the voter order, and the voters who prefer b to a form a consecutive block on the other end. (In political elections, one could say that the voters are ordered from the extreme-left one to the extreme-right one, and for each two candidates a and b, if a is more "leftist" than b, then the left-wing voters prefer a to b, whereas the right-wing voters prefer b to a.)

Both single-peaked and single-crossing assumptions are quite convincing (as far as idealized models go, of course). Indeed, as argued by a number of economists, single-crossing profiles arise naturally in many settings (with taxation issues being prominently represented; we point interested readers to the book of Persson and Tabellini [37] for several settings where single-crossing profiles arise). Single-peaked profiles have been studied for much longer and they arise in many natural settings as well (see, e.g., the classic work of Black [3], where single-peaked profiles were introduced). In effect, it is very interesting that Faliszewski et al. [25] and Brandt et al. [5] have shown that if elections are single-peaked, then control by adding/deleting candidates/voters is solvable in polynomial time for plurality, Condorcet, and approval rules (and, indeed, for many other voting rules as well). These results suggest that many of the known hardness proofs for control problems rely on building elections that are unlikely to occur in practice. This view was strengthened by Faliszewski et al. [23] who also considered nearly single-peaked elections. In this paper, we provide further evidence that hardness of election control comes from looking at contrived preference profiles by showing that all the control by adding/deleting candidates/voters problems for plurality, Condorcet, and approval are solvable in polynomial-time for single-crossing elections (see Table 1 for a summary).

1.2 Related work

The study of the complexity of control problems was initiated by Bartholdi et al. [2], who introduced the topic and defined the constructive variants of the problem. Their work was extended by Hemaspaandra et al. [28] to include the destructive cases. Since these two papers, many researchers studied the complexity of control for various voting rules [18,24,36], considered various different extensions of the control problems [22,33], and different formal approaches to their study [19,30] (the references here are just examples; we point the reader to the surveys of Faliszewski et al. [21] and Faliszewski and Rothe [20] for detailed accounts). The study of the complexity of control for restricted domains was initiated by Faliszewski et al. [25], who considered control in single-peaked elections (and who themselves were motivated by the results of Walsh [45] and Conitzer [9]). Brandt et al. [5] and Faliszewski et al. [23] continued this line of work by studying further voting rules and considering nearly single-peaked elections. However, until very recently, there were no papers on control for

Table 1 The complexity of control problems for plurality, Condorcet, and approval rules, for the unrestricted domain (denoted "Un.") [2,28], the single-peaked domain (denoted "SP") [5,25], the single-crossing domain (denoted "SC"), and the voter-interval domain [14] (a variant of the single-crossing domain for approval elections, denoted "VI"; results in bold are due to this paper, the other ones follow from the unrestricted cases)

Problem		Plurality Un./SP/SC	Condorcet Un./SP/SC	Approval Un./SP/VI
CC	AC	NP-com./P/**P**	–	–
	DC	NP-com./P/**P**	P/P/P	P/P/P
	AV	P/P/P	NP-com./P/**P**	NP-com./P/**P**
	DV	P/P/P	NP-com./P/**P**	NP-com./P/**P**
DC	AC	NP-com./P/**P**	P/P/P	P/P/P
	DC	NP-com./P/**P**	–	–
	AV	P/P/P	P/P/P	P/P/P
	DV	P/P/P	P/P/P	P/P/P

A dash in an entry means that the given system is *immune* to the type of control in question (i.e., it is impossible to achieve the desired effect by the action this control problem allows; see [2,28] for the immunity results)

single-crossing elections. Indeed, to the best of our knowledge, our work is the first that tackles this topic. Recently, Bulteau et al. [7] also considered control in single-crossing elections (their main focus was on combinatorial control, where one can add/delete whole groups of candidates/voters; in their model it turned out that a certain variant of control by adding voters is polynomial-time solvable for single-crossing elections, but is NP-complete for single-peaked ones).

In general, there are relatively few papers that study the complexity of election problems under the single-crossing assumption. The few examples include, e.g., the works of Elkind et al. [16] and Cornaz et al. [10], where the authors consider the problem of decloning candidates to obtain a single-crossing election, the work of Skowron et al. [44], who study the complexity of proportional representation in single-crossing elections, and the work of Elkind et al. [15], who study the complexity of recognizing incomplete single-crossing profiles.

From our perspective, one of the most relevant papers is that of Elkind and Lackner [14]. In this work the authors put forward a number of restricted domains for elections where the voters have dichotomous preferences (that is, each voter approves of some candidates and disapproves of the others). While the notion of single-peakedness for dichotomous preferences was introduced by Faliszewski et al. [25], Elkind and Lackner discuss (among other notions) dichotomous single-crossing elections. We use one of their notions in Sect. 5, where we consider approval elections.

1.3 Organization of the paper

The paper is organized as follows. First, in Sect. 2 we provide necessary preliminaries regarding election control and the single-crossing assumption. In the next two sections we present our results regarding plurality and Condorcet elections. We show that all our control problems for these rules are polynomial-time solvable for single-crossing elections. Then, in Sect. 5 we consider approval elections, for which the notion of single-crossing is far from clear. Yet, by looking at a rather natural variant of single-crossing for dichotomous preferences, we also obtain polynomial-time algorithms. We conclude in Sect. 6.

2 Preliminaries

For each positive integer s, we let $[s]$ denote the set $\{1, \ldots, s\}$.We assume that the reader is familiar with basic notions of complexity theory, such as the classes P and NP, the notions of NP-hardness and NP-completeness, and basic reducibility types (see, e.g., the textbook of Papadimitriou [35] for more details).

Elections In the ordinal model, an election is a pair $E = (C, V)$ where C is a set of candidates and V is a set of voters. Each voter v in V has a *preference order* $\succ_v$, i.e., a linear order over C that ranks all the candidates from the most desirable one to the least desirable one. For the case of approval voting, elections are defined in the same way, except that voters do not rank the candidates but simply provide the sets of candidates that they approve of (they disapprove of the other ones). Such voters are said to have dichotomous preferences.

Throughout this paper we mostly focus on the elections in the ordinal model, and we indicate clearly when we mean approval voting (this is almost exclusively limited to Sect. 5).

Voting rules We focus on three voting rules, the plurality rule, the Condorcet rule, and the approval rule. Let $E = (C, V)$ be an election (in the ordinal model). The plurality score of a candidate c in election E, denoted $score^P_E(c)$, is the number of voters in V that rank c first. A candidate is a *plurality winner* if he or she has the highest plurality score. (In effect, we assume the unique winner model, i.e., we require the winner to be the only candidate with the highest score; however, all our results hold in the nonunique winner model as well.)

A candidate is a *Condorcet winner* if he or she defeats every other candidate in a pairwise contest. Formally, for an election $E = (C, V)$ and two candidates $c, c' \in C$, we write $N_E(c, c')$ to denote the number of voters from V that prefer c to c'. A candidate c is a Condorcet winner if for each candidate $c' \in C \setminus \{c\}$ we have $N_E(c, c') > N_E(c', c)$. Naturally, it is possible that there is no Condorcet winner in an election. In such a case, the election has no winners according to the Condorcet rule (while this may seem a bit strange, it is an inherent property of the Condorcet rule, which aims at choosing a candidate that reflects a form of consensus; if there is no consensus, the rule cannot provide a winner).

The approval rule is defined for elections where the voters have dichotomous preferences. An approval score of candidate c in election E, denoted $score^A_E(c)$, is the number of voters in E that approve of c. An *approval winner* is the candidate with the highest approval score.

Single-crossing elections The standard notion of single-crossing elections applies to the ordinal model of elections and here we focus on this case. We discuss an analogous notion for approval voting in Sect. 5.

Let $E = (C, V)$ be an election (in the ordinal model), where C is a set of candidates and V is a set of voters. We can alternatively represent the same election by replacing V with an ordered sequence of the same voters, denoted $\hat{V} = (\hat{v}_1, \ldots, \hat{v}_n)$. Using this notation, we define a single-crossing election as follows [34,39]:[2]

Definition 1 Let $E = (C, V)$ be an election and let $\hat{V} = (\hat{v}_1, \ldots, \hat{v}_n)$ be an ordered sequence of the voters from V. We say that $\hat{V}$ is a *single crossing* order of voters V in

[2] Strictly speaking, Mirrlees [34] and Roberts [39] deal with continuous domains. What we discuss here is sometimes referred to as order-restricted preferences [41]. Nonetheless, we chose to use the term *single crossing* because it is far more informative and because using it in this meaning is common within (computational) social choice literature [6,10,15,16,43,44].

election E if for each pair of candidates a, b such that $a \succ_{\hat{v}_1} b$, there exists a value $t_{a,b} \in [n]$ such that $\{i \in [n] \mid a \succ_{\hat{v}_i} b\} = [t_{a,b}]$.

Definition 2 An election $E = (C, V)$ is *single-crossing* if there is a single-crossing order of V for E.

Intuitively, the above definitions say that as we consider the voters in the single-crossing order from one end of the order to the other, the relative ranking of each two candidates changes at most once (that is, each pair of candidates "crosses" at most once).

In our proofs we will often use the notation $t_{a,b}$ in the sense of Definition 1. While this notation does not specify the election on which it is used, this election will always be clear from the context.

Example 1 Consider an election $E = (C, V)$, where $C = \{a, b, c\}$ and $V = \{v_1, v_2, v_3\}$. The voters have preference orders:

$$v_1 : a \succ b \succ c,$$
$$v_2 : c \succ a \succ b,$$
$$v_3 : c \succ b \succ a.$$

This election is single-crossing and $\hat{V} = (v_1, v_2, v_3)$ is one possible single crossing order of the voters that witnesses this fact. For this order, we have $t_{a,b} = 2$, $t_{a,c} = 1$, and $t_{b,c} = 1$. We note that E is not single-peaked (under any ordering of the alternatives; this is true because in a single-peaked election there can be at most two different candidates that are ever ranked last).

There are efficient polynomial-time algorithms that given an election $E = (C, V)$ check if it is single-crossing and, if so, provide the single-crossing order of the voters [6,11,16]. Thus, from now on, we will assume that each single-crossing election comes together with a single-crossing order.

Control problems We are interested in four variants of election control: control by adding candidates (AC), control by deleting candidates (DC), control by adding voters (AV) and control by deleting voters (DV). For each of these control types we are interested both in constructive control (CC), where the goal is to ensure a given candidate's victory, and in destructive control (DC), where the goal is to prevent some candidate from winning.

Definition 3 Let R be a voting rule. We are given a set of candidates C, a collection of voters V, a nonnegative integer k, and a designated candidate $p \in C$. In control by adding voters (AV) in addition we are given a set W of unregistered voters. Similarly, in control by adding candidates (AC) we are given a set A of unregistered candidates. In constructive variants of control problems we ask whether it is possible for p to become a unique winner under voting rule R in the following way:

1. In constructive control by adding voters (R-CCAV), we ask whether there exists a set $W' \subseteq W$, such that p is the unique winner of R-election $(C, V \cup W')$, where $\|W'\| \leq k$.
2. In constructive control by deleting voters (R-CCDV), we ask whether there exists a set $V' \subseteq V$, such that p is the unique winner of R-election $(C, V \setminus V')$, where $\|V'\| \leq k$.
3. In constructive control by adding candidates (R-CCAC), we ask whether there exists a set $A' \subseteq A$, such that p is the unique winner of R-election $(C \cup A, V)$, where $\|A'\| \leq k$.
4. In constructive control by deleting candidates (R-CCDC), we ask whether there exists a set $C' \subseteq C$, such that p is the unique winner of R-election $(C \setminus C', V)$, where $\|C'\| \leq k$.

The destructive variants are defined analogously except that we ask whether it is possible to ensure that p is not the unique winner, and for the deleting candidates variant it is not allowed to delete candidate p.

When we speak of control problems for the case of single-crossing elections, we mean that the election that contains all the candidates and voters specified in the problem is single-crossing (thus, for example, in CCAC we require that election $(C \cup A, V)$ is single-crossing, and in CCAV we require that election $(C, V \cup W)$ is single-crossing).

3 Plurality elections

We now present our results for the plurality rule (see Table 1 for a quick summary). Voter control problems (that is, the problems of constructive and destructive control by adding/deleting voters) are polynomial-time solvable under plurality [2,28], so we focus on the candidate control problems, which are known to be NP-complete in the unrestricted case [2,28].

In plurality elections, it is important to note that by adding a candidate we can never increase the score of any other candidate who already is in the election. Let us formulate the following definition, which will be used across different proofs in this section.

Definition 4 Let $I = (C, A, V, p, k)$ be an instance of the control by adding candidates problem for the plurality rule. For a set $A' \subseteq A$ and a candidate $a \in C \cup A$, we write that *a reduces* the score of p in election $(C \cup A', V)$ if and only if $score^P_{((C \cup A') \setminus \{a\}, V)}(p) > score^P_{(C \cup A' \cup \{a\}, V)}(p)$.

In other words, if candidate $a \in A'$ reduces the score of p in $((C \cup A') \setminus \{a\}, V)$, then adding a to this election would lower the score of p.

3.1 Constructive control cases

We show that constructive control by adding/deleting candidates can be solved in polynomial time for the case of plurality rule, provided that the underlying election is single-crossing. Both these results are based on the same (quite involved) dynamic programming algorithm that solves a slightly more general problem of control by adding an exact number of candidates. The main idea behind this algorithm is to note that the single-crossing order of the voters implies a certain order in which the candidates should be considered. This is captured in the following (folklore) lemma.

Lemma 1 *In a single-crossing election $E = (C, V)$, if $\hat{V} = (\hat{v}_1, \dots, \hat{v}_n)$ is a single-crossing order for E, then for each candidate $c \in C$ the set of voters who rank c first is a contiguous subsequence of $\hat{V}$.*

Proof Assume to the contrary that there exist two candidates $a, b \in C$ and integers k_1, k_2, and k_3, $1 \le k_1 < k_2 < k_3 \le n$, such that $\hat{v}_{k_1}$ and $\hat{v}_{k_3}$ rank a first while $\hat{v}_{k_2}$ ranks b first. Clearly, this means that $\hat{V}$ is not a single-crossing order for E (the relative order of a and b changes more than once as we consider the voters in the single-crossing order $\hat{V}$). □

The next theorem is the main technical tool for solving constructive candidate control problems under plurality rule. In essence, it gives a polynomial-time algorithm for the problem of control by adding an exact number of candidates for plurality, for the case, where the set of unregistered candidates does not contain a candidate that after being added would affect the score of our preferred candidate p.

Theorem 1 *A variant of Plurality-CCAC for the case of single-crossing elections, where the set of unregistered candidates does not contain candidates which would reduce the score of the preferred candidate and where it is required to add exactly a given number of candidates, is in* P.

Proof We give an algorithm based on dynamic programming. Let (C, A, V, p, k) be an instance of Plurality-CCAC satisfying the requirements of the theorem, where C is the set of registered candidates, A is the set of unregistered candidates, V is the set of voters, p is the preferred candidate, and k is the number of candidates that we need to add. Further, let $\hat{V} = (\hat{v}_1, \dots, \hat{v}_n)$ be a single-crossing order of voters from V (for the candidate set $C \cup A$). For each candidate $c \in C \cup A$, let $\mathit{first}(c)$ and $\mathit{last}(c)$ be, respectively, the minimal and the maximal index of a voter from $\hat{V}$ that ranks c first in election $(C \cup \{c\}, V)$ (it is easy to see that, without loss of generality, we can assume that these values are always well-defined[3]). From Lemma 1, we know that for a given $c \in C \cup A$ the interval $\hat{v}_{\mathit{first}(c)}, \dots, \hat{v}_{\mathit{last}(c)}$ consists of all the voters from V which rank c first in $E = (C \cup \{c\}, V)$; let us refer to this interval as the *visibility interval* of candidate c. Considering the election from Example 1, we see that visibility interval for candidate a is (v_1), for candidate c is (v_2, v_3), and candidate b is not visible. We also see that the *first* and *last* indices for candidates a and c are:

$$\mathit{first}(a) = \mathit{last}(a) = 1 \quad \text{and} \quad \mathit{first}(c) = 2,\ \mathit{last}(c) = 3.$$

We provide further examples of the notation introduced here and below as Example 2, a bit later.

For each set $X \subseteq C \cup A$ and each set $Y \subseteq V$, we define:

$$\mathit{maxscore}(X, Y) = \max_{c \in X \setminus \{p\}} \left(\mathit{score}^P_{(X,Y)}(c)\right)$$

In other words, $\mathit{maxscore}(X, Y)$ is the highest possible plurality score of a candidate from X, but other than p, in election $E = (X, Y)$.

Next, we define a linear order L over the set of candidates A: For each pair a, b of candidates from A, we have $a\ L\ b$ if and only if the following holds (by $\succ_{\mathit{last}(a)}$ we mean the preference order of voter $\hat{v}_{\mathit{last}(a)}$):

$$(\mathit{last}(a) < \mathit{last}(b)) \vee (\mathit{last}(a) = \mathit{last}(b) \wedge a \succ_{\mathit{last}(a)} b).$$

So the order L lines up unregistered candidates by the index of the last voter in their *visibility interval*. In case two *visibility intervals* for two candidates end with the same voter, the candidate that is preferred by this voter takes precedence.

For a candidate $c \in A$, let $\mathit{PreA}(c)$ be the subset of A containing c and all the candidates that precede c under L. For each integer u and each candidate $c \in A$, we define $f(u, c)$ as follows:

$$f(u, c) = \min_{\substack{A' \subseteq \mathit{PreA}(c) \\ \|A'\| = u}} \left\{\mathit{maxscore}(C \cup A', V)\right\}$$

In other words, $f(u, c)$ is the minimal possible score that the highest-scoring candidate (not counting p) can have after adding exactly u candidates from $\mathit{PreA}(c)$ to election (C, V). □

[3] Strictly speaking, candidates which cannot score any points might still be needed. Consider a case where A contains a single candidate who is ranked last by all the voters, p is the current winner, and we ask if there is a set A' of size exactly one, such that p is a winner in $(C \cup A', V)$. We would have to take $A' = A$ and that would mean taking the candidate for whom *first* and *last* are not defined. However, with the way we use Theorem 1 in the paper, such a situation is never relevant. Handling this possibility is easy, but we omit it for the sake of clarity and brevity.

Observation 1 *Let a_{last} be the last candidate from A under L. Since no candidate from A can "steal" points from candidate p (this follows from our assumption on the input data), p can be made the unique winner of the election by adding exactly k candidates from A if and only if $f(k, a_{last}) < score^{P}_{(C,V)}(p)$.*

The above is true because $PreA(a_{last}) = A$, so the value of $f(k, a_{last})$ describes the minimal possible score of the best candidate other than p in election $(C \cup A', V)$ across all possible sets $A' \subseteq A$ of size k. Therefore if that score is lower than the score of p in (C, V), then p will be the winner of $(C \cup A', V)$ (by assumption the score of p cannot change after adding candidates).

Example 2 To illustrate the above definitions, let us consider an instance of Plurality-CCAC with the set of registered candidates $C = \{x, p\}$, where p is the designated candidate, with the set of unregistered candidates $A = \{c, d, e\}$, and with the voter set $V = \{v_1, \ldots, v_9\}$:

$$\begin{aligned}
v_1 &: e \succ d \succ c \succ x \succ p,\\
v_2 &: d \succ c \succ x \succ p \succ e,\\
v_3 &: c \succ x \succ d \succ p \succ e,\\
v_4 &: c \succ x \succ d \succ p \succ e,\\
v_5 &: x \succ c \succ d \succ p \succ e,\\
v_6 &: x \succ c \succ d \succ p \succ e,\\
v_7 &: p \succ x \succ c \succ d \succ e,\\
v_8 &: p \succ x \succ c \succ d \succ e,\\
v_9 &: p \succ x \succ c \succ d \succ e.
\end{aligned}$$

We note that $E = (C \cup A, V)$ is single-crossing and that $\hat{V} = (v_1, \ldots, v_9)$ is a single crossing order of the voters that witnesses this fact. Further, one can verify that sets C, A, and V satisfy all the constraints of Theorem 1 (in particular, adding the candidates from the set $\{c, d, e\}$ to election (C, V) cannot affect the score of p). Let $k = 2$ be the number of candidates that we need to add.

The *first*$(\cdot)$ and *last*$(\cdot)$ values of the unregistered candidates are as follows:

$$\mathit{first}(c) = 1, \quad \mathit{last}(c) = 4, \quad \mathit{first}(d) = 1, \quad \mathit{last}(d) = 2, \quad \mathit{first}(e) = 1, \quad \mathit{last}(e) = 1.$$

Thus the order L is $e \ L \ d \ L \ c$ and we have that $PreA(e) = \{e\}$, $PreA(d) = \{e, d\}$ and $PreA(c) = \{e, d, c\} = A$.

Let us calculate the value $f(2, d)$. By definition, it is the highest score of a candidate other than p in election $(C \cup \{e, d\}, V)$ (note that there is only one size-two subset in $PreA(d)$). This score is four (it is the score of candidate x in election $(C \cup \{e, d\}, V)$). To calculate the value $f(2, c)$, we need to consider three two-element subsets of $PreA(c)$, namely, $\{e, d\}$, $\{e, c\}$ and $\{d, c\}$. We already know that we have $maxscore(C \cup \{e, d\}, V) = 4$. Routine calculations show that we also have:

$$maxscore(C \cup \{c, d\}, V) = 2 \quad \text{and} \quad maxscore(C \cup \{c, e\}, V) = 3.$$

In effect, we have $f(2, c) = 2$. We note that it is possible to ensure p's victory by adding candidates c and d (because the score of p is greater than two).

For each voter $v \in V$, let $PreV(v)$ be the set of voters that includes v and all the voters that precede v (with respect to the single crossing order $\hat{V}$). For each integer u, each candidate

$c \in A$, and each voter $\upsilon \in V$, let $\mathcal{L}(u, c, \upsilon)$ be the family of subsets of candidates such that for each $X \in \mathcal{L}(u, c, \upsilon)$ it holds that: (a) X has exactly u elements, (b) $c \in X$, (c) $X \subseteq PreA(c)$, and (d) candidate c scores a point from voter υ in election $(C \cup X, V)$. For a given integer u, a candidate $c \in A$, and a voter $\upsilon \in V$, we define a helper function g as follows:

$$g(u, c, \upsilon) = \min_{A' \in \mathcal{L}(u,c,\upsilon)} \left\{ maxscore(C \cup A', PreV(\upsilon)) \right\}$$

For a given candidate $c \in A$ and a given integer u, the expression for $g(u, c, \hat{\upsilon}_{last(c)})$ is very similar to that for $f(u, c)$. The only differences are that it considers subsets of $PreA(c)$ that contain c, and that *maxscore* is computed on a limited set of voters (only voter υ and all the voters that precede υ under $\hat{V}$).

We now show how function f can be expressed using function g. For each candidate $c \in A$, by $PreV(c)$ we mean $PreV(\hat{\upsilon}_{last(c)})$, that is, $PreV(c)$ contains all the voters from $\hat{V}$ starting from the first one and up to the last one from the *visibility interval* of c. To express f using g correctly, we need to reconcile the differences between the definitions of f and g mentioned above. As g considers only subsets of $PreA(c)$ of size u that contain c (as opposed to f which does not have the limitation of c being a part of the subset), we scan through all possible candidates $c' \in PreA(c)$ in order to calculate f. The second difference is that f considers all the voters from V in order to compute *maxscore*, while g takes into account only voters up to and including υ. We note that neither of the remaining voters (that is, neither of the voters that follow υ under $\hat{V}$) can rank candidates from $PreA(c)$ as their top choices (otherwise the *visibility intervals* of these candidates would extend beyond υ). Taking this into account, we express f using g as follows:

$$f(u, c) = \min_{c' \in PreA(c)} \left\{ \max(g(u, c', \hat{\upsilon}_{last(c')}), maxscore(C, V \setminus PreV(c'))) \right\} \quad (1)$$

This means that it suffices to be able to compute g in polynomial time to be able to also compute f in polynomial time.

Example 3 Let us go back to Example 2. Naturally, we have:

$$PreV(\upsilon_1) = \{\upsilon_1\}, \quad PreV(\upsilon_2) = \{\upsilon_1, \upsilon_2\}, \quad PreV(\upsilon_3) = \{\upsilon_1, \upsilon_2, \upsilon_3\}, \quad \ldots$$

We also note that $\mathcal{L}(2, d, \upsilon_2) = \{\{e, d\}\}$ and that $\mathcal{L}(2, d, \upsilon')$ is empty for all other voters. Similarly, $\mathcal{L}(2, c, \upsilon_3) = \mathcal{L}(2, c, \upsilon_4) = \{\{e, c\}, \{d, c\}\}$, and $\mathcal{L}(2, c, \upsilon')$ is empty for all remaining voters υ'. Thus for $u = 2$ it only makes sense to consider values of g for candidate d and voter υ_2 or for candidate c and voters υ_3 or υ_4.

Clearly, $g(2, d, \upsilon_2) = 1$ as the only set A' to consider is $\{e, d\}$. In this case, voter υ_1 gives his or her point to e and voter υ_2 gives his or her point to d. Similarly, we have $g(2, c, \upsilon_3) = 2$. In this case we need to consider election among voters υ_1, υ_2, and υ_3 either with added candidates $\{e, c\}$ or with added candidates $\{d, c\}$. In either case, there is a candidate (who is not p) that receives exactly two points. Finally, one can verify that $g(2, c, \upsilon_4) = 2$ (this time by considering the first four voters and the same two sets of candidates to add).

We now compare values $f(2, d) = 4$ and $f(2, c) = 2$ calculated in Example 2 with values that follow from Eq. (1). According to Eq. (1), we have that:

$$f(2, d) = \max\left(g(2, d, \upsilon_2),\ maxscore(C, V\ PreV(d))\right) = \max(1, 4) = 4.$$

Calculating $f(2, c)$ requires a bit more effort, but still can be handled easily and efficiently:

$$\begin{aligned} f(2,c) &= \min\Big(\max\big(g(2,c,v_4),\ maxscore(C, V\ PreV(c))\big),\\ &\qquad \max\big(g(2,d,v_2),\ maxscore(C, V\ PreV(d))\big)\Big)\\ &= \min(\max(2,2), \max(1,4)) = 2. \end{aligned}$$

As expected, in both cases we get the same results as in Example 2.

We now describe a recursive method for computing g (we present this method as Algorithm 1, though the algorithm references the conditions we list below, so the reader might want to consider the algorithm and the description here in parallel). For each integer u, candidate $c \in A$, and voter $v \in V$, in order to compute $g(u, c, v)$ we scan through candidates $c' \in PreA(c)$, such that c scores a point from v in election $E = (C \cup \{c, c'\}, V)$, compute certain values (see below), and eventually return the smallest one of them. Let us fix $c \in A$ and $v \in V$. For each possible $c' \in PreA(c)$, such that c scores a point from v in $E = (C \cup \{c, c'\}, V)$ we consider the following cases:

Condition 1 (***last*(*c*′) < *first*(*c*)**). In this case the visibility intervals of c and c' do not overlap and we return the maximum of $g(u-1, c', last(c'))$ and $maxscore(C \cup \{c\}, PreV(c) \setminus PreV(c'))$.

Condition 2 (***first*(*c*′) > *first*(*c*)**). In this case the visibility interval of c' is within the visibility interval of c. Since V is single-crossing, we know that if both c and c' take part in the election, candidate c' will score no points. We skip this case for now and take it into account later.

Condition 3 (***last*(*c*′) ≥ *first*(*c*)**). In this case the visibility intervals of c and c' overlap. We return the maximum of $g(u-1, c', t_{c',c})$ and $maxscore(C \cup \{c\}, PreV(c) \setminus PreV(t_{c',c}))$, where $t_{c',c}$ is such that all the voters $\hat{v}_1, \ldots, \hat{v}_{t_{c',c}}$ prefer c' to c, and the remaining ones prefer c to c' (recall Definition 1).

Going back to Condition 2, we know that for each c' that satisfies this condition, adding c' has no impact on the scores of the other candidates as long as c is taking part in the election (the visibility interval of c covers the visibility interval of c'). Thus, whether we add such a candidate to the election or not, has no impact on the value of g, except that we might want to add such candidates to ensure that we add exactly u candidates and not any less. Algorithm 1 shows how to recursively compute g referring to all the possible cases described above. All border cases are covered and we take Condition 2 into account in the following way: We compute the number t of candidates that satisfy it and allow the recursive calls to g to use between $u-1$ and $u-1-t$ candidates (modeling the fact that we can add up to t candidates satisfying Condition 2). The algorithm returns ∞ if no subset $A' \subseteq A$ satisfying the definition of g exists.

Given the recursive formulation, expressed as Algorithm 1, we use standard dynamic programming techniques to compute g in polynomial time (strictly speaking, for each nonnegative integer $u < \|A\|$, each candidate $c \in A$ and each voter $v \in V$, $g(u, c, v)$ is computable in time $O(n^2\|A\|^3)$). Using Eq. (1), we compute $f(u, i)$ for each nonnegative $u \leq \|A\|$ and for each $c \in A$, in time $O(n^2\|A\|^4)$. As per Observation 1, being able to compute f suffices to solve our problem.

Now we generalize the above theorem by lifting the restriction that adding a candidate cannot hurt our preferred candidate.

Algorithm 1 Compute $g(u, c, v)$ recursively

```
if $c$ does not score point from $v$ in $E = (C \cup \{c\}, V)$ then
    return $\infty$
else if $u = 1$ then
    return $maxscore(C \cup \{c\}, PreV(v))$
end if
$result \leftarrow \infty$
$t \leftarrow$ the number of candidates $c' \in PreA(c) \setminus \{c\}$ that satisfy Condition 2
for $c' \in PreA(c) \setminus \{c\}$ do
    if $c$ scores a point from $v$ in $E = (C \cup \{c, c'\}, V)$ then
        $r' \leftarrow \infty$
        if Condition 1 is satisfied then
            $r' \leftarrow \min_{t' \in \{0,\dots,t\}}\{\max(g(u-1-t', c', last(c'));$
            $maxscore(C \cup \{c\}, PreV(c) \setminus PreV(c'))\}$
        else if Condition 3 is satisfied then
            $r' \leftarrow \min_{t' \in \{0,\dots,t\}}\{\max(g(u-1-t', c', t_{c',c});$
            $maxscore(C \cup \{c\}, PreV(c) \setminus PreV(t_{c',c}))\}$
        end if
        $result \leftarrow \min(result; r')$
    end if
end for
return $result$
```

Theorem 2 *A variant of Plurality-CCAC for the case of single-crossing elections, where it is required to add exactly a given number of candidates, is in* P.

Proof We show a Turing reduction of our current problem to the one from Theorem 1 (in other words, we show how we can solve our current problem using the algorithm from Theorem 1 as a subroutine). Let $I = (C, A, V, p, k)$ be our input instance. Let $\hat{V} = (\hat{v}_1, \dots, \hat{v}_n)$ be a single-crossing order of voters from V. For each set $H \subseteq C \cup A$ and each set $G \subseteq A$ such that $G \cap H = \emptyset$, let $J(H, G) = (H, G, V, p, k - \|H \cap A\|)$ be an instance of the problem in the format for Theorem 1 (though, to use Theorem 1, we would have to make sure the set of candidates G does not contain a candidate that would affect the score of p after being added to the election).

For each pair of candidates $a, b \in A$ (we allow $a = b$), let $G_{a,b}$ be a subset of A created by removing from set A candidates a, b and all candidates that can reduce the score of p (in the sense from Definition 4) in election $(C \cup \{a, b\}, V)$. Now we claim that for $k > 2$, I has a solution if and only if there exist $a, b \in A$, such that $J(C \cup \{a, b\}, G_{a,b})$ has a solution. The right-to-left implication is trivial, so let us focus on proving the other direction.

Let us assume that instance I has a solution and let A' be the set of candidates added in order to solve I. Let L be a subset of candidates from A' which are preferred to p by $\hat{v}_1$. We consider the case in which both L and $A' \setminus L$ are nonempty sets. We will later show how to generalize the following reasoning to the case where one of those sets is empty. Now let x be a candidate from L with maximum value of $t_{x,p}$ (recall Definition 1). If more than one candidate reaches the maximum value, let us select the one that is most preferred over all the others by the voter at index $t_{x,p}$ in $\hat{V}$. Similarly, let us select candidate y from $A' \setminus L$ with minimum value of $t_{p,y}$ (we break ties by selecting the one that is most preferred by the voter at index $t_{p,y}$ in $\hat{V}$). From Lemma 1, we know that the set of voters who rank p first in $(C \cup A', V)$ is a contiguous subsequence of $\hat{V}$. Clearly, this subsequence is located between indices $t_{x,p}$ and $t_{p,y}$. Thus it is easy to see that no candidate from $A' \setminus \{x, y\}$ can reduce the score of p in $(C \cup \{x, y\}, V)$. Therefore, clearly, set A' is a subset of $G_{x,y} \cup \{x, y\}$, and since $\|A' \setminus \{x, y\}\| = k - 2$, it is easy to see that by adding candidates from $A' \setminus \{x, y\}$ we can solve instance $J(C \cup \{x, y\}, G_{x,y})$. It should be obvious now, that in case when either L or $A' \setminus L$ is empty, we can solve instances $J(C \cup \{x\}, G_{x,x})$ or $J(C \cup \{y\}, G_{y,y})$ respectively.

The case where $k \in \{0, 1\}$ is trivial since then we can solve our problem by brute-force. Therefore, in order to determine whether I has a solution we need to verify whether the solution exists for at least one out of less than $\|A\|^2$ instances of the problem from Theorem 1. Since the described reduction to the problem from Theorem 1 is polynomial-time and this problem is in P, we conclude our problem is in P as well. □

Finally, it is easy to prove our main result of this section.

Theorem 3 *Plurality-CCAC and Plurality-CCDC for single-crossing elections are both in* P.

Proof We Turing-reduce Plurality-CCAC and Plurality-CCDC for single-crossing elections to the problem from Theorem 2.

Let $I_{CCAC} = (C, A, V, p, k)$ be an instance of Plurality-CCAC problem. That is, we are given a set of candidates C, a set of unregistered candidates A, a voter collection V, a designated candidate $p \in C$, and a nonnegative integer k—the upper bound for the number of unregistered candidates that can be added to the election. In addition, we know that $E = (C \cup A, V)$ is single-crossing. The reduction proceeds as follows. For each nonnegative integer s let $J_s = (C, A, V, p, s)$ be an instance of the problem from Theorem 2. It is easy to see that there exists a solution for I_{CCAC} if and only if for some $s \in \{0, \dots, k\}$ there exists a solution for J_s. Clearly, the reduction runs in polynomial-time and, thus, Plurality-CCAC for single-crossing elections is in P.

Let $I_{CCDC} = (C, V, p, k)$ be an instance of Plurality-CCDC problem, where election $E = (C \cup A, V)$ is single-crossing. For each nonnegative integer s, let $J'_s = (\{p\}, C \setminus \{p\}, V, p, s)$ be an instance of the problem from Theorem 2. It is easy to see that solution for I_{CCDC} exists if and only if there exists a solution of J'_s for some $s \in \{m - 1 - k, \dots, m - 1\}$. (In other words, it exists in some instance J'_s, where we have to "bring back" all the candidate except up to k of them.) The reduction clearly runs in polynomial-time and, so, Plurality-CCDC for single-crossing election is in P. □

3.2 Destructive control cases

Let us now consider the case of destructive candidate control under plurality. For the case of control by deleting candidates, we simply use an adapted version of the algorithm for the constructive case. For control by adding candidates we derive a dedicated, faster algorithm. To this end, we use the following result.

Lemma 2 *If destructive control by adding candidates is possible in single-crossing election under plurality, then it can be achieved by adding at most three candidates from the set of unregistered ones.*

Proof We are given a set of candidates C, a set of unregistered candidates A, a collection of voters V, and a designated candidate p. We know that election $E = (C \cup A, V)$ is single-crossing.

Assume that there exists $A' \subseteq A$, $\|A'\| \geq 4$, such that in election $(C \cup A', V)$ candidate p is not a unique winner under plurality rule. Let $\hat{V} = (v_1, \dots, v_n)$ be the single-crossing order of voters from V in election (C, V). From Lemma 1, we know that voters that rank p first in (C, V) represent a consecutive subsequence of $\hat{V}$ and we refer to that sequence as $\hat{V}^p$ from now on. Another observation is that if some candidate $a \in A'$ reduces the score of

p in (C, V), then the set of voters that rank a first in $(C \cup \{a\}, V)$ but that rank p first in (C, V) is a consecutive subsequence of $\hat{V}^p$ and is either a prefix of $\hat{V}^p$ or a suffix of $\hat{V}^p$. This is so because, due to Lemma 1, the set of voters that rank p first cannot be split into two consecutive subsequences. A corollary of this is the following observation. □

Observation 2 *For every three distinct candidates $a, b, c \in A'$ that each reduce the score of p in (C, V), there exists one candidate $x \in \{a, b, c\}$ that does not reduce the score of p in $(C \cup \{a, b, c\}, V)$.*

(For example, if both a and b reduced the score of p in $(C \cup \{a, b, c\}, V)$, then it would have to be the case that one of them were ranked first by some prefix of the voters in $\hat{V}^p$, whereas the other one were ranked first by some suffix of these voters. However, if c also reduced the score of p in $(C \cup \{a, b, c\}, V)$ then either some prefix or some suffix of the voters from $\hat{V}^p$ would have to rank c first, which would be impossible.)

We now show how to reduce the size of the unregistered candidate set A' by removing a single candidate from it, without making p a unique winner. Let us select any four distinct candidates $a, b, c, d \in A'$ and consider the three following cases:

1. At least two candidates from $\{a, b, c, d\}$ do not reduce the score of p in (C, V). Let $x \in \{a, b, c, d\}$ be one of those candidates with the lowest score in $E = (C \cup A', V)$ (we break ties in an arbitrary fashion). We know that x is not a unique winner of $(C \cup A')$. Since x does not reduce the score of p in (C, V), the score of p will remain the same after we remove x from the election E (the scores of other candidates may increase). Therefore in election $(C \cup A \setminus \{x\})$ candidate p is not a unique winner.
2. Exactly three candidates from $\{a, b, c, d\}$ reduce the score of p in (C, V). Let us name those three candidates x, y, z and let us name the remaining 4th candidate r. Based on Observation 2, in group x, y, z, there is one candidate g that does not reduce the score of p in $(C \cup \{x, y, z\}, V)$. If g has lower score than r in $E = (C \cup A', V)$ we see that g can be safely removed from E without making p a unique winner. Otherwise we can remove r, retaining the same characteristics.
3. All candidates from $\{a, b, c, d\}$ reduce the score of p in (C, V). Based on Observation 2, we can now select two candidates $x, y \in \{a, b, c, d\}$ that do not reduce the score of p in $(C \cup \{a, b, c, d\}, V)$. Similarly as in the case above, we can remove the one with lower score in $(C \cup A, V)$, thereby not improving the score of candidate p and not removing the previous best scoring candidate.

If our assumption regarding the existence of A' is true, then we can use the above-described technique to find a candidate $x \in A'$ such that in election $(C \cup A' \setminus \{x\}, V)$ candidate p is not a unique winner. This way, by removing candidates one by one, eventually we can produce a set A'' such that $\|A''\| = 3$ and p is not a winner of $(C \cup A'', V)$. This proves the lemma.

In effect, to solve Plurality-DCAC it suffices to try all up-to-three-elements subsets of candidates to add. This leads to an algorithm that is much faster than the one for the constructive case.

Theorem 4 *Plurality-DCAC for single-crossing elections is in* P.

Proof We are given a set of candidates C, a set of unregistered candidates A, a collection of voters V, a designated candidate p and an integer k, the upper bound for the number of voters from A that can be added to the election. We are given that $E = (C \cup A, V)$ is single-crossing. From Lemma 2 follows that if destructive control by adding candidates is possible in E, then it can be achieved by adding at most three candidates. Therefore we can scan through all the subsets of A of size at most $\min(3, k)$ and check whether adding each given subset of candidates render that p is not a unique winner. Since there are no more than $\|A\|^3$ such subsets, we can find a solution of Plurality-DCAC in polynomial time. □

For the case of destructive control by deleting candidates, we invoke the constructive algorithm for every candidate other than the designated one (taking into account small modification due to fact that we use the unique-winner model and that it is illegal to delete the designated candidate).

Theorem 5 *Plurality-DCDC for single-crossing elections is in* P.

Proof We are given a set of candidates C, a collection of voters V such that $E = (C, V)$ is single-crossing, and a designated candidate p. We show a Turing reduction from the modified Plurality-CCDC for single-crossing elections, where (a) the designated candidate p wins when no other candidate has higher score, and (b) where there is another designated candidates, d, that cannot be deleted. The modified version of Plurality-CCDC can be easily proved to be in P by minor modifications in the proofs from Theorems 2 and 3. Let $I = (C, V, p, k)$ be an instance of Plurality-DCDC for single-crossing election. For each candidate $c \in C$, let $J_c = (C, V, c, p, k)$ be an instance of the modified Plurality-CCDC for single-crossing (p is the candidate that cannot be deleted). It is easy to see that I has a solution if and only if there exists a solution of J_c for some $c \in C \setminus \{p\}$. Thus the reduction is correct and runs in polynomial-time. □

4 Condorcet elections

Let us now move on to the case of Condorcet elections. Under the Condorcet rule, candidate control and destructive voter control problems are easy even in the unrestricted setting, but constructive voter control problems are NP-complete [2,28]. Yet, these problems are polynomial-time solvable for the case of single-peaked elections and here we show that the same holds for single-crossing elections (see Table 1).

Our algorithms rely on a variant of the median voter theorem for single-crossing elections (see the paper of Rothstein [42] and the discussion in the work of Saporiti and Tohmé [43]). For the sake of completeness, below we state and prove the exact variant of this theorem that we use (recall Definition 1 for the meaning of notation $t_{a,b}$ where a and b are two candidates).

Lemma 3 *Let $E = (C, V)$ be a single-crossing election and let $\hat{V} = (\hat{v}_1, \ldots, \hat{v}_n)$ be a single-crossing order of the voters from V. Candidate $p \in C$ is a Condorcet winner if and only if it is the most preferred candidate by the median voter ($\hat{v}_{\lceil n/2 \rceil}$) or both median voters in case n is even ($\hat{v}_{n/2}$ and $\hat{v}_{n/2+1}$).*

Proof Let c_P be the the most preferred candidate by the median voter or both median voters when n is even. Let $C_L = \{c \in C \mid c \succ_{\hat{v}_1} c_P\}$ and $C_R = \{c \in C \mid c_P \succ_{\hat{v}_1} c\}$. From the definition of single-crossingness, it is easy to see that for each candidate $c_R \in C_R$, c_P is preferred over c_R by all the voters from $\{\hat{v}_1, \ldots, \hat{v}_{\lceil n/2 \rceil}\}$. Similarly, for each candidate $c_L \in C_L$, c_P is preferred over c_L by all the voters from $\{\hat{v}_{\lceil n/2 \rceil}, \ldots, \hat{v}_n\}$. Since, clearly, $C_L \cup C_R = V \setminus \{c_P\}$ we see that c_P is preferred by more than half of the voters over each candidate from $V \setminus \{c_P\}$ and thus is a Condorcet winner, which ends the proof for the left-to-right direction.

Correctness of the right-to-left direction for the case when n is odd comes directly from the above as in that case median voter always exists. For the case when n is even, it is sufficient to note that when each of the two median voters prefers a different candidate then Condorcet winner does not exist at all. □

Using this result, it is possible to derive simple greedy algorithms for Condorcet-CCDV and Condorcet-CCAV.

Theorem 6 *Condorcet-CCDV and Condorcet-CCAV for single-crossing elections are in* P.

Proof Proofs for the cases of adding and deleting voters are very similar. To solve CCDV and CCAV we simply try to delete/add voters in a correct place considering the single-crossing order of voters until the median voter ranks p first. Below we formalize those steps for both CCDV and CCAV.

For Condorcet-CCDV we are given a set of candidates C, a set of voters V of size n, a designated candidate $p \in C$, and an integer k, the upper bound on the number of voters that we can remove. By assumption, we know that $E = (C, V)$ is single-crossing and we are given a single-crossing order $\hat{V} = (\hat{v}_1, \dots, \hat{v}_n)$ of the voters.

We define V_P to be the set of voters from $\hat{V}$ that rank p first. Note that voters from V_P form a contiguous block within $\hat{V}$. Let V_L be the set of voters that precede the voters from V_P (under $\hat{V}$) and, similarly, let V_R be the set of voters that follow the voters from V_P. We now focus on finding a way for a voter from V_P to become the median voter (or a way for two voters from V_P to become the two median voters). If $V_P = \emptyset$ then p can never be a Condorcet winner, so let us assume that this is not the case. If we have $\|V_L\| < n/2$ and $\|V_R\| < n/2$ then no voter needs to be deleted and p is a Condorcet winner. In all the remaining cases, at most one of those two sets has more than $n/2$ voters. Let us assume without loss of generality that $\|V_L\| > \|V_R\|$. It suffices to keep on deleting voters from V_L until either p becomes a Condorcet winner or we delete more voters than we are allowed. It is easy to see that we can calculate sets V_P, V_L and V_R in polynomial time and, thus, Condorcet-CCDV is in P.

Let us now consider the adding voters case. As an additional part of the input, we are given a set W of size n' of unregistered voters. We know that election $(C, V \cup W)$ is single-crossing. Let $\hat{V} = \{\hat{v}_1, \dots, \hat{v}_{n+n'}\}$ be a single-crossing order of the voters in this election. We use the same definition for sets V_P, V_L and V_R as in the deleting-voters case. In addition we define W_P, W_L and W_R as subsets of W in a similar manner: W_P is the set of voters from W that rank p first, W_L is the set of voters from W that precede those from $V_p \cup W_p$ (under $\hat{V}$), and W_R is the set of voters from W that follow those from $V_P \cup W_P$. Based on Lemma 3, we can express the Condorcet-CCAV problem as that of finding a set $W' \subseteq W$ of size at most k, such that the median voter from the set $V \cup W'$ (under $\hat{V}$) is from $V_P \cup (W_P \cap W')$ (or, that both the median voters in $V \cup W'$ are from $V_P \cup (W_P \cap W')$, if $\|V\| + \|W'\|$ is even). It is easy to see that we can always start by adding all the voters from W_P (at most k of them). So, without loss of generality, we may assume W_P to be empty (if it is not empty, we add at most k elements from W_P to V_P and reduce k by the number of voters we have added). With that assumption in mind, we reject if V_p is empty (it is impossible for p to become a Condorcet winner). Then we verify if the median voter (or, both median voters if the number of voters in the election is even) is from the set V_P. If so, then p already is a Condorcet winner and we don't need to add any voters. Otherwise we will be adding voters from either W_L or W_R. Without loss of generality, we assume that $\|V_L\| > \|V_R\|$. We keep on adding voters from V_R until either p becomes the Condorcet winner or we cannot add any more voters. Polynomial running time of the algorithm is straightforward to verify. □

Theorem 6 has immediate consequences regarding winner determination under the Young's rule. The winner(s) under the Young's rule are those candidates that can become

Condorcet winners after deleting the fewest voters [46]. While it is well-known that computing Young winners is NP-hard (indeed, complete for parallel access to NP [40]), there is a polynomial-time algorithm for the case of single-peaked elections [5]. As an immediate corollary of Theorem 6, we get a polynomial-time algorithm for the case of single-crossing elections.

Corollary 1 *There is a polynomial-time algorithm that given a single-crossing election computes Young winners of this election.*

For more information regarding various Condorcet-based election rules and their complexity, we point the readers to the recent surveys of Brandt et al. [4], Fischer et al. [26] and Caragiannis et al. [8].

5 Approval elections

The notion of single-crossing elections is well-established for the ordinal model of elections. On the other hand, the case of single-crossing elections has, so far, been almost completely neglected. To the best of our knowledge, the only paper that discusses this notion is a very recent work of Elkind and Lackner [14]. Among a wealth of other domain restrictions for approval elections, Elkind and Lackner provide the following definition of *voter-interval* restriction (VI) that, to our taste, is an appealing analog of single-crossing elections for approval voting.

Definition 5 Let $E = (C, V)$ be an approval election and let $\hat{V} = (\hat{v}_1, \dots, \hat{v}_n)$ be an ordered sequence of the voters from V. We say that $\hat{V}$ is a *voter-interval* order of voters V in election E if for every candidate c it holds that the voters that approve of c form an interval with respect to $\hat{V}$ (i.e., there are numbers $f(c)$ and $e(c)$ such that exactly voters $\hat{v}_{f(c)}, \hat{v}_{f(c)+1}, \dots, \hat{v}_{e(c)}$ approve of c).

Definition 6 An approval election $E = (C, V)$ has the *voter interval* property if there is a voter-interval order of V for E.

Elkind and Lackner show that there is a polynomial-time algorithm that given an approval election decides if it has the voter-interval property and, if so, computes the voter-interval order.

The reader may wonder why this definition is natural. There are at least two good arguments in its favor. First, the voter-interval property is, in some sense, a notion orthogonal to that of single-peakedness for approval voting. While we have not discussed single-peakedness in detail in this work, let us briefly explain the relation between these two notions. The notion of single-peakedness for approval elections was defined by Faliszewski et al. [25] as follows: An approval election is single-peaked if it is possible to order the candidates so that for each voter v, the candidates that v approves of form an interval. The voter-interval property is defined in the same way, with the exception that we swap the roles of candidates and voters. Since in the ordinal election model single-peakedness and single-crossingness also "feel orthogonal" (although this is not a formal statement), we take it as an argument in favor or the voter-interval property.

Second, the voter-interval property seems to follow, to some extent, the intuition that lead to defining the single-crossing notion of ordinal preferences. One view of the notion of single-crossingness in ordinal elections is as follows: We consider two candidates, a and b, where one is more “left-wing” and the other one is more “right-wing”. As we pass on over the order of voters, say from left to right, we first consider very left-wing voters who prefer a to b, we move toward more centrist ones, eventually we see the first voter who prefers b to a and then we move on toward more and more right-wing voters. For the case of the voter-interval property the situation is similar. We take candidate c and we move over the voter spectrum from the left-hand side toward the right-hand side. While c might not be approved by the extremist voters, eventually it would move toward the range of voters for whom c is a “good enough” candidate, that approve him or her. Eventually, we pass over those voters and, once again, reach extremist voters who disapprove of c.

Yet, the voter-interval property is not a perfect analog of single-crossingness. Let us consider the following example.

Example 4 Consider once again the election from Example 1:

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

This election is single-crossing with respect to the order (v_1, v_2, v_3) and its reverse, but not with respect to any other order. Now, based on this election we derive an approval one by saying that every voter approves of his or her top two candidates. The resulting election is not voter-interval with respect to the order (v_1, v_2, v_3) because v_1 and v_3 approve of b, but v_2 does not (analogous reasoning shows that the election is not voter-interval with respect any other voter order).

In spite of this shortcoming, we still believe that the voter-interval property is a very interesting domain restriction that captures many intuitions behind the notion of single-crossing elections. It is also useful algorithmically. In particular, below we show that constructive control by adding/deleting voters in voter-interval elections is polynomial-time solvable.

Theorem 7 *Approval-CCDV for voter-interval elections is in* P.

Proof Consider an instance of Approval-CCDV with candidate set C, voter set V, bound k on the number of voters that can be deleted, and preferred candidate p. We assume that election $E = (C, V)$ has the voter-interval property and we let $\hat{V} = (\hat{v}_1, \dots, \hat{v}_n)$ be a voter-interval order of the voters from V. We assume that p is approved of by at least one voter. Otherwise we can output the negative solution for our problem immediately.

Clearly, it never makes sense to delete voters that approve of p. For each candidate $c \in C \setminus \{p\}$, we define $\gamma(c)$ to be $score_E^A(c) - score_E^A(p)$, that is, the difference between the score of c and the score of p in the original election. Let $\widetilde{C}$ be the set of candidates $c \in C \setminus \{p\}$ for whom $\gamma(c) \geq 0$.

Let V' be the subset of voters from V who do not approve of p, and let $n' = \|V'\|$. Let $(v'_1, \dots, v'_{n'})$ be a voter-interval order of the voters from V' (it is easy to see that this order exists). For each candidate $c \in \widetilde{C}$, let $f(c)$ and $e(c)$ be two numbers such that in V' exactly voters $v'_{f(c)}, v'_{f(c)+1}, \dots, v'_{e(c)}$ approve of p. (Technically, these numbers may not be well-defined. However, this happens only for those candidates in $\widetilde{C}$ of whom no voter in V' approves. Such candidates are approved of exactly by the same voters as p and, thus, if they exist it is impossible to ensure that p is a unique winner of the election through deleting voters).

Our algorithm proceeds by executing the following (greedy) operation until either we ensure that p is a winner or exceed the number of voters that we can delete. First, we find a candidate $c \in \widetilde{C}$ for whom $e(c)$ is minimal (we break ties arbitrarily). We check if $e(c) - \gamma(c) + 1 \geq f(c)$. If so, we delete the $\gamma(c)$ voters $v'_{e(c)}, v'_{e(c)-1}, \ldots, v'_{e(c)-\gamma(c)+1}$. If not, we reject (it is impossible to delete sufficiently many voters that approve of c). Finally, we recalculate V', the values γ, and $\widetilde{C}$ (taking into account the fact that we deleted some voters). If $\widetilde{C}$ is not empty, then we repeat the above procedure. If it is, we check if we deleted at most k voters. If so, we accept. Otherwise, we reject. □

Example 5 We illustrate the algorithm on the following election with candidate set $C = \{p, c_1, c_2, c_3, c_4, c_5\}$ and with nine voters $v_1, \ldots, v_9$ (below for each voter we indicate with a '+' which candidates he or she approves of):

	p	c_1	c_2	c_3	c_4	c_5
v_1						$+$
v_2			$+$			$+$
v_3		$+$	$+$			$+$
v_4		$+$	$+$			
v_5	$+$	$+$	$+$			
v_6	$+$	$+$	$+$		$+$	
v_7	$+$			$+$	$+$	
v_8				$+$	$+$	
v_9				$+$		

We see that the election has the voter-interval property for the natural ordering of the voters. The approval score of p is three, whereas every other candidate has score three or higher. Since voters v_5, v_6 and v_7 approve p, we never delete either of them. For all the candidates (not counting p), candidate c_5 has the smallest $e(\cdot)$ value and so we first delete voter v_3. This ensures that c_5 has score two (lower than p). The next candidates to consider (in the order of the lowest $e(\cdot)$ value among the candidates whose score matches or exceeds that of p) are c_1 and c_2. We can choose either of them and we pick c_1. In effect, we delete voter v_4. Then we have to consider c_2 and delete voter v_2. Finally, due to candidate c_4, we need to delete voter v_8 (which also takes care of candidate c_3). All in all, we see that to ensure that p is the unique winner, we need to delete voters v_1, v_2, v_3, v_4 and v_8. One can verify that, indeed, this is an optimal solution.

To see why this algorithm is correct, it suffices to show the correctness of the first greedy step performed in the algorithm. Let V', values γ, and $\widetilde{C}$ be as just before deleting the first voter, and let c be the candidate chosen by the algorithm in the first iteration. We have to delete $\gamma(c)$ voters that approve of c, that is, some $\gamma(c)$ voters among $v'_{f(c)}, \ldots, v'_{e(c)}$. For each two numbers i, j such that $f(c) \leq i < j \leq e(c)$ it holds that the set of candidates approved by v'_i is equal to or is a subset of the candidates approved by v'_j. (This is due to the fact that the election is voter-interval and due to the choice of c.) Thus deleting voters $v'_{e(c)}, v'_{e(c)-1}, \ldots, v'_{e(c)-\gamma(c)+1}$ is optimal. After executing one iteration of our algorithm we face a problem of the same type and we proceed iteratively. This completes the proof

The case of constructive control by adding voters turns out to be quite similar. We first add all the votes that approve of the preferred candidate and then delete sufficiently many of them, to not exceed the number of voters that we can add.

Theorem 8 *Approval-CCAV for voter-interval elections is in* P.

Proof Consider an instance of Approval-CCAV with candidate set C, voter set V, set of unregistered voters W, bound k on the number of voters that we can add, and preferred candidate p. We assume that election $(C, V \cup W)$ has the voter-interval property. Further, without loss of generality, we assume that every voter in W approves of p (otherwise we could remove this voter from W because it is never beneficial to add a voter who does not approve of p).

We form election $E = (C, V \cup W)$. If p is not a unique winner in this election then, clearly, it is impossible to ensure his or her victory (this is because every voter in W approves of p). We compute number $k' = \max(\|W\| - k, 0)$. Intuitively, k' is the number of W-voters that we need to delete from E to make sure that it can be obtained from (C, V) by adding k voters from W (since every voter in W approves of p, we can focus on the case where we add exactly k voters).

After we delete k' of the W-voters from E, the score of p will decrease to $score_E^A(p) - k'$. Thus, we have to delete voters in such a way that afterward every candidate in $C \setminus \{p\}$ has lower score. For each candidate $c \in C \setminus \{p\}$ we define $\gamma(c) = score_E^A(c) - (score_E^A(p) - k') + 1$. Intuitively, for each $c \in C \setminus \{p\}$, $\gamma(c)$ is the number of W-voters that approve of c, that we need to delete.

At this point, our problem is, in essence, the same as in the case of control by deleting voters. We consider an election $E' = (C, W)$, which has the voter-interval property, we have to delete exactly k' voters, and we have to make sure that for each $c \in C \setminus \{p\}$ we delete at least $\gamma(c)$ voters that approve of c. Thus we use the same algorithm as in the proof of Theorem 7 above. (Note that, technically, the algorithm from the above theorem might delete fewer than k' voters to ensure the constraints regarding the γ values; in this case we pick further voters to delete arbitrarily).

The correctness of the algorithm and the running time follow directly from its construction. □

6 Conclusions

We have studied the complexity of control for plurality and Condorcet voting rules, for the case of elections that satisfy the single-crossing assumption. We have shown that, as in the case of single-peaked elections, all our control problems turn out to be polynomial-time solvable (whereas, without assumptions on the nature of voters' preferences, candidate control problems are NP-complete for plurality and constructive voter control problems are NP-complete for Condorcet's rule). We believe that our results are significant for at least two reasons. First, they provide another argument that NP-hardness results regarding election problems for the unrestricted domain are a weak sign of their practical difficulty. Second, currently there are only few computational results regarding the complexity of single-crossing elections and it is useful to provide results and techniques that can be used to study this domain restriction.

There is a number of research directions motivated by our work. Naturally, it would be interesting to study the complexity of control for single-crossing elections for other voting rules. Second, it would be interesting to find other variants of the notion of single-crossingness for approval-based elections than the voter interval property. It would also be interesting to see how close real-life elections are to being single-crossing, and if algorithms and approaches similar to ours could be used on nearly single-crossing elections (using PrefLib [32] data, such tests would be possible). Finally, we would like to find out to what extent our results could be generalized to the notion of top-monotonicity of Barberà and Moreno [1] that captures both the notions of single-peaked and single-crossing elections.

Acknowledgements We would like to thank the ECAI reviewers for a very useful set of comments. This work was supported in part by NCN grants 2012/06/M/ST1/00358, and by the Polish Ministry of Science and Higher Education (under AGH University Grant 11.11.230.015; statutory research).

References

1. Barberà, S., & Moreno, B. (2011). Top monotonicity: A common root for single peakedness, single crossing and the median voter result. *Games and Economic Behavior*, *73*(2), 345–359.
2. Bartholdi, J, I. I. I., Tovey, C., & Trick, M. (1992). How hard is it to control an election? *Mathematical and Computer Modeling*, *16*(8/9), 27–40.
3. Black, D. (1958). *The theory of committees and elections*. Cambridge: Cambridge University Press.
4. Brandt, F., Brill, M., & Harrenstein, P. (2016). Tournament solutions. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Handbook of computational social choice*, Chap. 3. Cambridge: Cambridge University Press.
5. Brandt, F., Brill, M., Hemaspaandra, E., & Hemaspaandra, L. (2015). Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research*, *53*, 439–496.
6. Bredereck, R., Chen, J., & Woeginger, G. (2013). A characterization of the single-crossing domain. *Social Choice and Welfare*, *41*(4), 989–998.
7. Bulteau, L., Chen, J., Faliszewski, P., Niedermeier, R., & Talmon, N. (2015). Combinatorial voter control in elections. *Theoretical Computer Science*, *589*, 99–120.
8. Caragiannis, I., Hemaspaandra, E., & Hemaspaandra, L. (2016). Dodgson's rule and Young's rule. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Handbook of computational social choice*, Chap. 5. Cambridge: Cambridge University Press.
9. Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, *35*, 161–191.
10. Cornaz, D., Galand, L., & Spanjaard, O. (2013). Kemeny elections with bounded single-peaked or single-crossing width. In *Proceedings of the 23rd international joint conference on artificial intelligence* (pp. 76–82).
11. Doignon, J., & Falmagne, J. (1994). A polynomial time algorithm for unidimensional unfolding representations. *Journal of Algorithms*, *16*(2), 218–233.
12. Dutta, B., Jackson, M. O., & Le Breton, M. (2001). Strategic candidacy and voting procedures. *Econometrica*, *69*(4), 1013–1037.
13. Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. In *Proceedings of the 10th international World Wide Web conference (March 2001)* (pp. 613–622). New York: ACM Press.
14. Elkind, E., & Lackner, M. (2015). Structure in dichotomous preferences. In *Proceedings of the 24th international joint conference on artificial intelligence (2015)*.
15. Elkind, E., Faliszewski, P., Lackner, M., & Obraztsova, S. (2015). The complexity of recognizing incomplete single-crossing preferences. In *Proceedings of the 29th AAAI conference on artificial intelligence* (pp. 865–871).
16. Elkind, E., Faliszewski, P., & Slinko, A.(2012). 'Clone structures in voters' preferences'. In *Proceedings of the 13th ACM conference on electronic commerce (June 2012)* (pp. 496–513).
17. Ephrati, E., & Rosenschein, J. (1997). A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, *20*(1–4), 13–67.
18. Erdélyi, G., & Rothe, J. (2010). Control complexity in fallback voting. In *Proceedings of computing: The 16th Australasian theory symposium*. Conferences in Research and Practice in Information Technology Series (January 2010) (Vol. 32, pp. 39–48). Sydney: Australian Computer Society.
19. Erdélyi, G., Fellows, M., Rothe abd, J., & Schend, L. (2015). Control complexity in Bucklin and fallback voting: An experimental analysis. *Journal of Computer and System Sciences*, *81*(4), 661–670.

20. Faliszewski, P., & Rothe, J. (2015). Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Handbook of computational social choice*, Chap. 7. Cambridge: Cambridge University Press.
21. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2010). Using complexity to protect elections. *Communications of the ACM*, *53*(11), 74–82.
22. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2011). Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, *40*, 305–351.
23. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2014). The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence*, *207*, 69–99.
24. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009). Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, *35*, 275–341.
25. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2011). The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, *209*(2), 89–107.
26. Fischer, F., Hudry, O., & Niedermeier, R. (2016). Weighted tournament solutions. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, & A. D. Procaccia (Eds.), *Handbook of computational social choice*, Chap. 4. Cambridge: Cambridge University Press.
27. Ghosh, S., Mundhe, M., Hernandez, K., & Sen, S. (1999). Voting for movies: The anatomy of recommender systems. In *Proceedings of the 3rd annual conference on autonomous agents (1999)* (pp. 434–435). New York: ACM Press.
28. Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2007). Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, *171*(5–6), 255–285.
29. Lang, J., Maudet, N., & Polukarov, M. (2013). New results on equilibria in strategic candidacy. In *Proceedings of the 6th international symposium on algorithmic game theory*. LNCS (October 2013) (Vol 8146, pp. 13–25). Berlin: Springer.
30. Liu, H., Feng, H., Zhu, D., & Luan, J. (2009). Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, *410*(27–29), 2746–2753.
31. Lu, T., & Boutilier, C. (2011). Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd international joint conference on artificial intelligence* (pp. 280–286).
32. Mattei, N., & Walsh, T. (2013). Preflib: A library for preferences. In *Proceedings of the 3rd international conference on algorithmic decision theory* (pp. 259–270).
33. Meir, R., Procaccia, A., Rosenschein, J., & Zohar, A. (2008). The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, *33*, 149–178.
34. Mirrlees, J. (1971). An exploration in the theory of optimal income taxation. *Review of Economic Studies*, *38*, 175–208.
35. Papadimitriou, C. (1994). *Computational complexity*. Reading: Addison-Wesley.
36. Parkes, D., & Xia, L. (2012). A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *Proceedings of the 26th AAAI conference on artificial intelligence (July 2012)* (pp. 1429–1435).
37. Persson, T., & Tabellini, G. (2000). *Political economics: Explaining economic policy*. Cambridge: MIT Press.
38. Polukarov, M., Obraztsova, S., Rabinovich, Z., Kruglyi, A., & Jennings, N. (2015). Convergence to equilibria in strategic candidacy. In *Proceedings of the 24th international joint conference on artificial intelligence*.
39. Roberts, K. (1977). Voting over income tax schedules. *Journal of Public Economics*, *8*(3), 329–340.
40. Rothe, J., Spakowski, H., & Vogel, J. (2003). Exact complexity of the winner problem for Young elections. *Theory of Computing Systems*, *36*(4), 375–386.
41. Rothstein, P. (1990). Order restricted preferences and majority rule. *Social Choice and Welfare*, *7*(4), 331–342.
42. Rothstein, P. (1991). Representative voter theorems. *Public Choice*, *72*(2–3), 193–212.
43. Saporiti, A., & Tohmé, F. (2006). Single-crossing, strategic voting and the median choice rule. *Social Choice and Welfare*, *26*(2), 363–383.
44. Skowron, P., Yu, L., Faliszewski, P., & Elkind, E. (2013). The complexity of fully proportional representation for single-crossing electorates. In *Proceedings of the 6th international symposium on algorithmic game theory* (pp. 1–12).
45. Walsh, T. (2007). Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI conference on artificial intelligence (July 2007)* (pp. 3–8). Menlo Park: AAAI Press.
46. Young, H. (1977). Extending Condorcet's rule. *Journal of Economic Theory*, *16*(2), 335–353.