Completing the Puzzle: Solving Open Problems for Control in Elections

Gábor Erdélyi and **Christian Reger**
School of Economic Disciplines
University of Siegen
57072 Siegen, Germany

Yongjie Yang
School of Information Science and Engineering
Central South University
410083 Changsha, China

Chair of Economic Theory
Saarland University
66123 Saarbrücken, Germany

Abstract

We are investigating the computational complexity of electoral control in elections. Electoral control describes the scenario where the election chair seeks to alter the outcome of the election by structural changes such as adding, deleting, or replacing either candidates or voters. Such control actions have been studied in the literature for several prominent voting rules. In this paper, we complement those results by solving several open cases for Copeland$^\alpha$, Maximin, k-Veto, Plurality with Runoff, and Veto with Runoff.

1 Introduction

Since the seminal works of Bartholdi et al. (Bartholdi III, Tovey, and Trick 1989; 1992; Bartholdi III and Orlin 1991), many strategic voting problems have been proposed and studied from the complexity theoretic point of view. These strategic voting problems include *manipulation*, where voters cast their votes strategically, *bribery*, where an external agent changes some voters' votes, and *control*, where an external agent (usually called the chair) tries to alter the outcome of an election by structural changes such as adding, deleting, or replacing either candidates or voters. For a broad overview of these strategic actions, their applications in multiagent systems, recommender systems, ranking algorithms, etc., and for a survey of the related results we refer to the book chapters (Baumeister and Rothe 2015; Faliszewski and Rothe 2016) and the references cited therein.

In this paper, we will focus on control, in particular on adding, deleting, and replacing either candidates or voters. There is a long line of research centered on the complexity of control. Many voting rules have been investigated, such as Approval Voting and its variants, Condorcet, and Plurality (Bartholdi III, Tovey, and Trick 1992; Hemaspaandra, Hemaspaandra, and Rothe 2007; Betzler and Uhlmann 2009; Erdélyi, Nowak, and Rothe 2009; Erdélyi et al. 2015), Copeland (Faliszewski et al. 2009; Betzler and Uhlmann 2009), Maximin (Faliszewski, Hemaspaandra, and Hemaspaandra 2011; Maushagen and Rothe 2016), k-Approval, and k-Veto (Lin 2011; Loreggia et al. 2015). A general (multimode) control problem, allowing an external agent to perform different types of control actions at once such as deleting and/or adding voters and/or candidates, has been introduced in (Faliszewski, Hemaspaandra, and Hemaspaandra 2011). The reader may ask, why do we need yet another paper on the complexity of control? The answer is, because our knowledge on the complexity of control is incomplete, there are several voting rules for which we still have some unsolved open cases. In this paper, we are filling those gaps. In the following, we will highlight our contributions:

- Faliszewski et al. (2009) and Loreggia (2012) investigated the complexity of control in Copeland$^\alpha$ elections leaving the case destructive control by replacing voters for any α, $0 \leq \alpha \leq 1$ open. We solve this open problem.
- Faliszewski et al. (2011) and Maushagen and Rothe (2016) investigated the complexity of control in Maximin elections, leaving the cases constructive and destructive control by replacing either candidates or voters open. We solve these problems, moreover, we solve a more general problem called exact destructive control by adding and deleting candidates, a special form of multimode control.
- Lin (2011) and Loreggia et al. (2015) focused on control in k-Veto. Open cases are k-Veto constructive control by replacing voters for $k \geq 2$. We solve these open cases, providing a dichotomy result for k-Veto with respect to the values of k.
- Finally, we investigate the complexity of control for two common voting rules, which, surprisingly, have not been considered yet in the literature, namely Plurality and Veto with Runoff.

2 Preliminaries

An *election* E is given by a tuple $E = (C,V)$ where C is a finite set of *candidates* and V is a finite multiset of *voters*. Each vote is defined as a linear order over C, indicating the preference of the voter over C. In particular, if a voter $v \in V$ prefers candidate a to candidate b, denoted as $a \succ b$, a is ordered before b in v. A *voting correspondence* (or *voting rule*) τ maps each election (C,V) to a subset of candidates called the *winners* of the election. For two candidates $a,b \in C$, let $N_E(a,b)$ be the number of voters preferring a to b. For a vote $v \in V$, let $top(v)$ and $bot(v)$ denote the candidates ranked at the top and at the last position in v, respectively. Furthermore, for any set X of candidates or voters let n_X denote the cardinality of X. We consider the following voting rules.

- *Copeland$^\alpha$:* For each pairwise comparison between two candidates a and b, if $N_E(a,b) > N_E(b,a)$, a receives 1 point and b 0 points. If $N_E(a,b) = N_E(b,a)$, both a and b receive α points, where $\alpha \in [0,1]$. The candidates with the highest total points are the winners.
- *Maximin:* The Maximin score of a candidate a is defined as $\min_{b \in C \setminus \{a\}} N_E(a,b)$. Candidates with the highest Maximin score are the winners.
- *k-Approval:* Each voter gives 1 point to every candidate ranked on the top-k positions. The winners are the candidates with the highest total score. 1-Approval is often referred to as *Plurality*.
- *k-Veto:* Each voter gives 0 points (we also say vetoes) to every candidate ranked on the last k positions. The winners are the candidates with the least vetoes. 1-Veto is often referred to as *Veto*.
- *Plurality with Runoff:* Each voter only approves of his top-ranked candidate. If there is a candidate c who is approved by every voter, then c is the unique winner. Otherwise, this voting rule takes two stages to select the winner. In the first stage, all candidates except the ones who receive the most and second-most approvals are eliminated from the election. If more than two candidates remain, a tie-breaking rule is applied to select exactly two of the remaining candidates. Then, the remaining two candidates, say c and d, compete in the second stage (runoff stage). In particular, if $N_E(c,d) > N_E(d,c)$ then c wins; and if $N_E(d,c) > N_E(c,d)$ then d wins. Otherwise, a tie-breaking rule applies to determine the winner between c and d.
- *Veto with Runoff:* Each voter vetoes exactly one candidate. This voting rule is defined similar to Plurality with runoff, with a slight difference in the first stage. All candidates except the ones who receive the least and second-least vetoes are eliminated from the election.

A voting rule is said to be *unanimous* if the same candidate is ranked in the top position in all votes, this candidate wins.

In this paper, we consider various control problems which can be considered as special cases of the following problem.

τ-CONSTRUCTIVE MULTIMODE CONTROL	
Given:	An election $(C \cup D, V \cup W)$ with registered candidate set C, unregistered candidate set D, registered voter set V, unregistered voter set W, a designated candidate $c \in C$, and four non-negative integers $\ell_{AV}, \ell_{DV}, \ell_{AC}, \ell_{DC}$, with $\ell_{AV} \leq \|W\|$, $\ell_{DV} \leq \|V\|$, $\ell_{AC} \leq \|D\|$, and $\ell_{DC} \leq \|C\|$.
Question:	Are there $V' \subseteq V$, $W' \subseteq W$, $C' \subseteq C$, $D' \subseteq D$, with $\|V'\| \leq \ell_{DV}$, $\|W'\| \leq \ell_{AV}$, $\|C'\| \leq \ell_{DC}$, and $\|D'\| \leq \ell_{AC}$, such that c is a winner of the election $((C \setminus C') \cup D', (V \setminus V') \cup W')$ under voting rule τ?

In τ-Destructive Multimode Control we ask whether there exist subsets V', W', C', and D' such that c is not a winner in $((C \setminus C') \cup D', (V \setminus V') \cup W')$ under τ.

In this paper we consider several special cases of multimode control, such as adding, deleting, or replacing either candidates or voters. The following list gives an overview of the restrictions compared to the general multimode control problem:

Problems	Restrictions
Add. Voters	$\ell_{AC} = \ell_{DC} = \ell_{DV} = 0$ and $D = \emptyset$
Del. Voters	$\ell_{AC} = \ell_{DC} = \ell_{AV} = 0$ and $D = W = \emptyset$
Add. Candidates	$\ell_{DC} = \ell_{AV} = \ell_{DV} = 0$ and $W = \emptyset$
Del. Candidates	$\ell_{AC} = \ell_{AV} = \ell_{DV} = 0$ and $D = W = \emptyset$
Repl. Voters	$\|V'\| = \|W'\|$, $\ell_{AV} = \ell_{DV}$, $\ell_{AC} = \ell_{DC} = 0$ and $D = \emptyset$
Repl. Candidates	$\|C'\| = \|D'\|$, $\ell_{AC} = \ell_{DC}$, $\ell_{AV} = \ell_{DV} = 0$ and $W = \emptyset$

Throughout the paper, we will use a four-letter code for our problems. The first two characters CC/DC stand for constructive/destructive control, the third character A/D/R stands for adding/deleting/replacing, and the last one C/V for voters/candidates (for example, DCRV stands for destructive control by replacing voters). For simplicity, in each problem in the above table, we use ℓ to denote the integer(s) in the input that is not necessarily required to be 0. For example, when considering CCRV, we use ℓ to denote $\ell_{AV} = \ell_{DV}$.

We assume the reader to be familiar with basics in complexity theory, such as NP-completeness or the complexity class P. Our NP-hardness results are mainly based on reductions from the following NP-complete problem (Gonzalez 1985).

3-EXACT COVER BY 3-SETS (3-X3C)	
Given:	A set $U = \{u_1, \ldots, u_{3\kappa}\}$ and a collection $S = \{S_1, \ldots, S_n\}$ of 3-element subsets of U such that each $u_j \in U$ occurs in exactly three subsets $S_i \in S$. (We have $\|S\| = n = 3\kappa = \|U\|$.)
Question:	Does S contain an exact cover for U, i.e., a subcollection $S' \subseteq S$ such that every element of U occurs in exactly one member of S'?

The EXACT 3-SET COVER (X3C) problem is a generalization of 3-X3C where the elements in U do not necessarily occur exactly in three subsets of S.

For showing membership in P, we will in some proofs make use of the following problem introduced by (Grötschel, Lovász, and Schrijver 1988) (for membership in P, see also (Gabow 1983)).

GENERALIZED WEIGHTED b-EDGE MATCHING	
Given:	An undirected multigraph $G = (V,E)$ without loops, capacity functions $b_l, b_u : V \to \mathbb{N}_0$, $c_l, c_u : E \to \mathbb{N}_0$ and a weight function $w : E \to \mathbb{N}_0$.
Question:	Can we find a function $x : E \to \mathbb{N}_0$ that maximizes $\sum_{e \in E} w(e)x(e)$ such that $c_l(e) \leq x(e) \leq c_u(e)$ holds for each edge $e \in E$ and $b_l(v) \leq \sum_{e \in E \cap \delta(v)} x(e) \leq b_u(v)$ holds for each node $v \in V$? ($\delta(v)$ is the set of edges incident to node v.)

Important special cases are *b-edge matching* and *b-edge cover*. For both problems, we have $c_l(e) = 0$ and $c_u(e)$ as the number of appearances of edge e. For the matching problem, we have $b_l(v) = 0$ and $w(e) = 1$ for each $v \in V, e \in E$.[1] In contrast, we have $b_u(v) = \infty$ and $w(e) = -1$ for the cover problem. The weights are negative as we want to minimize the number of edges for the cover problem. Conversely we seek to maximize the number of edges for the matching problem. Throughout this paper, we will say *generalized b-edge matching* and *generalized b-edge cover* and use these notions interchangeably with *(b-)edge matching* and *(b-)edge cover*, respectively, for practical reasons (when there is no way of confusion). These problems are defined in a way that the number of edges is maximized or minimized (i.e., $w(e) = 1$ or -1), but both vertices and edges may have both upper and lower capacity constraints.

3 Results

Copeland$^\alpha$. We start by completing our knowledge on control in Copeland$^\alpha$ elections. Faliszewski et al. (2009) and Loreggia (2012) investigated the complexity of control in Copeland$^\alpha$ elections, leaving the cases destructive control by replacing voters and constructive and destructive control by replacing candidates open. A voting rule satisfies *Insensitive to Bottom-ranked Candidates (IBC)* if the winners do not change after a subset of candidates which are ranked after all other candidates in all votes are deleted. Note that both Copeland$^\alpha$ and Maximin satisfy IBC. Loreggia et al. (2015; 2016) established the following relationship between CCRC and CCDC, and between DCRC and DCDC.

Lemma 1. *(Loreggia 2016; Loreggia et al. 2015) Let τ be a voting rule satisfying IBC. Then, τ-*CCRC *is* NP*-hard if τ-*CCDC *is* NP*-hard, and τ-*DCRC *is* NP*-hard if τ-*DCDC *is* NP*-hard.*

Due to Lemma 1 and the facts that Copeland$^\alpha$ satisfies IBC, and Copeland$^\alpha$-CCDC and Copeland$^\alpha$-DCDC are NP-hard (Faliszewski et al. 2009), it follows that Copeland$^\alpha$-CCRC and Copeland$^\alpha$-DCRC are NP-hard.

It remains the case of destructive control by replacing voters.

Theorem 2. Copeland$^\alpha$-DCRV *is* NP*-complete for any α, $0 \le \alpha \le 1$.*

We omit the proof due to space constraints. The proof is a slight modification of the Copeland$^\alpha$-CCAV proof in (Faliszewski et al. 2009) with the only difference that there are k further votes ranking the designated candidate first. These votes are replaced with the highest priority.

Maximin. Let us now turn to Maximin. Faliszewski et al. (2011) have already investigated the complexity of constructive and destructive control by adding and deleting either candidates or voters. We will complete the picture on control in Maximin elections by providing results on constructive and destructive control by replacing candidates and voters. It is known that constructive control by deleting candidates for Maximin is polynomial-time solvable (Faliszewski, Hemaspaandra, and Hemaspaandra 2011). Hence, we could not obtain the NP-hardness of MAXIMIN-CCRC from Lemma 1. However, Loreggia (Loreggia 2016) introduced another useful lemma.

Lemma 3. *(Loreggia 2016) Let τ be an unanimous voting rule that satisfies IBC. If τ-*CCAC *is* NP*-hard, then τ-*CCRC *is* NP*-hard.*

Due to this lemma and the facts that (1) Maximin is unanimous; (2) Maximin satisfies IBC; and (3) CCAC for Maximin is NP-complete (Faliszewski, Hemaspaandra, and Hemaspaandra 2011), we know that MAXIMIN-CCRC is NP-complete.

The following theorem handles constructive and destructive control by replacing voters. Our proof is a modification of the proof of constructive control by adding voters in Maximin (Faliszewski, Hemaspaandra, and Hemaspaandra 2011).

Theorem 4. MAXIMIN-CCRV *and* MAXIMIN-DCRV *are* NP*-complete.*

Proof Sketch. We start with the constructive case. Let (U,S) be a given X3C instance. We construct the following CCRV instance. Let the set of candidates be $C = U \cup \{c,d\}$ with the distinguished candidate c. The registered voter set consists of the following voters:

1. There are $3\kappa + 1$ voters of the form $d \succ U \succ c$,
2. there are κ voters of the form $c \succ U \succ d$, and
3. there are κ voters of the form $c \succ d \succ U$.

Moreover, for each $S_i \in S$, we create an unregistered vote in W of the form $(U \setminus S_i) \succ c \succ S_i \succ d$. Finally, we set $\ell = \kappa$, i.e., we are allowed to replace at most κ voters.

We claim that we can make candidate c the winner of the election by replacing up to κ voters if and only if S contains an exact cover for U. The argument for the correctness is similar to the one for MAXIMIN-CCAV in (Faliszewski, Hemaspaandra, and Hemaspaandra 2011).

The destructive version works identically, except that the first voter group contains only 3κ voters and the distinguished candidate is d.

It remains to show the complexity of destructive control by replacing candidates in Maximin. In contrast to the NP-hardness results for the other replacing cases, we show that Maximin-DCRC is polynomial-time solvable. In fact, we show the P-membership of a more general problem called *exact destructive control by adding and deleting candidates*, denoted by τ-EDCAC+DC. In particular, this problem is a variant of the multimode control problem where $\ell_{AV} = \ell_{DV} = 0, W = \emptyset$. Moreover, it must hold that in the solution $|C'| = \ell_{DC}$ and $|D'| = \ell_{AC}$ (i.e., the chair removes and adds an exact number of candidates). Note that the number of candidates added and the number of candidates removed have not to be the same.

Theorem 5. MAXIMIN-EDCAC+DC *is in* P.

[1] We could also use an arbitrary positive constant weight for each e, but for sake of simplicity we use 1.

Proof. Our input is an exact destructive control by adding and deleting candidates instance as defined above. Suppose that the chair adds exactly ℓ_{AC} candidates from D and deletes exactly ℓ_{DC} candidates from C. Note that $\ell_{DC} < |C|$ since the chair must not delete the despised candidate c. Our algorithm works as follows. It checks if there is a pivotal candidate $c' \neq c$ that beats c in the final election.

In case c has at most k points, there is some candidate $d \in (C \cup D) \setminus \{c\}$, not necessarily different from c' with $N(c,d) \leq k$. Our algorithm checks whether there is a final election including c, c' and d, c has at most k points and c' has at least $k+1$ points, where $k \in \{0,1,\ldots,|V|-1\}$. Note that we may restrict ourselves to values $k \leq \lceil \frac{|V|}{2} \rceil - 1$. Otherwise c does not lose any pairwise comparison and is a weak Condorcet winner and thus a Maximin winner, since no other candidate can do better.

Given a pair (c',k) with $c' \in (C \cup D) \setminus \{c\}$, our algorithm does the following:

- Compute $D(c') = \{d \in (C \cup D) \setminus \{c\} : N(c,d) \leq k \wedge (c' = d \vee N(c',d) > k)\}$. If $D(c') = \emptyset$ or $N(c',c) \leq k$, we immediately reject for the pair (c',k). Otherwise, we guess a candidate $d \in D(c')$ (not necessarily different from c'). d has the function to fix the score of c below k. In order to keep c''s score above the score of c, it must hold either $c' = d$ or $N(c',d) > k$. [2] Unless we abort, go to the next step.
- Check whether $\ell_{DC} \leq |C| - 1 - |C \cap \{c',d\}|$ and $\ell_{AC} \geq |D \cap \{c',d\}|$. Otherwise we reject because there is no way for the chair to keep (add) both c' and d in(to) the final election. If yes, proceed with the next step.
- Compute $C_1 = \{c'' \in C \setminus \{c,c',d\} : N(c',c'') \leq k\}$. Candidates in C_1 must all be deleted in order to keep the maximin score of c' higher than k. If $|C_1| > \ell_{DC}$, we discard this subcase and try the next triple (c',k,d). Otherwise, the chair deletes all candidates in C_1 and arbitrary other candidates in $C \setminus \{c,c',d\}$ such that exactly ℓ_{DC} candidates have been deleted. We go to the next step.
- Compute $D_1 = \{a \in D \setminus \{c',d\} : N(c',a) > k\}$. Candidates in D_1 are the only candidates which may be added and the score of c' does not decrease. Hence, if $|D_1| < \ell_{AC} - |D \cap \{c',d\}|$, we reject for the triple (c',k,d) since the chair must add some candidates leading to a lower score than $k+1$ for c'. Otherwise, we accept.

The correctness and polynomial running time immediately follow for this algorithm. ❑

Due to the Theorem 5, we obtain the following result.

Corollary 6. MAXIMIN-DCRC *is in* P.

We would like to point out that Faliszewski et al. (2011) showed that MAXIMIN-CCAC$_{\mathrm{U}}$+DC is polynomial-time solvable. In this case the chair is allowed to add as many unregistered candidates as he wants but can only delete a limited number of candidates.

[2] Note that if the Maximin score of c is less than k, c' can also beat c with k points, but this case is captured by another pair (c',k).

***k*-Veto.** Turning now to k-Veto, it is known that VETO-CCRV and k-VETO-DCRV for all possible k are polynomial-time solvable (Loreggia et al. 2015). We complement these results by showing that 2-VETO-CCRV is polynomial-time solvable and that k-VETO-CCRV is NP-complete for $k \geq 3$, achieving a dichotomy result for k-Veto with respect to the values of k.

Let V^c (W^c) be the set consisting of all voters in V (W) vetoing c, and define $V^{\neg c} = V \setminus V^c$ ($W^{\neg c} = W \setminus W^c$). For an election (C,V) and a candidate $c \in C$, let $vetoes_{(C,V)}(c)$ be the number of voters in V vetoing c.

Theorem 7. 2-VETO-CCRV *is in* P.

Proof. Our input is a constructive control by replacing voters instance as defined in the Preliminaries. Our algorithm distinguishes the following cases:

- $vetoes_{(C,V)}(c) \leq \min(\ell, n_W - vetoes_{(C,W)}(c))$: In this case, the algorithm returns "Yes" since c can be made a winner with zero vetoes by replacing all registered votes vetoing c with equal number of unregistered votes not vetoing c.
- $n_W - vetoes_{(C,W)}(c) \leq \min(\ell, vetoes_{(C,V)}(c))$: In this subcase, the chair definitely replaces $n_W - vetoes_{(C,W)}(c)$ voters in V vetoing c by the same number of voters from W not vetoing c. Possibly, he exchanges further $\ell - n_W + vetoes_{(C,W)}(c)$ V-voters vetoing c by W-voters vetoing c. Anyway, c has exactly $v_c = vetoes_{(C,V)}(c) - (n_W - vetoes_{(C,W)}(c)) = vetoes_{(C,V \cup W)}(c) - n_W$ vetoes in the final election. Since none of the voters in $V^{\neg c}$ are exchanged and since all voters from $W^{\neg c}$ join the election, we are searching for no more than v_c voters in $V^c \cup W^c$ that shall belong to the final election, such that at least $\max(0, vetoes_{(C,V)}(c) - \ell)$ among them belong to V^c (as the chair can replace at most ℓ voters in total) [3]. This leads to the following b-edge cover problem.
  Each candidate $d \neq c$ yields a vertex d. There are two vertices c_V and c_W representing vetoes that non-distinguished candidates receive from voters in V or W vetoing c, respectively. Each voter in V^c (W^c) vetoing d and c yields an edge between d and c_V (c_W). The capacities are $b(d) = v_c - vetoes_{(C,V^{\neg c} \cup W^{\neg c})}(d)$, $b(c_V) = \max(0, vetoes_{(C,V)}(c) - \ell)$, $b(c_W) = 0$. The capacities for the d take into account that d already receives $vetoes_{(C,V^{\neg c} \cup W^{\neg c})}(d)$ vetoes from voters not vetoing c, which are fixed in the final election.
  There is an edge cover with at most v_c edges if and only if c can be made a winner in the final election as then (and only then) the chair can add at most ℓ voters and as few as possible of these voters veto c. Notice that the capacities ensure that at least $\max(0, vetoes_{(C,V)}(c) - \ell)$ voters in V vetoing c remain in the final election (that is, at most ℓ voters are exchanged in total), and—provided that the matching has the correct size—that at most $\ell -$

[3] If $vetoes_{(C,V)}(c) \leq \ell$, the algorithm described below can be simplified to a simple greedy algorithm that – given candidate $d \neq c$ – tries to add (at least) $\delta(d) := \max(0, v_c - vetoes_{(C,V^{\neg c} \cup W^{\neg c})}(d))$ voters in $V^c \cup W^c$ vetoing d (and c) to the election, without considering more than v_c voters from $V^c \cup W^c$ in total.

$n_W + \mathit{vetoes}_{(C,W)}(c)$ voters are added from W^c (and hence at most ℓ in total).

- $\ell \leq \min(\mathit{vetoes}_{(C,V)}(c), n_W - \mathit{vetoes}_{(C,W)}(c))$: In this subcase, the chair exchanges ℓ voters in V vetoing c for ℓ voters from W not vetoing c.
  We may fix the voters in $V^{\neg c}$ in the final election as none of them is exchanged. Additionally we want to make sure that $\mathit{vetoes}_{(C,V)}(c) - \ell$ voters from V^c are in the final election, and the vetoes of (at most) ℓ voters from $W^{\neg c}$ are accounted for. This leads to the following b-edge cover problem with upper and lower vertex constraints.
  We are given the vertex set $\{c_V\} \cup (C \setminus \{c\})$. Each voter in V^c vetoing $d \neq c$ (and c) yields an edge $\{c_V, d\}$. Each voter in $W^{\neg c}$ vetoing d and e ($d \neq c \neq e \neq d$) yields an edge $\{d, e\}$. The capacities are $b_l(c_V) = b_u(c_V) = \mathit{vetoes}_{(C,V)}(c) - \ell$, $b_l(d) = \max(0, \mathit{vetoes}_{(C,V)}(c) - \ell - \mathit{vetoes}_{(C,V^{\neg c})}(d))$, and $b_u(d) = \infty$ ($d \in C \setminus \{c\}$).
  c can be made a winner by exchanging (exactly) ℓ voters in V vetoing c for ℓ voters from W not vetoing c if and only if there is a minimal edge cover with at most $\mathit{vetoes}_{(C,V)}(c)$ edges. The capacity constraints ensure that exactly $\mathit{vetoes}_{(C,V)}(c) - \ell$ voters vetoing c remain in V, and every non-distinguished candidate d receives the missing required number of vetoes either from the remaining voters in V^c or from voters added from $W^{\neg c}$. The voters in $V^{\neg c}$ and their vetoes assigned are fixed in advance. The size of the matching guarantees that ℓ added voters suffice to ensure that every $d \neq c$ has enough vetoes in the final election.

Each subcase can be calculated in polynomial time. Consequently, the overall algorithm terminates in polynomial time. ❑

We fill the complexity gap for CCRV for k-Veto by showing that k-VETO-CCRV is NP-complete for every $k \geq 3$. The proof is an adaption of the hardness proof of constructive control by adding voters for 3-Veto (Lin 2011). All problems considered in this paper are in NP. Hence, for NP-completeness results, we only prove NP-hardness.

Theorem 8. *k-VETO-CCRV is NP-complete for every constant $k \geq 3$.*

Proof. We show our result only for $k = 3$ and argue at the end of the proof how to handle the cases $k \geq 4$.

Our proof provides a reduction from X3C. Given an instance (U, S) of X3C, we construct an instance of 3-VETO-CCRV as follows. Let the candidate set be $C = \{c\} \cup \{d_1, d_2, d_3\} \cup U$, with the designated candidate c. The set V of registered voters consists of the following $2n - 2\kappa + 3\kappa n$ voters.

1. There are $n + \kappa$ voters vetoing c, d_1 and d_2,
2. there are n voters vetoing d_1, d_2 and d_3, and
3. for each j, $1 \leq j \leq 3\kappa$, there are $n - 1$ voters vetoing u_j and two arbitrary dummy candidates in $\{d_1, d_2, d_3\}$.

Note that with the registered voters c has $n + \kappa$ vetoes, each $u_j \in U$ has $n - 1$ vetoes and d_i, $i \in \{1, 2, 3\}$ has at least n vetoes. Let the set W of unregistered voters consists of the following n voters. For each $S_i \in S$, there is a voter vetoing the candidates in S_i. Finally, we are allowed to replace at most κ voters, i.e., $\ell = \kappa$.

We claim that c can be made a 3-Veto winner of the election by replacing at most κ voters if and only if an exact cover of U exists.

($\Leftarrow$:) Assume that U has an exact cover $S' \subseteq S$. After replacing the κ votes from W vetoing subsets $S_i \in S'$ with arbitrary κ voters in V vetoing c, c has $(n + \kappa) - \kappa = n$ vetoes, every $u \in U$ has $(n - 1) + 1 = n$ vetoes, and each of d_1, d_2, d_3 has at least n vetoes. Clearly, c becomes a winner.

($\Rightarrow$:) Assume that c can be made a 3-Veto winner of the election by replacing voters. Let $V' \subseteq V$ and $W' \subseteq W$ be the two sets such that $|V'| = |W'| = \kappa$ and c becomes a winner after the replacement. Note that no matter which voters are in W', there must be a candidate $u_j \in U$ who has at most n vetoes after the replacement. This implies that each voter in V' must veto c. As a result, c has $(n + \kappa) - \kappa = n$ vetoes after the replacement. This further implies that for each j, $1 \leq j \leq 3\kappa$, $u_j \in U$ there is at least one voter in W' who vetoes u_j. As $|W'| = \kappa$, due to the construction of W, the collection of 3-subsets corresponding to the κ voters in W' must form an exact 3-set cover.

To show the NP-hardness for $k \geq 4$, we can add $k - 3$ dummy candidates being vetoed by every vote. ❑

Plurality and Veto with Runoff. We now turn to the final two voting rules considered in this paper. Both voting rules are common voting rules, however, there are no results on control in Plurality and Veto with Runoff. We first show that CCAV/CCDV/CCRV for both Plurality and Veto with Runoff are polynomial-time solvable. Instead of showing the results separately one-by-one, we prove that a variant of multimode control, *exact constructive control by adding and deleting voters* (denoted by τ-ECCAV+DV), is polynomial-time solvable, where τ is Plurality with Runoff and Veto with Runoff. In this exact variant, we require that the number of added and deleted voters is exactly equal to the corresponding given integer, i.e., we require that $|V'| = \ell_{DV}$ and $|W'| = \ell_{AV}$. Moreover, we have $\ell_{AC} = \ell_{DC} = 0$ and $D = \emptyset$. Note that CCAV/CCDV/CCRV are polynomial-time many-one reducible to ECCAV+DV.

From now on, we assume that ties are broken in favor of candidate c. For an election (C, V) and a candidate $d \in C$, let $\mathit{score}_{(C,V)}(d)$ be the number of voters in V approving d. In the proof of the following theorem we will show membership in P by reducing the given problem to the polynomial-time solvable problem INTEGRAL MIN-COST FLOW (Ahuja, Magnanti, and Orlin 1993).

Theorem 9. *Both* PLURALITY WITH RUNOFF-ECCAV+DV *and* VETO WITH RUNOFF-ECCAV+DV *are in* P.

Proof. Due to space constraints, we only provide the algorithm for Plurality with Runoff. Our input is an exact constructive control by adding and deleting voters instance as defined above. Our algorithm guesses a candidate $d \in C$ and four integers $\ell^c_{AV}, \ell^d_{AV}, \ell^c_{DV}, \ell^d_{DV}$ such that $0 \leq \ell^c_X + \ell^d_X \leq \ell_X$ for $X \in \{\text{AV, DV}\}$. The guessed candidate d is supposed to be the one who competes with c in the runoff stage. Moreover, ℓ^c_{AV} (resp. ℓ^d_{AV}) is supposed to be the number of voters added from W that approve c (resp. d), and ℓ^c_{DV} (resp. ℓ^d_{DV}) is supposed to be the number of voters deleted from V that approve c (resp. d). Given such guessed candidate and integers, we determine whether we can add exactly ℓ_{AV} votes wherein ℓ^c_{AV} (resp. ℓ^d_{AV}) of them approve c (resp. d), and delete exactly ℓ_{DV} votes wherein ℓ^c_{DV} (resp. ℓ^d_{DV}) of them approve c (resp. d). Clearly, the original instance is a Yes-instance if and only if at least one guess leads to a Yes answer to the above question. We show how to find the answer to the above question in polynomial-time. We immediately discard the guess if one of the following conditions holds:

(1) $\ell^c_{DV} > score_{(C,V)}(c)$
(2) $\ell^d_{DV} > score_{(C,V)}(d)$
(3) $\ell^c_{AV} > score_{(C,W)}(c)$
(4) $\ell^d_{AV} > score_{(C,W)}(d)$

Assume that none of the above conditions holds. Then, the scores of c and d are fixed. In particular, the final score of $e \in \{c,d\}$, denoted by $score(e)$, is $score_{(C,V)}(e) + \ell^e_{AV} - \ell^e_{DV}$. Let $s = \min\{score(c), score(d)\}$. To ensure c and d to be in the runoff, each candidate $a \in C \setminus \{c,d\}$ may have at most s points in total. A second condition for c to be a runoff winner against d is that c wins d in the pairwise comparison between them. Since there are $n' = n_V + \ell_{AV} - \ell_{DV}$ voters in the final election (C,V'), c must win at least $\lceil \frac{n'}{2} \rceil$ duels against d or, equivalently speaking, at most $\lfloor \frac{n'}{2} \rfloor$ comparisons may be won by d. Let $A = C \setminus \{c,d\}$ and $score_{(C,V)}(A) = \sum_{a \in A} score_{(C,V)}(a)$. Moreover, for $X \in \{AV, DV\}$, let $\ell^A_X = \ell_X - \ell^c_X - \ell^d_X$. As d in turn wins $score(d)$ comparisons against c in all votes where d is the top candidate, there may be at most $\lfloor \frac{n'}{2} \rfloor - score(d)$ voters favouring some $a \in A$ and preferring d to c in the final election. Hence, if this number is negative, we immediately reject for the current guess and regard the next one. Otherwise, we search for exactly $\underbrace{n_V - score_{(C,V)}(c) - score_{(C,V)}(d)}_{=score_{(C,V)}(A)} - \ell^A_{DV}$ voters in V not deleted and voting for candidates in A and for exactly ℓ^A_{AV} voters added from W and preferring some $a \in A$ such that the final election contains at most $\lfloor \frac{n'}{2} \rfloor - score(d)$ voters who rank some $a \in A$ first and prefer d over c. This leads to the following mincost flow problem.

There is a source x, a sink y, and two nodes V^A and W^A. Moreover, each voter in $V \cup W$ favoring some $a \in A$ yields a node. Moreover, each $a \in A$ yields a node a. If not mentioned other, each cost is equal to zero. There is an arc from x to V^A with capacity $score_{(C,V)}(A) - \ell^A_{DV}$. There is another arc from x to W^A with capacity ℓ^A_{AV}. Each voter $v \in V$ with top candidate in A yields an arc (V^A, v) with capacity 1. The cost of this arc is equal to one if and only if v prefers d to c. Analogously we define edges from W^A to w corresponding to voters in W favoring some $a \in A$. There is an edge from v to a with capacity one if and only if v prefers a most (where $v \in V \cup W$). Each $a \in A$ yields an arc (a, y) with capacity s.

There is a (maximum) flow with value $score_{(C,V)}(A) - \ell^A_{DV} + \ell^A_{AV}$ and (minimum) cost of at most $\lfloor \frac{n'}{2} \rfloor - score(d)$ since then and only then our algorithm can find exactly $score_{(C,V)}(A) - \ell^A_{DV}$ (remaining) voters in V voting for some $a \in A$ and exactly ℓ^A_{AV} A-voters added from W, each a has no more points than c or d, and (since each pairwise comparison of c against d costs one unit) a weak majority of voters prefers c to d in the final election. ❑

The exact versions of the destructive multimode control for Plurality and Veto with Runoff are polynomial-time solvable too. We omit the proofs.

Theorem 10. *Both* PLURALITY WITH RUNOFF-EDCAV+DV *and* VETO WITH RUNOFF-EDCAV+DV *are in* P.

Given the above results, we obtain the following corollary.

Corollary 11. PLURALITY WITH RUNOFF-Y *and* VETO WITH RUNOFF-Y *are in* P *for all* $Y \in \{CCAV, CCDV, CCRV, DCAV, DCDV, DCRV\}$.

Consider now candidate control. We show that candidate control for Plurality and Veto with Runoff are generally NP-complete.

Theorem 12. PLURALITY WITH RUNOFF-CCAC, VETO WITH RUNOFF-CCAC, PLURALITY WITH RUNOFF-DCAC, *and* VETO WITH RUNOFF-DCAC *are* NP-*complete.*

Proof. Due to space constraints, we only prove VETO WITH RUNOFF-CCAC. We prove the claim by a reduction from the 3-X3C problem. For a given 3-X3C instance (U,S), we create the following instance.

Let $C = U \cup \{c,q\}$ be the set of registered candidates, S be the set of unregistered candidates (for each $S_i \in S$ we create a candidate denoted by the same symbol), and c the distinguished candidate. Let V be the set of voters consisting of the following voters.

1. There is a voter of the form $S \succ U \succ c \succ q$,
2. for each i, $1 \le i \le n$, there are $3\kappa + 1$ voters of the form $q \succ U \succ c \succ S \setminus \{S_i\} \succ S_i$,
3. for each i, $1 \le i \le n$, there are $3\kappa + 1$ voters of the form $c \succ U \succ q \succ S \setminus \{S_i\} \succ S_i$,
4. for each j, $1 \le j \le 3\kappa$, there are $6\kappa - 3$ voters of the form $c \succ q \succ S \succ U \setminus \{u_j\} \succ u_j$,
5. for each i, $1 \le i \le n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are three voters of the form $q \succ U \succ S \setminus \{S_i\} \succ c \succ S_i$,
6. for each i, $1 \le i \le n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are two voters of the form $c \succ q \succ U \setminus \{u_{i_1}\} \succ S \setminus \{S_i\} \succ u_{i_1} \succ S_i$,
7. for each i, $1 \le i \le n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are two voters of the form $c \succ q \succ U \setminus \{u_{i_2}\} \succ S \setminus \{S_i\} \succ u_{i_2} \succ S_i$, and
8. for each i, $1 \le i \le n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are two voters of the form $c \succ q \succ U \setminus \{u_{i_3}\} \succ S \setminus \{S_i\} \succ u_{i_3} \succ S_i$.

We are allowed to add at most κ candidates, i.e, $\ell = \kappa$. Note that candidate c has $3\kappa(3\kappa+1)+9\kappa$ vetoes, q has $3\kappa(3\kappa+1)+1$ vetoes, and for every j, $1 \leq j \leq 3\kappa$, u_j has $6\kappa+3$ vetoes. Hence, c is not a Veto with Runoff winner of the election.

We claim that c can be made a Veto with Runoff winner by adding at most κ candidates if and only if an exact cover of U exists.

($\Leftarrow$:) Assume that there is an exact 3-set cover S' of U. After adding the candidates in S', candidate q has 1 veto, for every i, $1 \leq i \leq n$, S_i has at least $(3\kappa+1)+(3\kappa+1) = 6\kappa+2$ vetoes, for every j, $1 \leq j \leq 3\kappa$, u_j has $6\kappa+3-2 = 6\kappa+1$ vetoes, and c has 6κ vetoes. Hence, q and c move into the runoff stage. As more voters prefer c over q than the other way round, c becomes a final winner.

($\Rightarrow$:) Assume that adding $S' \subseteq S$ from the unregistered set of candidates to the election makes c a winner under Veto with Runoff. Observe first that S' must contain exactly κ candidates, since otherwise c would have at least $6\kappa+3$ vetoes, while at least one candidate in U would have at most $6\kappa+3-2 = 6\kappa+1$ vetoes. Hence, this candidate and q would be the two candidates going to the runoff stage. So, we have $|S'| = \kappa$. It follows that c has 6κ vetoes after adding candidates in S'. If S' is not an exact 3-set cover, there must be a candidate $u_j \in U$ occurring in at least two subsets of S'. Then, this candidate has at most $6\kappa+3-4 = 6\kappa-1$ vetoes, leading q and this candidate to be the ones competing in the runoff stage. So, we can conclude that S' is an exact 3-set cover. ❑

Theorem 13. Plurality with Runoff-CCDC, Veto with Runoff-CCDC, Plurality with Runoff-DCDC, *and* Veto with Runoff-DCDC *are* NP-*complete.*

Proof. Due to space constraints, we only prove Plurality with Runoff-CCDC. We prove the claim by a reduction from the 3-X3C problem. For a given 3-X3C instance (U,S), we create the following instance. Let $C = \{c,q\} \cup U \cup S$ be the set of candidates and c the distinguished candidate. Let V be the set of voters consisting of the following $9\kappa^2+21\kappa+1$ voters.

1. There are 2κ voters of the form $q \succ u_1 \succ u_2 \succ \cdots \succ u_{3\kappa} \succ S \succ c$,
2. there are $\kappa+1$ voters of the form $q \succ u_{3\kappa} \succ u_{3\kappa-1} \succ \cdots \succ u_1 \succ S \succ c$,
3. for each j, $1 \leq j \leq 3\kappa$, there are $3\kappa-3$ voters of the form $u_j \succ C \setminus \{c,q,u_j\} \succ c \succ q$,
4. for each i, $1 \leq i \leq n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are three voters of the form $S_i \succ c \succ C \setminus (S_i \cup \{c,q\}) \succ q$,
5. for each i, $1 \leq i \leq n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are two voters of the form $S_i \succ u_{i_1} \succ C \setminus \{c,q,u_{i_1}\} \succ c \succ q$,
6. for each i, $1 \leq i \leq n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are two voters of the form $S_i \succ u_{i_2} \succ C \setminus \{c,q,u_{i_2}\} \succ c \succ q$, and
7. for each i, $1 \leq i \leq n$, with $S_i = \{u_{i_1}, u_{i_2}, u_{i_3}\} \in S$, there are two voters of the form $S_i \succ u_{i_3} \succ C \setminus \{c,q,u_{i_3}\} \succ c \succ q$.

Furthermore, let $\ell_{DC} = \kappa$.

We claim that c can be made a Plurality with Runoff winner by deleting at most κ candidates if and only if an exact cover of U exists.

($\Leftarrow$:) Assume there is an exact 3-set cover S'. After deleting the candidates in S', q has $2\kappa+\kappa+1 = 3\kappa+1$ approvals, c has 3κ approvals, every remaining $S_i \in S$ has 9 approvals and for every j, $1 \leq j \leq 3\kappa$, u_j has $3\kappa-3+2 = 3\kappa-1$ approvals. Hence, q and c go to the runoff stage, leading c to be the final winner.

($\Rightarrow$:) Assume that it is possible to make c a Plurality with Runoff winner of the election by deleting at most κ candidates. Let us denote these candidates by C'. Note that $q \notin C'$, since otherwise there would be two candidates in U receiving at least $3\kappa-3+2\kappa = 5\kappa-3$ and $3\kappa-3+\kappa+1 = 4\kappa-2$ approvals, preventing c from winning. Furthermore, none of the candidates in U can be deleted, i.e., $U \cap C' = \emptyset$. In fact, if we delete a candidate $u_j \in U$, then the candidate ranked immediately after u_j in the $3\kappa-3$ votes created for u_j would receive at least $(3\kappa-3)+(3\kappa-3) = 6\kappa-6$ approvals, preventing c from going to the runoff stage. This means that the deletion of one candidate in U invites the deletion of all candidates in U, to make c the winner. However, we are allowed to delete at most κ candidates. Therefore, we have $C' \subseteq S$. After deleting the candidates in C', c has $3|C'|$ approvals. Note that $|C'| = \kappa$ must hold, otherwise at least one candidate in U would receive more approvals than candidate c, after deleting all candidates in C', hence, this candidate and q would be the two candidates going to the runoff stage. In consequence, we know that c receives 3κ approvals after deleting all candidates in C'. If C' is not an exact 3-set cover, then there must be a candidate $u_j \in U$ who occurs in at least two subsets of C'. Due to the construction, this candidate receives at least $3\kappa-3+2+2 = 3\kappa+1$ approvals, implying that q and this candidate are the two candidates going to the runoff stage, contradicting that c is the final winner after deleting all candidates in C'. Thus, C' must be an exact 3-set cover. ❑

Note that all hardness result hold regardless of the tie-breaking rule used. It remains to show the replacing candidates cases. We omit the proof.

Theorem 14. Plurality with Runoff-CCRC, Veto with Runoff-CCRC, Plurality with Runoff-DCRC, *and* Veto with Runoff-DCRC *are* NP-*complete.*

4 Conclusion

We have investigated the computational complexity of control for Copeland$^{\alpha}$, Maximin, k-Veto, Plurality with Runoff, and Veto with Runoff, closing the gaps in the literature. We would like to point out that the complexity of partitioning either the set of candidates or the set of voters is still open for the voting rules Plurality with Runoff and Veto with Runoff.

References

Ahuja, R.; Magnanti, T.; and Orlin, J. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall.

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

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

Bartholdi III, J. J.; Tovey, C. A.; and Trick, M. A. 1992. How hard is it to control an election? *Mathematical and Computer Modelling* 16(8-9):27–40.

Baumeister, D., and Rothe, J. 2015. Preference aggregation by voting. In Rothe, J., ed., *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer-Verlag. Chapter 4, 197–325.

Betzler, N., and Uhlmann, J. 2009. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science* 410(52):5425–5442.

Erdélyi, G.; Fellows, M.; Rothe, J.; and Schend, L. 2015. Control complexity in Bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences* 81(4):632–660.

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

Faliszewski, P., and Rothe, J. 2016. Control and bribery in voting. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A., eds., *Handbook of Computational Social Choice*. Cambridge University Press. Chapter 7, 146–168.

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

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

Gabow, H. N. 1983. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, 448–456.

Gonzalez, T. F. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science* 38:293–306.

Grötschel, M.; Lovász, L.; and Schrijver, A. 1988. *Geometric Algorithms and Combinatorial Optimization*. Heidelberg: Springer.

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

Lin, A. 2011. The complexity of manipulating *k*-approval elections. In *Proceedings of the 3rd International Conference on Agents and Artificial Intelligence*, 212–218.

Loreggia, A.; Narodytska, N.; Rossi, F.; Venable, K.; and Walsh, T. 2015. Controlling elections by replacing candidates or votes. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, 1737–1738.

Loreggia, A. 2012. Iterative voting and multi-mode control in preference aggregation. Master's thesis, University of Padova.

Loreggia, A. 2016. *Iterative Voting, Control and Sentiment Analysis*. Ph.D. Dissertation, University of Padova.

Maushagen, C., and Rothe, J. 2016. Complexity of control by partitioning veto and maximin elections and of control by adding candidates to plurality elections. In *Proceedings of the 22nd European Conference on Artificial Intelligence*, 277–285.