# Towards Completing the Puzzle: Solving Open Problems for Control in Elections

Gábor Erdélyi
School of Mathematics and Statistics,
University of Canterbury
Christchurch, New Zealand
gabor.erdelyi@canterbury.ac.nz

Christian Reger
School of Economic Disciplines,
University of Siegen
Siegen, Germany
christian.reger@ymail.com

Yongjie Yang*
[1]Chair of Economic Theory,
Saarland University
Saarbrücken, Germany
[2]Central South University
Changsha, China
yyongjiecs@gmail.com

## ABSTRACT

We investigate 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 a lot of prominent voting correspondences. 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.

## KEYWORDS

voting control; complexity; veto with runoff; plurality with runoff

**ACM Reference Format:**
Gábor Erdélyi, Christian Reger, and Yongjie Yang. 2019. Towards Completing the Puzzle: Solving Open Problems for Control in Elections. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

## 1 INTRODUCTION

Since the seminal works of Bartholdi et al. [5–7], 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 [8, 16] 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 correspondences have been investigated, such as Approval Voting and its variants, Condorcet, Plurality [7, 9, 12, 13, 21], Copeland [9, 14], Maximin [15, 28, 34], $k$-Veto, and $k$-Approval [23, 26, 35]. 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 [15]. 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 correspondences 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. [14] and Loreggia [24] investigated the complexity of control in Copeland$^\alpha$ elections leaving the case destructive control by replacing voters for any rational $\alpha$ where $0 \leq \alpha \leq 1$ open. We settle this open problem.
- Faliszewski et al. [15] and Maushagen and Rothe [28] 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 also solve a more general problem called exact destructive control by adding and deleting candidates, a special form of multimode control.
- Lin [23] and Loreggia et al. [26] 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 correspondences, which, surprisingly, have not been considered yet in the literature, namely Plurality with Runoff 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 *votes*. 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*) $\tau$ 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 $b$. We drop $E$ from the notation if it is clear from the context. Furthermore, for any set $X$ of candidates or voters let $n_X$ denote the cardinality of $X$. For ease of exposition, in this paper we exchangeably use the words vote and voter. We consider the following voting correspondences.

**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 $b$ receives 0 points. If $N_E(a, b) = N_E(b, a)$, both $a$ and $b$

*corresponding author

*Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.* 

receive $\alpha$ points, where $\alpha \in [0, 1]$ is a rational number. The candidates with the highest total points are the winners.

**Maximin** The Maximin score of a candidate $a \in C$ 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 on the top-$k$ positions. The winners are the candidates with the highest total score. Particularly, 1-Approval is often referred to as *Plurality* in the literature.

$k$-**Veto** Each voter gives 1 point (veto) 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 (PRun)** 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 correspondence takes two stages to select the winner. In the first stage, all candidates except the two who respectively receive the most and second-most approvals are eliminated from the election. If more than two candidates have the same highest total approvals, a tie-breaking rule is applied to select exactly two of them, and if there is one candidate with the most approvals but several candidates with the second-most approvals, a tie-breaking rule is used to select exactly one of those with the second-most approvals. 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 (VRun)** Each voter vetoes exactly the last-ranked candidate. This voting correspondence is defined similar to PRun, 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.

We study various control problems which can be considered as special cases of the following problem [15].

| $\tau$-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 $V' \subseteq V$, $W' \subseteq W$, $C' \subseteq C \setminus \{c\}$, $D' \subseteq D$, such that $|V'| \leq \ell_{DV}$, $|W'| \leq \ell_{AV}$, $|C'| \leq \ell_{DC}$, $|D'| \leq \ell_{AC}$, and $c$ is a winner of the election $((C \setminus C') \cup D', (V \setminus V') \cup W')$ under $\tau$? |

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

In this paper we study several special cases or restricted versions 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 | $\ell_{AC} = \ell_{DC} = \ell_{DV} = 0$, $D = \emptyset$ |
| Del. Voters | $\ell_{AC} = \ell_{DC} = \ell_{AV} = 0$, $D = W = \emptyset$ |
| Add. Candidates | $\ell_{DC} = \ell_{AV} = \ell_{DV} = 0$, $W = \emptyset$ |
| Del. Candidates | $\ell_{AC} = \ell_{AV} = \ell_{DV} = 0$, $D = W = \emptyset$ |
| Repl. Voters | $\|V'\| = \|W'\|$, $\ell_{AV} = \ell_{DV}$, $\ell_{AC} = \ell_{DC} = 0$, $D = \emptyset$ |
| Repl. Candidates | $\|C'\| = \|D'\|$, $\ell_{AC} = \ell_{DC}$, $\ell_{AV} = \ell_{DV} = 0$, $W = \emptyset$ |

Throughout the paper, we will use a four-letter code to denote 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 V/C for voters/candidates. For example, DCRV stands for destructive control by replacing voters. For simplicity, in each problem in the above table, we use $\ell$ to denote the integer(s) in the input that is not necessarily required to be 0. For example, when considering CCRV, we use $\ell$ to denote $\ell_{AV} = \ell_{DV}$. Control by adding/deleting votes/candidates has been extensively studied in the literature since the seminal work of Bartholdi (see, e.g., [7, 11, 22, 27, 29, 31, 33, 35]). However, the complexity of control by replacing candidates or voters has been studied just recently by Loreggia et al. [26].

We assume the reader to be familiar with basics in complexity theory, such as P, NP, NP-hard, NP-complete, etc. We refer to [32] for a concise introduction to the complexity theory and [2, 18] for more comprehensive discussions. For NP-completeness results we only provide the hardness proofs since the NP-membership of these problems is easy to check. Our NP-hardness results are mainly based on reductions from the following NP-hard problem [19].

| Restricted Exact Cover By 3-Sets (RX3C) | |
|---|---|
| **Given:** | A set $U = \{u_1, \ldots, u_{3\kappa}\}$ and a collection $\mathcal{S} = \{S_1, \ldots, S_{3\kappa}\}$ of 3-element subsets of $U$ such that each $u \in U$ occurs in exactly three subsets $S \in \mathcal{S}$. |
| **Question:** | Does $\mathcal{S}$ contain an exact 3-set cover for $U$, i.e., a subcollection $\mathcal{S}' \subseteq \mathcal{S}$ such that every element of $U$ occurs in exactly one member of $\mathcal{S}'$? |

For showing membership in P, we will in some proofs make use of the following problems. We also assume that the reader is familiar with graph theory (cf. [4, 10]).

| Integral Minimum Cost Flow (IMCF) | |
|---|---|
| **Given:** | A network $G = (V, E)$, capacity functions $b_\alpha, b_\beta : E \rightarrow \mathbb{N}_0$, a source vertex $x \in V$, a sink vertex $y \in V \setminus \{x\}$, a cost function $g : E \rightarrow N_0$, and an integer $r$. |
| **Question:** | Find a minimum cost flow from $x$ to $y$ of value $r$. Recall that a flow $f$ is a function assigning to each arc $(u, v) \in E$ an integer number $f(u, v)$ such that (1) $b_\alpha(u, v) \leq f(u, v) \leq b_\beta(u, v)$; and (2) for every node $v$ except $x$ and $y$, it holds that $\sum_{(u,v) \in E} f(u, v) = \sum_{(v,u) \in E} f(v, u)$. [1] The cost of a flow $f$ is $\sum_{(u,v) \in E} f(u, v) \cdot g(u, v)$, and the value of $f$ is $\sum_{(x,v) \in E} f(x, v)$. |

[1]For simplicity, we write $b_\alpha(u, v)$ for $b_\alpha((u, v))$, $b_\beta(u, v)$ for $b_\beta((u, v))$, and $g(u, v)$ for $g((u, v))$ throughout this paper.

In the above definitions, $b_\alpha$ and $b_\beta$ are respectively called the lower-bound capacity and the upper-bound capacity. The IMCF problem is well-known to be polynomial-time solvable [1].

| $b$-Edge Cover ($b$-EC) | |
|---|---|
| **Given:** | An undirected multigraph $G = (V, E)$ without loops, two capacity functions $b_\alpha, b_\beta : V \to \mathbb{N}_0$, and an integer $r$. |
| **Question:** | Is there a $b$-edge cover in $G$ of size at most $r$, i.e., a subset $E' \subseteq E$ of at most $r$ edges such that each node $v \in V$ is incident to at least $b_\alpha(v)$ and at most $b_\beta(v)$ edges in $E'$? |

The $b$-EC problem is also polynomial-time solvable [17, 20].

## 3 COPELAND$^\alpha$

We start by completing our knowledge on control in Copeland$^\alpha$ elections. Faliszewski et al. [14] and Loreggia [24] 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 correspondence satisfies *Insensitivity to Bottom-ranked Candidates (IBC)* if for any election with at least two candidates, 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. [25, 26] established the following relationship between CCRC and CCDC, and between DCRC and DCDC.

Lemma 3.1. *[25, 26] Let $\tau$ be a voting correspondence satisfying IBC. Then, $\tau$-CCRC is NP-hard if $\tau$-CCDC is NP-hard, and $\tau$-DCRC is NP-hard if $\tau$-DCDC is NP-hard.*

Due to Lemma 3.1 and the facts that Copeland$^\alpha$ satisfies IBC, and Copeland$^\alpha$-CCDC and Copeland$^\alpha$-DCDC are NP-hard [14], it follows that Copeland$^\alpha$-CCRC and Copeland$^\alpha$-DCRC are NP-hard.

It remains the case of destructive control by replacing voters. We complete it via the following theorem.

Theorem 3.2. *Copeland$^\alpha$-DCRV is NP-complete for any $\alpha$ such that $0 \leq \alpha \leq 1$.*

We omit the proof due to space constraints. The proof is a slight modification of the one for Copeland$^\alpha$-CCAV given in [14] with the only difference that there are $k$ further votes ranking the designated candidate first. These votes are replaced with the highest priority.

## 4 MAXIMIN

Let us now turn to Maximin. Faliszewski et al. [15] 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 [15]. Hence, we could not obtain the NP-hardness of Maximin-CCRC from Lemma 3.1. However, Loreggia [25] introduced another useful lemma.

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

Lemma 4.1. *[25] Let $\tau$ be an unanimous voting correspondence that satisfies IBC. If $\tau$-CCAC is NP-hard, then $\tau$-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 [15], 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 [15]. In the following, for two subsets $A$ and $B$ of candidates and a linear order $\succ$ over candidates, $A \succ B$ means that $a \succ b$ for every $a \in A$ and $b \in B$.

Theorem 4.2. *Maximin-CCRV and Maximin-DCRV are NP-complete.*

Proof. We start with the constructive case. Let $(U, \mathcal{S})$ be a given RX3C instance such that $|U| = |\mathcal{S}| = 3\kappa$. We construct the following CCRV instance. Let the set of candidates be $C = U \cup \{c, d\}$ s.t. $\{c, d\} \cap U = \emptyset$. The designated candidate is $c$. The registered votes are as follows.

- There are $3\kappa + 1$ votes of the form $d \succ U \succ c$.
- There are $\kappa$ votes of the form $c \succ U \succ d$.
- There are $\kappa$ votes of the form $c \succ d \succ U$.

Moreover, for each $S \in \mathcal{S}$, we create an unregistered vote in $W$ of the form $(U \setminus S) \succ c \succ S \succ d$. Finally, we set $\ell = \kappa$, i.e., we are allowed to replace at most $\kappa$ voters. We claim that we can make the candidate $c$ the winner of the election by replacing up to $\kappa$ voters if and only if $\mathcal{S}$ contains an exact 3-set cover for $U$. The argument for the correctness is similar to the one for Maximin-CCAV in [15].

The destructive version works identically, except that the first vote group contains only $3\kappa$ votes and the designated 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 $\tau$-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 deletes exactly $\ell_{DC}$ candidates and adds exactly $\ell_{AC}$ candidates). Note that the number of candidates added and the number of candidates deleted do not have to be the same.

Theorem 4.3. *Maximin-EDCAC+DC is in P.*

Proof. Our input is an EDCAC+DC instance as defined above. Suppose that the chair adds exactly $\ell_{AC}$ candidates from $D$ and deletes exactly $\ell_{DC}$ candidates from $C$. Note that $\ell_{DC} < |C|$ since the chair must not delete the designated 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 Maximin score at most $k$ for some integer $k$ in the final election, there exists 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 $d$, the candidate $c$ has Maximin score at most $k$ and $c'$ has Maximin score at least $k + 1$, where $k \in \{0, 1, \ldots, |V| - 1\}$. Note that we may restrict ourselves to values $k \leq \left\lceil \frac{|V|}{2} \right\rceil - 1$. Otherwise, $c$

does not lose any pairwise comparison and is a weak Condorcet winner and thus a Maximin winner.

Precisely, the algorithm first guesses the candidate $c' \in (C \cup D) \setminus \{c\}$ and the threshold score $k$ as discussed above, and then proceeds with the following steps.

(1) Let $D(c') = \{d \in (C \cup D) \setminus \{c\} : N(c,d) \leq k \wedge (c' = d \vee N(c',d) > k)\}$. 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 $c'$). The candidate $d$ has the function to fix the score of $c$ below or equal to $k$. In order to keep $c'$'s score above the score of $c$, it must hold either $c' = d$ or $N(c',d) > k$. [2] We go to the next step.

(2) Check whether $\ell_{\text{DC}} \leq |C| - 1 - |C \cap \{c',d\}|$ and $\ell_{\text{AC}} \geq |D \cap \{c',d\}|$. If this is the case, proceed with the next step. Otherwise, we reject because there is no way for the chair to keep (add) both $c'$ and $d$ in(to) the final election.

(3) Let $C_1 = \{c'' \in C \setminus \{c,c',d\} : N(c',c'') \leq k\}$. Candidates in $C_1$ must all be deleted in order to keep the maximin score of $c'$ higher than $k$. If $|C_1| > \ell_{\text{DC}}$, 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 $\ell_{\text{DC}}$ candidates have been deleted. We go to the next step.

(4) Let $D_1 = \{a \in D \setminus \{c',d\} : N(c',a) > k\}$. Candidates in $D_1$ are the only candidates which may be added and the score of $c'$ does not decrease. Hence, if $|D_1| < \ell_{\text{AC}} - |D \cap \{c',d\}|$, 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 original instance is a YES-instance, at least one guessed triple $(c',k,d)$ leads to a YES answer. The algorithm runs in polynomial time because there are polynomially many triples to check and each of them can be done in polynomial time as described above. □

Due to Theorem 4.3, we obtain the following result.

Corollary 4.4. *Maximin-DCRC is in P.*

Theorem 4.3 also generalizes the polynomial-time solvability results for Maximin-DCAC and Maximin-DCDC studied in [15]. We also would like to point out that Faliszewski et al. [15] showed that Maximin-CCAC$_{\text{U}}$+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.

## 5 $K$-VETO

Turning now to $k$-Veto, it is known that Veto-CCRV and $k$-Veto-DCRV for all possible $k$ are polynomial-time solvable [26]. We complement these results by showing that 2-Veto-CCRV is polynomial-time solvable and $k$-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 $vto_{(C,V)}(c)$ be the number of voters in $V$ vetoing $c$.

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

Theorem 5.1. *2-Veto-CCRV is in P.*

Proof. Let $(C, V \cup W)$, $\ell$, $c \in C$ be the components of the CCRV instance input as described in Section 2. Recall that $c$ is the designated candidate in the input. Our algorithm distinguishes the following cases

- $vto_{(C,V)}(c) \leq \min(\ell, n_W - vto_{(C,W)}(c))$.
  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 $c$.
- $n_W - vto_{(C,W)}(c) \leq \min(\ell, vto_{(C,V)}(c))$.
  In this case, the optimal choice for the chair is to replace $n_W - vto_{(C,W)}(c)$ voters in $V$ vetoing $c$ by the same number of voters from $W$ not vetoing $c$. Hence, all votes in $W^{\neg c}$ are ensured in the final election. In addition, all votes in $V^{\neg c}$ are also in the final election, as none of these votes needs to be exchanged in an optimal solution. However, the chair possibly needs to exchange further $\ell - n_W + vto_{(C,W)}(c)$ $V$-voters vetoing $c$ by the same number of $W$-voters vetoing $c$. Anyway, $c$ has exactly $v_c =$

$$vto_{(C,V)}(c) - (n_W - vto_{(C,W)}(c)) = vto_{(C,V \cup W)}(c) - n_W$$

  vetoes in the final election. Due to these observations, the question is equivalent to 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 $\max(0, vto_{(C,V)}(c) - \ell)$ and at most $vto_{(C,V)}(c) - n_W + vto_{(C,W)}(c)$ among them belong to $V^c$. We guess the exact number $\ell'$ where $\max(0, vto_{(C,V)}(c) - \ell) \leq \ell' \leq vto_{(C,V)}(c) - n_W + vto_{(C,W)}(c)$ of $V$-voters that are kept in the final election. This implies that we keep exactly $v_c - \ell'$ votes from $W^c$ in the final election. Clearly, if the given instance falls into this case and is a YES-instance, at least one of these guesses leads to a YES answer. In the following, we reduce the instance into an equivalent $b$-EC instance in polynomial time.
  For each candidate $d \in C \setminus \{c\}$, we create a vertex $d$. In addition, we create two vertices $c_V$ and $c_W$ representing vetoes that non-designated candidates receive from voters in $V$ or $W$ vetoing $c$, respectively. Each voter in $V^c$ ($W^c$) vetoing some candidate $d \in C \setminus \{c\}$ and $c$ yields an edge between $d$ and $c_V$ ($c_W$). The capacities are as follows.
  - $b_\alpha(c_V) = b_\beta(c_V) = \ell'$. These capacities ensure that exactly $\ell'$ votes from $V^c$ are kept in the final election.
  - $b_\alpha(c_W) = b_\beta(c_W) = v_c - \ell'$. These capacities ensure that exactly $v_c - \ell'$ votes from $W^c$ are kept in the final election.
  - $b_\beta(d) = |V \cup W|$ and $b_\alpha(d) = v_c - vto_{(C, V^{\neg c} \cup W^{\neg c})}(d)$ for every candidate $d \in C \setminus \{c\}$. As discussed above, all votes in $V^{\neg c} \cup W^{\neg c}$ are in the final elections. These votes give $vto_{(C, V^{\neg c} \cup W^{\neg c})}(d)$ vetoes to the candidate $d$. Hence, the lower-bound capacity for $d$ is to ensure that in the final election $d$ has at least the same number of vetoes as $c$. The upper-bound capacity for $d$ is not important and can be changed to any integer that is larger than the maximum possible vetoes the candidate $d$ can obtain.

It is fairly easy to check that there is a $b$-edge cover with at most $v_c$ edges if and only if $c$ can be made a winner in the final election by replacing exactly $vto_{(C,V)}(c) - \ell'$ votes.

- $\ell \leq \min(vto_{(C,V)}(c), n_W - vto_{(C,W)}(c))$.
  In this case, the optimal choice for the chair is to replace exactly $\ell$ voters in $V$ vetoing $c$ with $\ell$ voters from $W$ not vetoing $c$. In other words, we are ensured that the final election contains all voters in $V^{\neg c}$, exactly $vto_{(C,V)}(c) - \ell$ voters in $V^c$, and exactly $\ell$ voters from $W^{\neg c}$. This observation enables us to reduce the instance in this case to the following $b$-EC instance.
  The vertex set is $\{c_V\} \cup (C \setminus \{c\})$, i.e., we create a vertex $c_V$ first and then for each candidate in $C \setminus \{c\}$ we create a vertex denoted by the same symbol. For each voter in $V^c$ vetoing some $d \in C \setminus \{c\}$ (and $c$), we create an edge $(c_V, d)$. In addition, for each voter in $W^{\neg c}$ vetoing two distinct candidates $d$ and $e$, we create an edge $(d, e)$. The capacities of the vertices are as follows.
  - $b_\alpha(c_V) = b_\beta(c_V) = vto_{(C,V)}(c) - \ell$. This capacity ensures that exactly $vto_{(C,V)}(c) - \ell$ voters from $V^c$ remain in the final election.
  - For every $d \in C \setminus \{c\}$, we set $b_\beta(d) = |V \cup W|$ and
  $$b_\alpha(d) = \max\left(0, vto_{(C,V)}(c) - \ell - vto_{(C,V^{\neg c})}(d)\right).$$
    The lower bound ensures that in the final election $d$ has at least the same vetoes as $c$. Here, $vto_{(C,V^{\neg c})}(d)$ is the vetoes of $d$ obtained from voters in $V^{\neg c}$ which, as discussed above, are ensured in the final election. The upper bound is not very important and can be set as any integer larger than the maximum possible vetoes that $d$ can obtain in the final election.

  Given the above discussions, it is fairly easy to check that $c$ can be made a winner by replacing $\ell$ voters if and only if there is a $b$-edge cover of size at most $vto_{(C,V)}(c)$.

Each subcase can be done 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 [23].

Theorem 5.2. *$k$-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 the RX3C problem. Given an instance $(U, \mathcal{S})$ of the RX3C problem where $|U| = |\mathcal{S}| = 3\kappa$, we construct an instance of 3-Veto-CCRV as follows. Let the candidate set be $C = \{c\} \cup \{d_1, d_2, d_3\} \cup U$, where $\{c, d_1, d_2, d_3\} \cap U = \emptyset$. The designated candidate is $c$. For ease of exposition, let $n = 3\kappa$. The set $V$ consists of the following $2n - 2\kappa + 3\kappa \cdot n$ registered voters.

- There are $n + \kappa$ voters vetoing $c$, $d_1$ and $d_2$.
- There are $n$ voters vetoing $d_1$, $d_2$ and $d_3$.
- For each $u \in U$, there are $n - 1$ voters vetoing $u$ and any two arbitrary dummy candidates in $\{d_1, d_2, d_3\}$.

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

We claim that $c$ can be made a 3-Veto winner by replacing at most $\kappa$ voters if and only if an exact 3-set cover of $U$ exists.

($\Leftarrow$) Assume that $U$ has an exact 3-set cover $\mathcal{S}' \subseteq \mathcal{S}$. After replacing the $\kappa$ votes corresponding to $\mathcal{S}'$ from $W$ with arbitrary $\kappa$ voters in $V$ vetoing $c$, $c$ has $(n + \kappa) - \kappa = n$ vetoes, every $u \in U$ has $(n - 1) + 1 = n$ vetoes, and each $d_1$, $d_2$, and $d_3$ has at least $n$ vetoes. Clearly, $c$ becomes a winner.

($\Rightarrow$) Assume that $c$ can be made a 3-Veto winner by replacing at most $\ell$ voters. Let $V' \subseteq V$ and $W' \subseteq W$ be the two sets such that $|V'| = |W'|$ and $c$ becomes a winner after replacing all votes in $V'$ with all votes in $W'$. Observe first that $|V'|$ and $|W'|$ must be exactly $\kappa$, since otherwise $c$ has at least $n + 1$ vetoes and there exists one $u \in U$ having at most $n-1$ vetoes in the final election, contradicting that $c$ becomes a winner in the final election. In addition, no matter which $\kappa$ voters are in $W'$, there must be at least one candidate $u \in U$ 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 $u \in U$ there is at least one voter in $W'$ who vetoes $u$. As $|W'| = \kappa$, due to the construction of $W$, the collection of the 3-subsets corresponding to the $\kappa$ voters in $W'$ form an exact 3-set cover.

To show the NP-hardness for $k \geq 4$, we additionally create $k - 3$ dummy candidates being vetoed by every vote. The correctness argument is analogous. □

## 6 PLURALITY AND VETO WITH RUNOFF

We now turn to the final two voting correspondences considered in this paper. Both voting correspondences are common voting correspondences, however, somewhat surprisingly, there are no results on control in Plurality with Runoff and Veto with Runoff. We first show that CCAV/CCDV/CCRV for both Plurality with Runoff and Veto with Runoff are polynomial-time solvable when ties are broken in favor of the chair in both stages. Precisely, if several candidates are tied in the first stage, the chair has the right to select the two candidates who survive this stage, and if in the second stage $N_E(c, d) = N_E(d, c)$ for the two candidates $c$ and $d$ who survive the first stage, the chair is obligated to select the final winner between $c$ and $d$.

Instead of showing the results separately one-by-one, we prove that a variant of the multimode control problem, Exact Constructive Control by Adding and Deleting Voters, denoted by $\tau$-ECCAV+DV, is polynomial-time solvable, where $\tau$ 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 $|V'| = \ell_{\text{DV}}$ and $|W'| = \ell_{\text{AV}}$. Moreover, we have $\ell_{\text{AC}} = \ell_{\text{DC}} = 0$ and $D = \emptyset$. Note that CCAV/CCDV/CCRV are polynomial-time reducible to ECCAV+DV.

For an election $(C, V)$ and a candidate $d \in C$, let $score_{(C,V)}(d)$ be the number of voters in $V$ approving $d$. In the proof of the following theorem we will show the P-membership of PRun-ECCAV+DV and VRun-ECCAV+DV by reducing them to the polynomial-time solvable problem Integral Min-Cost Flow [1].

Theorem 6.1. *PRun-ECCAV+DV and VRun-ECCAV+DV are in P.*

Proof. Due to space constraints, we only provide the algorithm for Plurality with Runoff. Let $(C, V)$, $W$, $c \in C$, $\ell_{\text{AV}}$, and $\ell_{\text{DV}}$ be the components of the input of a given instance as described in the definition of $\tau$-ECCAV+DV. Here, $c$ is the designated candidate. Our algorithm guesses a candidate $d \in C \setminus \{c\}$ and four integers $\ell^c_{\text{AV}}$, $\ell^d_{\text{AV}}$, $\ell^c_{\text{DV}}$, and $\ell^d_{\text{DV}}$ such that $0 \leq \ell^c_X + \ell^d_X \leq \ell_X$ for $X \in \{\text{AV}, \text{DV}\}$. The guessed candidate $d$ is supposed to be the one who competes with $c$ in the runoff stage. Moreover, $\ell^c_{\text{AV}}$ (resp. $\ell^d_{\text{AV}}$) is supposed to be the number of voters added from $W$ that approve $c$ (resp. $d$), and $\ell^c_{\text{DV}}$ (resp. $\ell^d_{\text{DV}}$) 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 $\ell_{\text{AV}}$ votes from $W$ wherein $\ell^c_{\text{AV}}$ (resp. $\ell^d_{\text{AV}}$) of them approve $c$ (resp. $d$), and delete exactly $\ell_{\text{DV}}$ votes from $V$ wherein $\ell^c_{\text{DV}}$ (resp. $\ell^d_{\text{DV}}$) of them approve $c$ (resp. $d$). Clearly, the original instance is a YES-instance if and only if at least one of these guesses leads to a YES answer. We show how to find the answer to each subinstance in polynomial time. First, we immediately discard the guess if one of the following conditions holds: (1) $\ell^c_{\text{DV}} > score_{(C,V)}(c)$; (2) $\ell^d_{\text{DV}} > score_{(C,V)}(d)$; (3) $\ell^c_{\text{AV}} > score_{(C,W)}(c)$; or (4) $\ell^d_{\text{AV}} > score_{(C,W)}(d)$. Assume that none of the above conditions holds. Then, the scores of $c$ and $d$ in the final election are determined. Precisely, the final score of $e \in \{c, d\}$, denoted by $score(e)$, is $score_{(C,V)}(e) + \ell^e_{\text{AV}} - \ell^e_{\text{DV}}$. Let $s = \min\{score(c), score(d)\}$. To ensure $c$ and $d$ to be in the runoff stage, each candidate $a \in C \setminus \{c, d\}$ may have at most $s$ approvals in total. A second condition for $c$ to be a runoff winner against $d$ is that $c$ beats $d$ in the pairwise comparison between them. Since there are $n' = n_V + \ell_{\text{AV}} - \ell_{\text{DV}}$ voters in the final election $(C, V')$, $d$ must win at most $\left\lfloor \frac{n'}{2} \right\rfloor$ duels against $c$. Let $A = C \setminus \{c, d\}$ and $score_{(C,V)}(A) = \sum_{a \in A} score_{(C,V)}(a)$. Moreover, for $X \in \{\text{AV}, \text{DV}\}$, 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, if $\left\lfloor \frac{n'}{2} \right\rfloor < score(d)$, we reject for the current guess and regard the next one. Otherwise, we search for exactly

$$\underbrace{n_V - score_{(C,V)}(c) - score_{(C,V)}(d)}_{=score_{(C,V)}(A)} - \ell^A_{\text{DV}}$$

voters in $V$ not deleted and approving candidates in $A$, and exactly $\ell^A_{\text{AV}}$ voters added from $W$ and approving some $a \in A$ such that the final election contains at most $\left\lfloor \frac{n'}{2} \right\rfloor - score(d)$ voters who rank some $a \in A$ first and prefer $d$ over $c$. We solve this question by reducing it to the IMCF problem.

The construction of the IMCF instance is illustrated in Figure 1. Precisely, there is a source $x$, a sink $y$, and two nodes $V^A$ and $W^A$. Moreover, each voter in $V \cup W$ approving some $a \in A$ yields a node. Additionally, each $a \in A$ yields a node $a$. If not mentioned otherwise, each cost is equal to zero. There is an arc from $x$ to $V^A$ with lower-bound and upper-bound capacities $b_\alpha(x, V^A) = b_\beta(x, V^A) = score_{(C,V)}(A) - \ell^A_{\text{DV}}$. There is another arc from $x$ to $W^A$ with lower-bound and upper-bound capacities $b_\alpha(x, W^A) = b_\beta(x, W^A) = \ell^A_{\text{AV}}$.

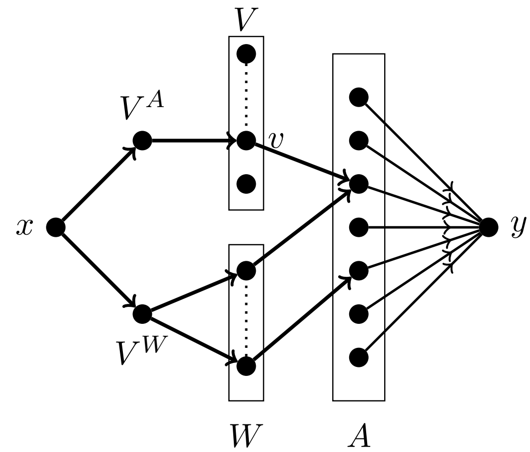


**Figure 1: An illustration of the construction of the IMCF instance in Theorem 6.1.**

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 1 if $v$ prefers $d$ to $c$. Analogously we define edges from $W^A$ to vertices $w$ corresponding to voters in $W$ approving some $a \in A$. There is an edge from some $v \in V \cup W$ to some $a \in A$ with capacity one if and only if $v$ prefers $a$ most. Each $a \in A$ yields an arc $(a, y)$ with capacity $s$.

One can check that there is a (maximum) flow with value

$$score_{(C,V)}(A) - \ell^A_{\text{DV}} + \ell^A_{\text{AV}}$$

and (minimum) cost of at most $\left\lfloor \frac{n'}{2} \right\rfloor - score(d)$ if and only if we can find exactly $score_{(C,V)}(A) - \ell^A_{\text{DV}}$ (remaining) voters in $V$ approving some $a \in A$ and exactly $\ell^A_{\text{AV}}$ voters added from $W$ approving some $a \in A$ such that each $a \in A$ has at most $s$ approvals, and a weak majority of voters prefers $c$ to $d$ in the final election. □

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

Theorem 6.2. *PRun-EDCAV+DV and VRun-EDCAV+DV are in P.*

Given the above results, we obtain the following corollary.

Corollary 6.3. *PRun-Y and VRun-Y are in P for all $Y \in \{CCAV, CCDV, CCRV, DCAV, DCDV, DCRV\}$.*

Concerning candidate control, we have the following result.

Theorem 6.4. *PRun-CCAC, VRun-CCAC, PRun-DCAC, and VRun-DCAC are NP-complete.*

Proof. We prove the theorem by reductions from the RX3C problem. Due to space constraints, we give proofs only for VRun-CCAC and PRun-CCAC. We first consider Veto with runoff.

**VRun-CCAC.** For a given RX3C instance $(U, \mathcal{S})$ where $|U| = |\mathcal{S}| = 3\kappa$, we create the following instance. For each $u \in U$, we create a registered candidate denoted still by $u$ for simplicity. In addition, we create two registered candidates $c$ and $q$ with $c$ being

| | CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Copeland$^\alpha$ | NPC | NPC | NPC | NPC | NPC | NPC | NPC | NPC | **NPC** | NPC | NPC | NPC |
| Maximin | NPC | NPC | **NPC** | NPC | P | **NPC** | NPC | NPC | **NPC** | P | P | **P** |
| Plurality | P | P | P | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| 2-Approval | P | P | ? | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| 3-Approval | P | NPC | NPC | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| ≥ 4-Approval | NPC | NPC | NPC | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| Veto | P | P | P | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| 2-Veto | P | P | **P** | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| ≥ 3-Veto | NPC | NPC | **NPC** | NPC | NPC | NPC | P | P | P | NPC | NPC | NPC |
| PRun | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |
| VRun | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |

**Table 1: Our results are in boldface. Other results are from [7, 14, 15, 23, 26]. The complexity of CCRV for 2-Approval is open.**

the distinguished candidate. Hence, the set of registered candidates is $C = U \cup \{c, q\}$. The unregistered candidates are created according to $\mathcal{S}$, one for each $S \in \mathcal{S}$ denoted by the same symbol for simplicity. We create a multiset $V$ of votes as follows.

- We create a vote of the form $\mathcal{S} \succ U \succ c \succ q$.
- For each $u \in U$, we crate $6\kappa - 3$ votes of the form $c \succ q \succ \mathcal{S} \succ U \setminus \{u\} \succ u$.
- For each $S \in \mathcal{S}$, we create $6\kappa + 5$ votes as follows (number of votes: preferences):

$$3\kappa+1: \quad q \succ U \succ c \succ \mathcal{S} \setminus \{S\} \succ S.$$
$$3\kappa+1: \quad c \succ U \succ q \succ \mathcal{S} \setminus \{S\} \succ S.$$
$$3: \quad q \succ U \succ \mathcal{S} \setminus \{S\} \succ c \succ S.$$

- For each $S = \{u_x, u_y, u_z\} \in \mathcal{S}$, we further create six votes as follows (number of votes: preferences).

$$2: \quad c \succ q \succ U \setminus \{u_x\} \succ \mathcal{S} \setminus \{S\} \succ u_x \succ S.$$
$$2: \quad c \succ q \succ U \setminus \{u_y\} \succ \mathcal{S} \setminus \{S\} \succ u_y \succ S.$$
$$2: \quad c \succ q \succ U \setminus \{u_z\} \succ \mathcal{S} \setminus \{S\} \succ u_z \succ S.$$

We are allowed to add at most $\kappa$ candidates, i.e., $\ell = \kappa$. Note that in the election restricted to the registered candidates, $c$ has $3\kappa \cdot (3\kappa+1) + 9\kappa$ vetoes, $q$ has $3\kappa \cdot (3\kappa+1) + 1$ vetoes, and every $u \in U$ has $6\kappa + 3$ vetoes. Hence, $c$ is not a Veto with Runoff winner of the election. It remains to prove the correctness of the reduction.

($\Leftarrow$) Assume that there is an exact 3-set cover $\mathcal{S}' \subseteq \mathcal{S}$ of $U$. After adding the candidates in $\mathcal{S}'$, candidate $q$ has 1 veto, every $S \in \mathcal{S}'$ has at least $6\kappa + 11$ vetoes, every $u \in U$ has $6\kappa + 3 - 2 = 6\kappa + 1$ vetoes, and $c$ has $6\kappa$ vetoes. Hence, $q$ and $c$ move into the runoff stage. As more voters prefer $c$ over $q$, $c$ becomes a final winner.

($\Rightarrow$) Suppose that we can add a subset $\mathcal{S}' \subseteq \mathcal{S}$ of at most $\kappa$ unregistered candidates to makes $c$ a winner under Veto with Runoff. Observe first that $\mathcal{S}'$ must contain exactly $\kappa$ 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 in $U$ and $q$ would be the two candidates going to the runoff stage. Then, from $|\mathcal{S}'| = \kappa$, it follows that $c$ has $6\kappa$ vetoes after adding candidates in $\mathcal{S}'$. If $\mathcal{S}'$ is not an exact 3-set cover, there must be a candidate $u \in U$ occurring in at least two subsets of $\mathcal{S}'$. Then, the candidate $u$ has at most $6\kappa + 3 - 4 = 6\kappa - 1$ vetoes, leading to $q$ and $u$ the two competing in the runoff stage. So, we can conclude that $\mathcal{S}'$ is an exact 3-set cover.

**PRun-CCAC.** For each $u \in U$, we create a registered candidate denoted by the same symbol. In addition, we create two registered candidates $q$ and $c$ with $c$ being the designated candidate. Moreover, for each $S \in \mathcal{S}$, we create an unregistered candidate denoted by the same symbol. Regarding the votes, we create $15 + 24\kappa$ votes in total. First, we create 8 votes with $q$ in the first position. Second, we create 7 votes with $c$ in the first position. Third, for each $u \in U$, we crate 2 votes with $u$ in the first position. The preferences over other candidates except the top-ranked one in the above $15 + 6\kappa$ votes can be set arbitrarily. Finally, for each $S \in \mathcal{S}$ and each $u \in S$, we create 2 votes of the form $S \succ u \succ U \setminus \{u\} \succ c \succ q \succ \mathcal{S} \setminus \{S\}$. We complete the construction by setting $\ell = \kappa$, i.e., we are allowed to add at most $\kappa$ candidates. This completes the construction. It remains to prove the correctness of the reduction.

If there is an exact 3-set cover $\mathcal{S}' \in \mathcal{S}$, we claim that $\mathcal{S}'$ is a solution of the CCAC instance constructed above. Clearly, after adding candidates in $\mathcal{S}'$, $q$ has 8 approvals, $c$ has 7 approvals, every $S \in \mathcal{S}'$ has 6 approval, and every $u \in U$ has $8 - 2 = 6$ approvals. Then, according to the definition of Plurality with Runoff, $q$ and $c$ go into the runoff stage. Clearly, a majority of voters prefer $c$ to $q$, and hence $c$ becomes the unique winner after adding all candidates in $\mathcal{S}'$. Consider the opposite direction. Observe that to ensure $c$ to survive the first stage, at least $\kappa$ candidates must be added, since otherwise there is at least one candidate $u \in U$ which receives at least 8 approvals, resulting in $q$ and this candidate to go into the runoff stage. Let $\mathcal{S}'$ be a solution. As discussed $|\mathcal{S}'| = \kappa$. If $\mathcal{S}'$ is not an exact 3-set cover, again there is a candidate $u \in U$ such that $u$ is not in any subset of $\mathcal{S}'$. According to the construction of the instance, the candidate $u$ receives at least 8 approvals after adding candidates in $\mathcal{S}'$, and hence survives the first stage with $q$. Therefore, $\mathcal{S}'$ must be an exact 3-set cover of $U$. □

Now we study the complexity of control by deleting candidates for Plurality with Runoff and Veto with runoff.

Theorem 6.5. *PRun-CCDC, VRun-CCDC, PRun-DCDC, and VRun-DCDC are NP-complete.*

Proof. Due to space constraints, we only prove PRun-CCDC. For a given RX3C instance $(U, \mathcal{S})$ such that $|U| = |\mathcal{S}| = 3\kappa$, we create the following instance. Without loss of generality, assume that $\kappa \geq 4$. Let $U = \{u_1, u_2, \ldots, u_{3\kappa}\}$. Let $C = \{c, q\} \cup U \cup \mathcal{S}$ be the set of candidates and $c$ the designated candidate. We create a multiset $V$ of $9\kappa^2 + 21\kappa + 1$ votes as follows.

- We crate $2\kappa$ votes of the form
$$q \succ u_1 \succ u_2 \succ \cdots \succ u_{3\kappa} \succ \mathcal{S} \succ c.$$
- We create $\kappa + 1$ votes of the form
$$q \succ u_{3\kappa} \succ u_{3\kappa-1} \succ \cdots \succ u_1 \succ \mathcal{S} \succ c.$$
- For each $u \in U$, we create $3\kappa - 3$ votes of the form
$$u \succ U \setminus \{u\} \succ \mathcal{S} \succ c \succ q.$$
- For each $S \in \mathcal{S}$, we create three votes of the form
$$S \succ c \succ C \setminus (S \cup \{c, q\}) \succ q.$$
- For each $S = \{u_x, u_y, u_z\} \in \mathcal{S}$, we further create six votes as follows (number of votes: preferences):
$$\begin{array}{ll} 2: & S \succ u_x \succ C \setminus \{c, q, u_x\} \succ c \succ q. \\ 2: & S \succ u_y \succ C \setminus \{c, q, u_y\} \succ c \succ q. \\ 2: & S \succ u_z \succ C \setminus \{c, q, u_z\} \succ c \succ q. \end{array}$$

Furthermore, let $\ell_{\text{DC}} = \kappa$. It remains to prove the correctness.

($\Leftarrow$) Assume there is an exact 3-set cover $\mathcal{S}' \subseteq \mathcal{S}$. After deleting the candidates in $\mathcal{S}'$, $q$ has $2\kappa + \kappa + 1 = 3\kappa + 1$ approvals, $c$ has $3\kappa$ approvals, every remaining $S \in \mathcal{S} \setminus \mathcal{S}'$ has 9 approvals, and every $u \in U$ has $3\kappa - 3 + 2 = 3\kappa - 1$ approvals. Hence, $q$ and $c$ go to the runoff stage, leading to $c$ to be the final winner.

($\Rightarrow$) Assume that it is possible to make $c$ a Plurality with Runoff winner of the election by deleting a set $C' \subseteq C \setminus \{c\}$ of at most $\kappa$ candidates. 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. Therefore, $q$ has at least $3\kappa + 1$ approvals in the final election. Furthermore, none of the candidates in $U$ can be deleted, i.e., $U \cap C' = \emptyset$. In fact, if we delete some candidate $u \in U$, then the candidate ranked immediately after $u$ in the $3\kappa - 3$ votes created for $u$ would receive at least $(3\kappa - 3) + (3\kappa - 3) = 6\kappa - 6$ approvals, preventing $c$ from winning. 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 $\kappa$ candidates. In summary, we have $C' \subseteq \mathcal{S}$. After deleting the candidates in $C'$, $c$ has $3|C'|$ approvals. Note that $|C'| = \kappa$ must hold, since 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. Therefore, we know that $c$ receives $3\kappa$ approvals after deleting all candidates in $C'$. If $C'$ is not an exact 3-set cover, then there must be a candidate $u \in U$ who occurs in at least two subsets of $C'$. Due to the construction, the candidate $u$ receives at least $3\kappa - 3 + 2 + 2 = 3\kappa + 1$ approvals, implying that $q$ and $u$ are the two candidates surviving the first 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 the hardness results in the above two theorems hold regardless of the tie-breaking rule used.

Finally, we study the complexity of control by replacing candidates for Plurality with Runoff and Veto with runoff. Observe that Plurality with runoff is unanimous. Then, the NP-hardness of PRun-CCAC studied in Theorem 6.4 and Lemma 4.1 directly yield the NP-hardness of PRun-CCRC. In addition, Plurality with runoff satisfies IBC when ties are broken deterministically. Hence, from the NP-hardness of PRun-DCDC studied in Theorem 6.5 and Lemma 3.1, it follows that PRun-DCRC is NP-hard when ties are broken deterministically. We can extend this NP-hardness result to all tie-breaking rules by deriving a reduction where no tie occurs in the constructed instance of PRun-DCRC. However, it is easy to check that Veto with runoff is not unanimous and does not satisfy IBC too. Hence, we can not obtain the NP-hardness for VRun-CCRC and VRun-DCRC using Lemmas 3.1 and 4.1. Nevertheless, we can show the NP-hardness of these problems by modifications of the proofs for VRun-CCAC and VRun-CCDC studied in Theorems 6.4 and 6.5. In summary, we have the following results.

THEOREM 6.6. *PRun-CCRC, VRun-CCRC, PRun-DCRC, and VRun-DCRC are NP-complete.*

Similar to Theorems 6.4 and 6.5, in our NP-hardness reductions for the problems in Theorem 6.6 no tie occurs in both stages.

## 7 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 refer to Table 1 for a summary of the complexity of different control problems for these voting correspondences and our concrete contributions. Our proofs are based on the non-unique winner model but can be modified for the unique-winner model of the control problems. Notice that the complexity of CCRV for 2-Approval remained as the only open problem in Table 1. The polynomial-time algorithm for 2-Veto-CCRV in Theorem 5.1 can not be trivially extended to 2-Approval. In 2-Veto, any optimal solution only replaces voters in $V$ that vetoing the distinguished candidate. However, this is not the case in 2-Approval. In a worst case, we need to replace votes in $V$ that do not approve $c$ with some votes in $W$ that also do not approve $c$. It is not clear how to reduce such a worst-case instance to a $b$-EC instance.

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 Plurality with Runoff and Veto with Runoff. In addition, it is also interesting to study the parameterized complexity of control problems for Plurality with Runoff and Veto with Runoff. Third, it is important to point out that our NP-completeness results are purely worst-case analysis and whether they are difficult to solve in practice needs to be further investigated. Finally, our polynomial-time algorithm in Theorem 6.1 relies on that ties are broken in favor of the chair. It is interesting to see if the result still holds for other tie-breaking rules. It has been observed that tie-breaking rules may affect the complexity of strategic voting problems [3, 30, 36].

## ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees of AAMAS 2019 for their valuable comments and helpful suggestions. The work was supported by the National Natural Science Foundation of China (Grant No. 61702557), the China Postdoctoral Science Foundation (Grant No. 2017M612584), and the DFG (Grant No. ER 738/2-1, ER 738/2-2). The paper was written in part while the first and second authors were affiliated at University of Siegen.

## REFERENCES


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

[2] S. Arora and B. Barak. 2009. *Computational Complexity: A Modern Approach.* Cambridge University Press.

[3] H. Aziz, S. Gaspers, N. Mattei, N. Narodytska, and T. Walsh. 2013. Ties Matter: Complexity of Manipulation when Tie-Breaking with a Random Vote. In *Proceedings of the 27th AAAI Conference on Artificial Intelligence.* 74–80.

[4] J. Bang-Jensen and G. Gutin. 2008. *Digraphs: Theory, Algorithms and Applications.* Springer-Verlag, London.

[5] J. J. Bartholdi III and J. B. Orlin. 1991. Single Transferable Vote Resists Strategic Voting. *Social Choice and Welfare* 8, 4 (1991), 341–354.

[6] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. 1989. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare* 6, 3 (1989), 227–241.

[7] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. 1992. How Hard Is it to Control an Election? *Mathematical and Computer Modelling* 16, 8-9 (1992), 27–40.

[8] D. Baumeister and J. Rothe. 2015. Preference Aggregation by Voting. In *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, J. Rothe (Ed.). Springer-Verlag, Chapter 4, 197–325.

[9] N. Betzler and J. Uhlmann. 2009. Parameterized Complexity of Candidate Control in Elections and Related Digraph Problems. *Theoretical Computer Science* 410, 52 (2009), 5425–5442.

[10] D. B. West. 2000. *Introduction to Graph Theory.* Prentice-Hall.

[11] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. 2017. Elections with Few Voters: Candidate Control Can Be Easy. *Journal of Artificial Intelligence Research* 60 (2017), 937–1002.

[12] G. Erdélyi, M. Fellows, J. Rothe, and L. Schend. 2015. Control Complexity in Bucklin and Fallback Voting: A Theoretical Analysis. *Journal of Computer and System Sciences* 81, 4 (2015), 632–660.

[13] G. Erdélyi, M. Nowak, and J. Rothe. 2009. Sincere-Strategy Preference-Based Approval Voting Fully Resists Constructive Control and Broadly Resists Destructive Control. *Mathematical Logic Quarterly* 55, 4 (2009), 425–443.

[14] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. 2009. Llull and Copeland Voting Computationally Resist Bribery and Constructive Control. *Journal of Artificial Intelligence Research* 35 (2009), 275–341.

[15] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. 2011. Multimode Control Attacks on Elections. *Journal of Artificial Intelligence Research* 40 (2011), 305–351.

[16] P. Faliszewski and J. Rothe. 2016. Control and Bribery in Voting. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia (Eds.). Cambridge University Press, Chapter 7, 146–168.

[17] H. N. Gabow. 1983. An Efficient Reduction Technique for Degree-constrained Subgraph and Bidirected Network Flow Problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing.* 448–456.

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

[19] T. F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science* 38 (1985), 293–306.

[20] M. Grötschel, L. Lovász, and A. Schrijver. 1988. *Geometric Algorithms and Combinatorial Optimization.* Springer, Heidelberg.

[21] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. 2007. Anyone But Him: The Complexity of Precluding an Alternative. *Artificial Intelligence* 171, (5-6) (2007), 255–285.

[22] E. Hemaspaandra, L. A. Hemaspaandra, and H. Schnoor. 2014. A Control Dichotomy for Pure Scoring Rules. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence.* 712–720.

[23] A. Lin. 2011. The Complexity of Manipulating $k$-Approval Elections. In *Proceedings of the 3th International Conference on Agents and Artificial Intelligence.* 212–218.

[24] A Loreggia. 2012. *Iterative Voting and Multi-Mode Control in Preference Aggregation.* Master's thesis. University of Padova.

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

[26] A. Loreggia, N. Narodytska, F. Rossi, K.B. Venable, and T. Walsh. 2015. Controlling Elections by Replacing Candidates or Votes. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems.* 1737–1738.

[27] K. Magiera and P. Faliszewski. 2017. How Hard Is Control in Single-Crossing Elections? *Autonomous Agents and Multi-Agent Systems* 31, 3 (2017), 606–627.

[28] C. Maushagen and J. Rothe. 2016. Complexity of Control by Partitioning Veto and Maximin Elections and of Control by Adding Candidates to Plurality Elections. In *Proceedings of the 22th European Conference on Artificial Intelligence.* 277–285.

[29] C. G. Menton and P. Singh. 2013. Control Complexity of Schulze Voting. In *Proceedings of the 23th International Joint Conference on Artificial Intelligence.* 286–292.

[30] S. Obraztsova, E. Elkind, and N. Hazon. 2011. Ties Matter: Complexity of Voting Manipulation Revisited. In *Proceedings of the 10th Internatinal Conference on Autonomous Agents and Multiagent Systems.* 71–78.

[31] J. Rothe and L. Schend. 2012. Control Complexity in Bucklin, Fallback, and Plurality Voting: An Experimental Approach. In *Proceedings of the 11th International Symposium on Experimental Algorithms.* 356–368.

[32] C. A. Tovey. 2002. Tutorial on Computational Complexity. *Interfaces* 32, 3 (2002), 30–61.

[33] Y. Yang. 2017. The Complexity of Control and Bribery in Majority Judgment. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems.* 1169–1177.

[34] Y. Yang and J. Guo. 2014. Controlling Elections with Bounded Single-Peaked Width. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems.* 629–636.

[35] Y. Yang and J. Guo. 2017. The Control Complexity of $r$-Approval: From the Single-Peaked Case to the General Case. *Journal of Computer and System Sciences* 89 (2017), 432–449.

[36] Y. Yang and J. Wang. 2017. Anyone But Them: The Complexity Challenge for a Resolute Election Controller. In *Proceedings of the 16th Internatinal Conference on Autonomous Agents and Multiagent Systems.* 1133–1141.