# Computational Analysis of Electoral Control in Single- and Multiwinner Elections

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf

vorgelegt von


**Roman Zorn**
aus Lauterbach


Düsseldorf, August 2025

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Prof. Dr. Jörg Rothe

2. Prof. Dr. Egon Wanke

Tag der mündlichen Prüfung: 15.12.2025

## Abstract

In this thesis we study the computational complexity of several decision problems in the area of electoral control. Electoral control is a part of computational voting theory which in turn belongs to the research area of computational social choice. In this model, a group of voters expresses their preferences over a set of candidates, and a specified voting rule is used to determine the winners of this election. In electoral control, an agent – called the chair – can influence the structure of a given election by, for example, removing or adding candidates. The chair's goal is either to help a particular candidate win or to prevent a particular candidate from winning. Since we want to prevent these kinds of malicious actions, we herein study the computational complexity of decision problems that model them. In simple terms, the higher the complexity of such a problem, the harder it is for the chair to compute which candidates need to be added (or removed) to achieve their objective. Therefore, high complexity can be viewed as a desirable feature when selecting a voting rule.

A special focus of this thesis is the control action of replacing candidates or voters. This type of control action has not been studied as much as others in the literature before. When replacing, the chair may simultaneously add and remove voters or candidates, but the number added must equal the number removed, so that the election's overall size remains unchanged. This might be used in practice to conceal the chair's tampering with the election. We therefore study the complexity of control by replacing for various voting rules in this thesis.

Another important focus of our work is multiwinner elections. In this model, the winner of the election is not a single candidate but a set of candidates, called a committee. In this thesis we show the complexity of control by replacing for two widely used multiwinner voting rules. We also investigate multiwinner voting rules in the context of the complexity of their control problems concerning the cloning of candidates.

When cloning candidates, the chair can create new candidates that are so similar to already existing ones that each voter ranks these candidates as a contiguous block in their preference order. We study this model in both an optimistic and a pessimistic setting and under three different cost models. In the most general model, the cost of cloning a candidate varies from clone to clone and from candidate to candidate; in the most specific model, cloning is entirely free.

# Contents

# Chapter 1

# Introduction

Elections are a critically important part of our modern world. They have widespread use in politics and in our everyday life – for example, when a group of friends decides on their next leisure activity – and they also play a role in our modern business world, for instance, when a board of directors makes decisions for their company. As a mechanism for collective decision making, elections are also studied in social sciences. In this work, we examine them from a complexity-theoretic standpoint, investigating how hard it is for a so-called election *chair* to change the outcome of an election in their favor. This line of research has garnered increasing interest at scientific conferences on artificial intelligence, especially in the field of *multiagent systems*, where multiple agents try to solve some kind of problem together that might be too hard for a single agent. In this context, elections are a possible way for those agents to interact with each other.

This line of research on elections belongs to the field of computational social choice (COMSOC). COMSOC is a relatively new field of research that connects theoretical computer science and artificial intelligence with social choice theory. COMSOC investigates social choice problems through the lens of computer science. This includes for example investigating the runtime of algorithmic implementations of voting rules, investigating their axiomatic properties or, as done in this work, investigating the computational complexity of manipulating elections. COMSOC as a whole contains multiple sub-fields such as voting, judgment aggregation, and participatory budgeting and closely intersects with game theory and fair division. This thesis focuses primarily on preference aggregation by voting.

In this field of research, there are three main ways to influence an election's outcome. The first is *manipulation*, where one or more voters cast dishonest ballots to try to shape the election to their liking. The second is *bribery*, in which an external agent has a limited budget which they can use to change the votes in the election. The third – and the one this thesis focuses on – is *control*. Here, the election chair can alter the structure of an election in some way, for example by adding or removing candidates or voters or by partitioning them into multiple districts. By investigating the computational complexity of such control actions, we identify which voting rules might be more susceptible to such kinds of malicious acts and thus learn which rules should be avoided or at least treated with caution.

This line of research began with the seminal work by Bartholdi, Tovey, and Trick [3] who first proposed control as a means of tampering with elections. They introduced the *constructive* variant of the problem, in which the goal of the chair is to make a specific candidate win. This work was later continued by Hemaspaandra, Hemaspaandra, and Rothe [17] who introduced the *destructive* variant of the problem, where the chair tries to prevent a given candidate from becoming a winner. Since then, many papers have studied the complexity of control problems for various voting rules. Comprehensive overviews of these results can be found in the book chapters by Faliszewski and Rothe [13] and by Baumeister and Rothe [5].

Importantly, most of these results focus on the three "main" methods of control: adding, deleting, and partitioning voters or candidates. This thesis shifts attention towards another type of control action: replacing voters and candidates. This control action was first proposed by Loreggia et. al [22, 23] and describes the chair both removing and adding voters or candidates simultaneously, with the restriction that they must add and remove them in equal amount, hence not changing the size of the election. To date, this type of control has only been studied for a few single-winner voting rules. In this thesis, we extend those results to more single-winner rules and introduce results on this type of control for multiwinner voting rules.

The most common voting rules used in our everyday lives are single-winner rules aiming to select a single candidate as the winner of an election. Multiwinner voting rules are a more generalized version of single-winner voting rules: instead of electing one winner, they select a fixed-size set of winners, called a *committees*.Such rules are useful in various scenarios, for example, when electing a board of representatives. Meir et al. [24] were the first to investigate the complexity of control problems for multiwinner elections. We will build upon this basis and slightly remodel their approach to align more closely with the literature on control of single-winner elections and expand their results to control by replacing for two common multiwinner rules.

Another less-studied control action is *cloning* candidates. Here, the chair adds new candidates which are so similar to already existing ones that each voter ranks them next to each other in their preference order. This model was first introduced by Tideman [31] along with the notion of *independence of clones*, which basically describes whether a given voting rule is immune to control by cloning candidates. The first to study the computational complexity of control by cloning candidates were Elkind et al. [9, 10]. Their work was focused on single-winner voting rules and in this thesis we will adapt their model to multiwinner voting, presenting complexity results for several multiwinner voting rules. This will be done for multiple different models of cloning. We distinguish between *possible* cloning, where we ask if a candidate can become a member of a winning committee for some ordering of clones, and *necessary* cloning, require the candidate to become a memeber of a winning committee for all possible orderings of clones. Additionally, we consider three cost models: *zero cost*, where cloning is free or the chair has an unlimited budget; *unit cost*, where the chairs has a limited budget and creating clones always costs the same; and *general cost,* where the cost of cloning a candidate can vary depending on the candidate cloned and the number of clones already created for that candidate.

Chapter 2 of this thesis will provide background information and explain the basic concepts needed to understand the results in this thesis. It includes details on complexity theory as well as voting theory and the multiple scenarios of electoral control featured in this thesis. Chapter 3 presents our results on control – espacially for control by replacing candidates or voters – for various single-winner voting rules. Chapter 4 contains our results on control by replacing voters or candidates for multiwinner voting rules. Chapter 5 covers our results on the complexity of control by cloning in multiwinner elections. Finally, Chapter 6 summarizes our findings and gives an outlook on possible directions for further research.

# Chapter 2

# Background

In this chapter, we introduce the two main areas of research this thesis is focused on. For each, we provide the foundational concepts necessary to understand the results presented later in this thesis. We also discuss important notions and the models used in our research. Section 2.1 covers computational complexity theory, while Section 2.2 addresses voting theory.

## 2.1 Complexity Theory

This section provides an overview of the field of computational complexity theory. For additional information on this topic, we refer the reader to the books by Arora and Barak [1], Papadimitriou [28], and Rothe [29].

The primary objective of complexity theory is to determine the computational complexity of a given *problem*. There are various types of problems – for example, *optimization problems* and *decision problems*. In this thesis, we focus on the latter.
A decision problem consists of three components:

1. Its name, often accompanied by an abbreviation.
2. An input, describing what an instance of the problem looks like.
3. A yes-or-no question which we want to answer for any given instance of the problem.

One of the most well known and most important decision problems is the boolean satisfiability problem (SAT)[15]:

| SATISFIABILITY (SAT) | |
|---|---|
| **Given:** | A boolean formula $\varphi$ in conjunctive normal form. |
| **Question:** | Is there a truth assignment for the variables in $\varphi$ that satisfies $\varphi$? |

Any input that satisfies the specified input requirements is considered an *instance* of the problem. If, for a given instance, the answer to the problem's question is "yes", we call it a *YES-instance*; otherwise, it is called a *NO-instance*.

Now that we have defined what constitutes a problem, we turn our attention to its complexity. Most problems can be solved by an *algorithm*. A *deterministic* algorithm is a finite sequence of unambiguous instructions that, when executed for a given input $I$, outputs the correct answer to a problem's question for $I$. By the Church-Turing thesis [14], every effectively calculable algorithm can be computed by a Turing machine. Therefore, the terms "algorithm" and "Turing machine" are henceforth used interchangeably. Turing machines were introduced by Turing [32, 33] and will not be defined in detail in this work. Further information on them can, for example, be found in the books cited at the beginning of this section.
Importantly, there are two types of Turing machines:

- A *deterministic Turing machine* (DTM) executes exactly one computation step at a time. After each step, it deterministically chooses the next step. Figuratively, a DTM's execution is a single, linear path; at the end of this path, it outputs either "yes" or "no" as an answer to our question for the given input.
- A *nondeterministic Turing machine* (NTM), by contrast, branches at each step and computes all possible next steps simultaneously. Its computation forms a tree Graph rather than a single path. An NTM accepts an input for a given question if at least one path through its computation tree outputs a "yes". It is important to note that in practice a computer is not able to follow multiple computation paths simultaneously. Therefore, when executing an NTM, we need to simulate it via a DTM by executing every possible path sequentially until we find one that outputs a "yes" or until each one returned a "no".

In this thesis we classify algorithms by comparing their *worst-case runtime*. It is specified in relation to the size of their input, since that might obviously influence how long the algorithm takes to compute the solution. This means that we are looking at the maximum amount of computational steps a deterministic algorithm might take to solve an instance of the given input size. Using this, we can group algorithms into different *complexity classes*. These are defined by an upper bound for the worst-case runtimes of the algorithms contained within them. The two most important complexity classes – and also the ones this thesis focuses on – are P and NP:

- P consists of all decision problems that can be solved in *deterministic polynomial time*. This means there exists a deterministic algorithm that can solve all instances of this problem in time polynomial in the size of its input. We also say that these problems can be solved *efficiently*.

- NP consists of all decision problems that can be solved in *nondeterministic polynomial time*. This means there exists a nondeterministic algorithm that can solve all instances of this problem in time polynomial in the size of its input. Equivalently, there exists a deterministic algorithm that accepts all YES-instances of a given problem in time polynomial in the size of its input.

Since every DTM is also an NTM (that just happens to never compute two paths at once), it follows that P $\subseteq$ NP. However, it is still an open question if NP $\subseteq$ P. This is one of the most important unsolved problems in computer science and is one of the seven Millennium Prize Problems [6].

We can provide an upper bound for the complexity class of a given problem by constructing an algorithm that solves all instances of the problem within the computational time limits of this complexity class. However, it might be possible to construct a different, more efficient algorithm that solves the problem within the limits of a class of lower complexity. *Reductions* are a way to show a lower bound for the complexity class of a given problem by showing that it is at least as complex (or *hard*) as another problem. A problem $A$ is *polynomial-time many-one reducible* to problem $B$ (written as $A \leq_m^p B$) if and only if there exists a polynomial time computable, total function $f$ that maps instances of $A$ to instances of $B$ such that for every instance $I$ of $A$, $I$ is a YES-instance of $A$ if and only if $f(I)$ is a YES-instance of $B$. Importantly, the relation $\leq_m^p$ is both reflexive and transitive [29].
A problem is $\mathscr{C}$*-hard* (for a complexity class $\mathscr{C}$) if every other problem in class $\mathscr{C}$ is polynomial-time many-one reducible to this problem. Showing that a problem is $\mathscr{C}$-hard for a complexity class $\mathscr{C}$ also establishes a lower bound because it shows that the problem is at least as hard as any other problem in $\mathscr{C}$. Furthermore, we call a problem $\mathscr{C}$*-complete* if it both belongs to a class $\mathscr{C}$ and is $\mathscr{C}$-hard. We typically show that a problem is $\mathscr{C}$-hard via a polynomial-time many-one reduction from another $\mathscr{C}$-hard problem, using its transitivity.

Since Cook and Levin have independently proven that SAT (defined earlier) is NP-complete [8, 21] the catalog of problems known to be NP-complete has grown substantially. Many of them are listed in the book by Garey and Johnson [15]. One of these problems, which will also be used in reductions later in this thesis, is HITTING-SET, proven to be NP-complete by Karp [20]:

| HITTING-SET (HS) | |
|---|---|
| **Given:** | A set $X = \{x_1, \ldots, x_n\}$, a set of sets $\mathscr{S} = \{S_1, \ldots, S_m\}$ with $S_i \subseteq X$ for each $1 \leq i \leq m$ and an integer $r$. |
| **Question:** | Is there a subset $X' \subseteq X$ with $\|X'\| \leq r$ such that $X' \cap S_i \neq \emptyset$ for each $1 \leq i \leq m$? |

We will now give an example for a HITTING-SET instance.

**Example 2.1.** Let $r = 4$, $X = \{1,3,5,7,12,23,56\}$, and $\mathscr{S} = \{S_1, \ldots, S_6\}$ with

$$S_1 = \{1,2,3,4\},\ S_2 = \{6,7,11\},\ S_3 = \{56,65\},$$

$$S_4 = \{1,7,22\},\ S_5 = \{23\},\ S_6 = \{9,10,11,12\}.$$

The question is whether we can find a $r = 4$ element subset $X' \subseteq X$ that has at least one element in common with every set in $\mathscr{S}$. This is not the case here since we need to include 56, 23 and 12 to hit the sets $S_3$, $S_5$ and $S_6$. Moreover, we need to include either 1 and 7 or 3 and 7 to hit the other three sets. This means our set $X'$ would need to include 5 elements, exceeding our limit of 4. Hence, this is a NO-instance.
If we instead set $r = 5$, this becomes a YES-instance, with the solutions

$$X' = \{1,7,12,23,56\} \text{ and } X' = \{3,7,12,23,56\}.$$

## 2.2 Voting Theory

In this section we introduce important basics of computational voting theory used in this thesis. In this first part, we focus on *single-winner* elections. For a broader scope of this topic, see the book chapter by Baumeister and Rothe [5].
An *election* $E = (C,V)$ consists of a set of *candidates* $C$ and a list of *voters* (or *votes*) $V$. $V$ is a list because different voters might have identical preference orders. A vote consists of a strict linear order over the candidates in $C$ and expresses the voters preferences over the set of candidates. We write $a \succ b$ when a voter prefers candidate $a \in C$ over candidate $b \in C$. For simplicity, we sometimes omit "$\succ$" and write $b\ a\ c$ instead of $b \succ a \succ c$. Throughout this thesis, we mostly focus on *total* votes – that is, all candidates from $C$ are included in each vote. The relation "$\succ$" is

- *connected*: for each two distinct candidates $a,\ b \in C$, either $a \succ b$ or $b \succ a$,
- *transitive*: for each three candidates $a,\ b,\ c \in C$, $a \succ b$ and $b \succ c$ imply $a \succ c$, and

- *asymmetric*: for each two candidates $a,\ b \in C$, $a \succ b$ implies that $b \succ a$ does not hold.

Given an election $(C,V)$, a *voting rule* is used to determine a subset $W \subseteq C$ called the *winners* of the election. If $W = \{c\}$, we call $c \in C$ the *unique winner* of the election. If $|W| \geq 2$, we call each candidate $d \in W$ a *nonunique winner* of the election. We now introduce several voting rules used in this thesis.

Firstly, there are multiple voting rules belonging to a class of rules called *scoring protocols* (or *scoring rules*). Each scoring rule is defined by a *scoring vector* $\alpha = (\alpha_1, \dots, \alpha_m)$ where $m = |C|$ and the $\alpha_i$ are nonnegative integers satisfying $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$. A candidate gets $\alpha_i$ points from a vote if he is ranked $i$-th in that vote. A candidates score is the sum of points they get from each vote and the candidate (or candidates) with the highest scores are the winners of the election. Following are specific scoring rules mentioned in this thesis:

- *Plurality*: $\alpha = (1, 0, \dots, 0)$. Each voters gives one point to only their most preferred candidate.
- *k-Approval*: $\alpha = (\underbrace{1, \dots, 1}_{k}, 0, \dots, 0)$. Each voters gives one point to each of their $k$ most preferred candidates. Notably, Plurality is 1-Approval.
- *k-Veto*: $\alpha = (1, \dots, 1, \underbrace{0, \dots, 0}_{k})$. Each voters gives one point to each candidate but their $k$ least preferred candidates. 1-Veto is also called *Veto*.

Besides scoring rules, there are several other voting rules used in this thesis. Many of them are based on *pairwise comparisons*. For an election $E = (C,V)$, let $N_E(a,b)$ be the number of voters preferring candidate $a \in C$ over candidate $b \in C$. We now present the remaining single-winner voting rules from this thesis:

- *Condorcet*: A candidate $a \in C$ wins if and only if $N_E(a,b) > N_E(b,a)$ for every other candidate $b \in C \setminus \{a\}$. This means, that the Condorcet rules does not always produce a winner, and if it does, it is a unique winner.
- *Copeland*$^\alpha$: This rule is defined for every rational number $\alpha \in [0,1]$. For this rule we run a pairwise comparison between each pair of distinct candidates $a,b \in C$. If $N_E(a,b) > N_E(b,a)$, $a$ receives one point and $b$ none. If it is a tie, both candidates receive $\alpha$ points. The candidates with the highest scores summed over all pairwise comparisons win the election.

- *Maximin*: The maximin score of a candidate $a \in C$ is defined as $\min_{b \in C \setminus \{a\}} N_E(a,b)$. This means, that the maximin score of a candidate is the value of their worst pairwise comparison. The candidates with the highest maximin scores win.

- *Plurality with Runoff*: If a candidate is ranked first by every voter, they are the unique winner of this election. Otherwise, this voting rule takes two stages. In the first stage, all candidates but the two with the highest scores under the Plurality rule are eliminated. Ties are broken via some tie-breaking method. In the second stage, two candidates remain, call them $a, b \in C$. If $N_E(a,b) > N_E(b,a)$, $a$ wins the election. If $N_E(b,a) > N_E(a,b)$, $b$ wins the election. Otherwise, a tie-breaking method is applied to determine the winner between $a$ and $b$.

- *Veto with Runoff*: Veto with Runoff works analogously to Plurality with Runoff except that in the first round every candidate but the two with the highest score under the Veto rule are eliminated.

- *Fallback*: Voters may submit *partial* votes, meaning not every candidate must be included in their vote. In the first step, only look at the first position of each vote, giving one point to a candidate for each appearance. In the second step, repeat this process for the second position of the votes, adding the points to the candidates total and so on. If, after any step, one or more candidates reach more than $|V|/2$ points, the candidates with the most points win the election. Otherwise, the candidates with the most points after the last step win the election.

- *Range Voting*: Instead of giving their preferences as defined above, for this rule, voters independently assign each candidate points in form of an integer between 0 and $k$. The candidates with the most points summed over all voters win the election.

- *Normalized Range Voting*: Voters assign points the same way they do for Range Voting. For each voter $v \in V$, let $v_{min}$ and $v_{max}$ be the minimum and maximum amount of points this voter is assigning to any candidate and let $v_c$ be the amount of points they are assigning to any candidate $c \in C$. Then, the amount of points $c$ actually gets from $v$ is $\frac{k(v_c - v_{min})}{v_{max} - v_{min}}$. The candidates getting the most points summed over all voters win the election.

We will now use an example to illustrate how some of these rules work in practice.

**Example 2.2.** Let $E = (C,V)$ be an election with candidates $C = \{a,b,c,d\}$ and the following seven votes:

$$a \succ d \succ c \succ b$$
$$a \succ d \succ c \succ b$$
$$c \succ b \succ a \succ d$$
$$d \succ c \succ b \succ a$$
$$b \succ a \succ c \succ d$$
$$b \succ a \succ d \succ c$$
$$b \succ a \succ d \succ c$$

We can easily see that $b$ is the unique winner under the Plurality rule given they are ranked first by 3 voters while the other candidates are ranked first less often. On the other hand, $a$ is the unique winner under the Veto rule, being the only candidate ranked last by only a single voter.

Table 2.1 shows the values for all pairwise comparisons in this election. From these, we can see that $\{a,b\}$ are the Copeland$^\alpha$ winners of this election for all possible values of $\alpha$ (since there are no ties). Additionally, $b$ is the unique Maximin winner of this election with a maximin score of 3. Furthermore, there is no Condorcet winner in this election since every candidate looses at least one pairwise comparison.

In a Plurality with Runoff election, candidates $c$ and $d$ would be eliminated in the first round, as they have the lowest Plurality scores. In the second round, $b$ would win the election because it is winning its pairwise comparison with $a$.

Table 2.1: Pairwise comparisons $N_E(i,j)$ for Example 2.2

| $i$ \ $j$ | $a$ | $b$ | $c$ | $d$ |
|---|---|---|---|---|
| $a$ | - | 2 | 5 | 6 |
| $b$ | 5 | - | 3 | 4 |
| $c$ | 2 | 4 | - | 2 |
| $d$ | 1 | 3 | 5 | - |

Next, we turn our attention to *multiwinner* voting. More information on this topic can be found in the book chapter by Baumeister, Faliszewski, Rothe, and Skowron [4].

A *(multiwinner) election* is a triple $E = (C,V,k)$ where, similarly to single-winner voting, $C$ is the set of candidates and $V$ is a list of voters. Unlike in single-winner voting, we now seek a set of winning candidates of a fixed size. We call sets of candidates *committees*. A *multiwinner voting rule* is a function that takes a mulitwinner election and outputs a set $W = \{W_1,\dots,W_l\}$ of committees of size exactly $k$, the winning committees of that election.

We call a candidate $c \in C$ a *certain winner* of an election if they are included in every winning committee. We call them an *uncertain winner* if they are included in at least one but not all winning committees. Lastly, we call them a *certain nonwinner* if they are included in none of the winning committees.

In this thesis, we deal with the following multiwinner voting rules:

- *Single nontransferable vote* (SNTV): The winning committees consist of the $k$ candidates with the highest Plurality scores.
- *Bloc voting*: The winning committees consist of the $k$ candidates with the highest $k$-Approval scores.
- *k-Borda*: The winning committees consist of the $k$ candidates with the highest Borda scores. Borda is a single-winner scoring rule with the scoring vector $\alpha = (m-1, m-2, \dots, 1, 0)$, where $m = |C|$.
- *Single transferable vote* (STV): Let $q = \lfloor n/k+1 \rfloor + 1$ be the quota, where $n = |V|$. Iteratively, if a candidate is ranked first in at least $q$ votes add them to the winning committee and remove both them and $q$ votes that ranked them first from the election. If no candidate meets the quota, remove a candidate with the lowest Plurality score from the election instead. Repeat until the winning committee consists of exactly $k$ candidates.
- *t-Approval-CC*: Here, CC is stands for Chamberlin-Courant [7]. A voter gives a point to each committee that includes at least one candidate from the first $t$ positions of their vote. The committees with the most points over all voters win the election.
- *Borda-CC*: A voter gives each committee as many points as they would give to their most preferred candidate in that committee under the Borda rule. The committees with the most points over all voters win the election.

We now again illustrate these rules with an example.

**Example 2.3.** Let $E = (C, V, k)$ be an election with candidates $C = \{a, b, c, d, e\}$, $k = 3$, and the following seven votes:

$$a \succ e \succ c \succ b \succ d$$
$$e \succ c \succ d \succ b \succ a$$
$$c \succ e \succ d \succ a \succ b$$
$$d \succ b \succ c \succ a \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$a \succ d \succ e \succ c \succ b$$
$$e \succ c \succ b \succ a \succ d$$

In Table 2.2 we see the scores of the candidates for the relevant single-winner voting rules. Accordingly, the set of winning committees under SNTV is $\{\{a,b,e\},\{a,c,e\},\{a,d,e\}\}$. For Bloc voting, the only winning committee is $\{c,d,e\}$. And the only winning committee under $k$-Borda is $\{a,c,e\}$.

Table 2.2: Scores under single-winner scoring rules for the candidates from Example 2.3

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| Plurality | 2 | 1 | 1 | 1 | 2 |
| 3-Approval | 3 | 3 | 5 | 4 | 6 |
| Borda | 14 | 11 | 15 | 12 | 18 |

Next, we turn towards STV. Our quota $q = \lfloor n/k+1 \rfloor + 1 = \lfloor 7/3+1 \rfloor + 1 = 2$. There are two candidates, $a$ and $e$, that are ranked first twice in the initial election. We will choose to add $a$ to the winning committee via lexicographic tie-breaking and therefore remove $a$ and the first and sixth voters. The remaining votes are:

$$e \succ c \succ d \succ b$$
$$c \succ e \succ d \succ b$$
$$d \succ b \succ c \succ e$$
$$b \succ e \succ d \succ c$$
$$e \succ c \succ b \succ d$$

Now $e$ is the only candidate reaching the quota, so we add it to the winning committee and then remove it and the voters ranking it first from the election. The remaining votes are:

$$c \succ d \succ b$$
$$d \succ b \succ c$$
$$b \succ d \succ c$$

Since no candidate reaches the quota and all voters are tied in Plurality scores, we choose to remove $b$ via lexicographic tie-breaking. The remaining votes are:

$$c \succ d$$

$$d \succ c$$

$$d \succ c$$

Finally, $d$ is ranked first twice, so we add it to our winning committee. Therefore, $\{a,d,e\}$ wins this election.
In Table 2.3, we see scores for all committees under our two Chamberlin-Courant rules. As can be seen, $\{\{a,b,e\},\{a,d,e\}\}$ is our set of winning committees for Borda-CC. For 2-Approval-CC, all five committees scoring 7 points win this election.

Table 2.3: Points of the committees from Example 2.3

| | $\{a,b,c\}$ | $\{a,b,d\}$ | $\{a,b,e\}$ | $\{a,c,d\}$ | $\{a,c,e\}$ | $\{a,d,e\}$ | $\{b,c,d\}$ | $\{b,c,e\}$ | $\{b,d,e\}$ | $\{c,d,e\}$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 2-Approval-CC | 7 | 4 | 7 | 7 | 6 | 7 | 6 | 6 | 7 | 6 |
| Borda-CC | 25 | 22 | 26 | 25 | 25 | 26 | 22 | 24 | 25 | 24 |

We now turn to the computational complexity aspects of voting. The famous *Gibbard-Satterthwaite theorem* [16, 30] states, informally, that every natural, preference-based voting rule with more than two candidates can be manipulated by strategic voters. To combat this, Bartholdi, Tovey, and Trick suggested high computational complexity as a shield against manipulation [2] and later expanded this idea to control [3]. Therefore, we analyze the computational complexity of several control problems in this thesis. The most basic example of a control problem, given a voting rule $\mathscr{R}$, is as follows:

$\mathscr{R}$-CONSTRUCTIVE-CONTROL-BY-ADDING-CANDIDATES ($\mathscr{R}$-CCAC)

**Given:** Two sets of candidates, $C$ and $D$ with $C \cap D = \emptyset$, a list $V$ of votes over $C \cup D$, a distinguished candidate $c \in C$, and a positive integer $r \leq |D|$.

**Question:** Is it possible to add at most $r$ candidates from $D$ to $C$ such that $c$ is a certain $\mathscr{R}$ winner of the resulting election? That is, is there a subset $D' \subseteq D$ with $|D'| \leq r$ such that $c$ is an $\mathscr{R}$ winner of the election $(C \cup D', V)$?

For this problem, we are given an election and an additional set of candidates from which we can add up to $k$ candidates to the original election, with the goal of making some distinguished candidate win. For all these types of control problems, there are several variations. We will illustrate these with the following problem definition.

| $\mathscr{R}$-DESTRUCTIVE-CONTROL-BY-REPLACING-VOTERS ($\mathscr{R}$-DCRV) | |
|---|---|
| **Given:** | A set of candidates $C$, two lists $V$ and $U$ of votes over $C$, a distinguished candidate $c \in C$, and a positive integer $r \leq \|V\|$. |
| **Question:** | Are there sublists $V' \subseteq V$ and $U' \subseteq U$ such that $\|V'\| = \|U'\| \leq r$ and $c$ is not an $\mathscr{R}$ winner of the election $(C, (V \setminus V') \cup U')$? |

As we can see, each problem has a constructive and a destructive variant. For single-winner, voting this thesis focuses on the *nonunique-winner model*. In the constructive case, we ask whether $c$ can be made a winner, while in the destructive case we ask whether $c$ can be prevented from being a winner. There is also the *unique-winner model*, in which we ask whether $c$ can be made a unique winner of the election for the constructive case, while in the destructive case we try to prevent $c$ from becoming a unique winner (it may still be a winner).
To translate these problems to multiwinner voting we apply two simple steps. First, we add the target committee size $k$ to the instance and to the constructed election in the question. Second, we change the question: in the constructive case, we ask whether $c$ can be made a certain winner of the election; in the destructive case, we ask whether $c$ can be made a certain nonwinner of the election.
We say a voting rule is *immune* to a control problem if there is no instance in which the chair can successfully control the election (i. e., there is no YES-instance). Otherwise, the voting rule is *susceptible* to this kind of control. If a voting rule is susceptible to some kind of control, we call it *vulnerable* to this kind of control if the corresponding control problem belongs to P. If the control problem is NP-hard, we call the voting rule *resistant* to this kind of control.

Finally, we present our model for cloning candidates in multiwinner elections and define the corresponding control problems. Let $E = (C, V, k)$ be a multiwinner election with $m$ candidates and $n$ voters. Let $K = (K_1, \ldots, K_m)$ be a vector, called *cloning vector*, where, informally, each $K_i \geq 0$ indicates how many clones of candidate $c_i$ will replace the original, where $K_i = 0$ indicates that only the original candidate with no clones will remain in the election. A multiwinner election $E_K = (C', V', k)$ is *created by cloning $E$ via $K$* if

$$C' = (C \setminus \{c_i \in C | K_i \geq 1\}) \cup \{c_i^{(j)} | 1 \leq j \leq K_i\}$$

and $V'$ with each $v_i' \in V'$ being a total order over $C'$ that results from $v_i \in V$ by replacing the cloned candidates in the vote $v_i$ with their clones. All clones of a candidate appear as a block in each vote with no other candidates between them.
Notably, there are multiple possibilities to clone a multiwinner election depending of the order

of the clones of each candidates in each vote. The goal is to make the distinguished candidate $c \in C$ an uncertain winner of the cloned election. We consider two settings:

1. In the *optimistic setting* we require $c$ (or one of their clones) to become an uncertain winner of the cloned election for at least one possible ordering of the clones.
2. In the *pessimistic setting* we require $c$ (or one of their clones) to become an uncertain winner of the cloned election for every possible ordering of the clones.

We also consider three cost models for cloning. In the *general-cost model* (GC), for every candidate $c_i \in C$ there is a cost function $\rho_i : \mathbb{N} \to \mathbb{N}$ with $\rho_i(0) = \rho_i(1) = 0$ and for each $j, j' \in \mathbb{N}$ with $j < j'$ it holds that $\rho_i(j) \leq \rho_i(j')$. Here, $\rho_i(j)$ is the cost of cloning the $i$-th candidate $j$ times and replacing this candidate in all votes with these clones. There is also an integer $B$, called the *budget*.
The other two models are special cases of the general-cost model. In the *unit-cost model* (UC) in which $\rho_i(j) = j - 1$ for all $i$ and $j \geq 1$ every additional clone has an equal cost of one and the budget represents the maximum number of additional clones.
In the *zero-cost model* (ZC), a special case of the unit-cost model, either the budget is set to infinity, or $\rho_i(j) = 0$ for all $i$ and $j \geq 1$. Therefore, it is possible to create as many clones as wanted without being restrained by the budget.

To conclude this chapter, we define the decision problems for cloning.

$\mathscr{R}$-POSSIBLE-CLONING-GC

| | |
|---|---|
| **Given:** | A multiwinner election $E = (C, V, k)$, a cost function $\rho_i : \mathbb{N} \to \mathbb{N}$ for every $c_i \in C$, a distinguished candidate $c \in C$, and a budget $B$. |
| **Question:** | Is there a cloning vector $K = (K_1, \dots, K_m)$, with $\sum_{c_i \in C} \rho_i(K_i) \leq B$, such that $c$ (or one of its clones) is an uncertain winner under $\mathscr{R}$ in at least one cloned multwinner election $E_K$, resulting from cloning $E$ via $K$? |

The problem $\mathscr{R}$-NECESSARY-CLONING-GC is defined analogously, except that we ask whether $c$ is an uncertain winner under $\mathscr{R}$ for *all* multiwinner elections $E_K$ obtained from $E$ by cloning via $K$. If we use the unit-cost or the zero-cost model in this definition, we replace “GC” in the problem name by “UC” or “ZC” and omit the cost function from the problem instance; in the zero-cost model, we also omit the budget.

# Chapter 3

# Towards completing the puzzle: complexity of control by replacing, adding, and deleting candidates or voters

## 3.1 Summary

In this work, we studied the computational complexity of several control problems for various single-winner voting rules. Specifically, we focused on adding, deleting and replacing candidates or voters. While similar research has been conducted for a number of voting rules, this work seeks to fill in the remaining gaps. To that end, we placed special emphasis on control by replacing voters or candidates – an action that has received less attention than adding or deleting. In doing so, we obtained new complexity results for Copeland$^{\alpha}$, Maximin, $k$-Veto, Condorcet, Fallback, Range Voting, Normalized Range Voting, Plurality with Runoff, and Veto with Runoff. These results were achieved by means of reductions from known NP-hard problems and by providing polynomial-time algorithms for problems in P.

## 3.2 Personal Contribution

This work merges and extends two earlier papers.
I had no involvement in the first paper by G. Erdélyi, C. Reger, and Y. Yang, nor any of its results.
All initial technical results from the second paper by M. Neveling, J. Rothe, and myself were initially developed by me, with assistance from Marc Neveling. The bulk of the writing was

carried out by Marc Neveling, with finalization and polishing by Jörg Rothe and me.
I did not contribute to the new results that appear only in the journal version of this work. My role in merging the papers and producing the extended version was limited to providing feedback and minor corrections; the main effort in combining and writing the extended manuscript was undertaken by my co-authors.

## 3.3 Publication

G. Erdélyi, M. Neveling, C. Reger, J. Rothe, Y. Yang, and R. Zorn. "Towards completing the puzzle: complexity of control by replacing, adding, and deleting candidates or voters". In: *Journal of Autonomous Agents and Multi-Agent Systems*

The two preliminary versions of this paper, which were merged into the journal version, have been submitted and accepted at *18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019)* and in the *proceedings of the 15th International Computer Science Symposium in Russia (CSR 2020)* respectively; the latter paper was also presented at the *16th International Symposium on Artificial Intelligence and Mathematics (ISAIM 2020)* with non-archival website proceedings:

G. Erdélyi, C. Reger, and Y. Yang. "Towards completing the puzzle: Solving open problems for control in elections". In: *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*

M. Neveling, J. Rothe, and R. Zorn. "The complexity of controlling Condorcet, fallback, and $k$-veto elections by replacing candidates or voters". In: *Proceedings of the 15th International Computer Science Symposium in Russia*

Autonomous Agents and Multi-Agent Systems (2021) 35:41
https://doi.org/10.1007/s10458-021-09523-9

# Towards completing the puzzle: complexity of control by replacing, adding, and deleting candidates or voters

**Gábor Erdélyi[1] · Marc Neveling[2] · Christian Reger[3] · Jörg Rothe[2] · Yongjie Yang[4] · Roman Zorn[2]**

Accepted: 2 July 2021 / Published online: 29 July 2021


**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 rules. We complement those results by solving several open cases for Copeland$^{\alpha}$, maximin, $k$-veto, plurality with runoff, veto with runoff, Condorcet, fallback, range voting, and normalized range voting.

**Keywords** Computational complexity · Electoral control · Copeland · Maximin · Veto · Plurality with runoff · Veto with runoff · Condorcet · Fallback · Range voting · Normalized range voting

## 1 Introduction

Computational social choice has established itself as a central part in the research and development of multiagent systems and artificial intelligence. Without going into the details here, it is important to note that preference aggregation and voting—and the related scenarios of strategic behavior so as to change the outcome of elections—have many applications in artificial intelligence and, especially, in multiagent systems (e.g., in information extraction [57], planning [15], recommender systems [28], ranking algorithms [14], computational linguistics [53], automated scheduling [32], collaborative filtering [55], etc.). Interestingly, as noted by Hemaspaandra [36, p. 7971], *"At the 2017*

The authors are ordered alphabetically.
This paper merges and extends two preliminary versions that appeared in the proceedings of the *18th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS 2019) [21] and in the proceedings of the *15th International Computer Science Symposium in Russia* (CSR 2020) [50]; the latter paper was also presented at the *16th International Symposium on Artificial Intelligence and Mathematics* (ISAIM 2020) with nonarchival website proceedings.

✉ Yongjie Yang
yyongjiecs@gmail.com

Extended author information available on the last page of the article

*AAMAS conference, for example, there were four sessions devoted to Computational Social Choice; no other topic had that many sessions."*

Since the seminal work of Bartholdi, Orlin, Tovey, and Trick [5–7], the founders of computational social choice, many strategic voting problems have been proposed and studied from a complexity-theoretic point of view. These strategic voting problems include

- *manipulation* where voters cast their votes strategically;
- *bribery* where an external agent bribes some voters—without exceeding a given budget—so as to change their votes; and
- *electoral control* where an external agent (usually called the chair) tries to alter the outcome of an election by structural changes such as adding, deleting, partitioning, or replacing either candidates or voters.

For a broad overview of these strategic actions and their applications in artificial intelligence and multiagent systems and for a comprehensive survey of related results, we refer to the book chapters by Conitzer and Walsh [12], Faliszewski and Rothe [25], and Baumeister and Rothe [8] and to the comprehensive list of references cited therein.

We will focus on *electoral control*, first and foremost on control by replacing but also on control by adding and by deleting either candidates or voters. There is a long line of research centered on the complexity of control. So, before providing the specific motivation for our results, let us briefly outline the history of research on electoral control, focusing on the particular scenarios we will be concerned with.

Bartholdi, Tovey, and Trick [7] were the first to propose control of elections as a malicious way of tampering with their outcome via changing their structure, e.g., by adding or deleting voters or candidates. They introduced the constructive variant where the goal of an election chair is to make a favorite candidate win. Focusing on plurality and Condorcet elections, they determined which control scenarios these rules are *immune* to (i.e., impossible for the chair to successfully exert control), and in cases where these rules are not immune, they studied the complexity of the associated control problems, showing either *resistance* (NP-hardness) or *vulnerability* (membership in P). Complementing their work, Hemaspaandra, Hemaspaandra, and Rothe [33] introduced the destructive variant of control where the chair's goal is to prevent a despised candidate's victory. Pinpointing the complexity of destructive control in plurality and Condorcet elections, they also studied the constructive and destructive control complexity of approval voting.

As surveyed by Faliszewski and Rothe [25] and Baumeister and Rothe [8], plenty of voting rules have been analyzed in terms of their control complexity since then. In addition to the just mentioned results on plurality, Condorcet, and approval voting (and its variants) [7, 9, 16, 19, 33]; the complexity of control in various scenarios has been thoroughly analyzed for Copeland [9, 24]; maximin [23, 45, 47, 61]; $k$-veto and $k$-approval [39, 43, 46, 62]; Bucklin and fallback voting [16, 17, 20, 22], range voting and normalized range voting [48], and Schulze voting [49, 54]. Among these voting rules, *fallback voting* (a hybrid system due to Brams and Sanver [10] that combines Bucklin with approval voting) and *normalized range voting* (both will be defined in Sect. 3) are special in that they are the only two natural voting rules with a polynomial-time winner problem that are currently known to have the most resistances to standard control attacks. "Standard control" here refers to control by adding, deleting, or partitioning either candidates or voters because these are the control types originally introduced by Bartholdi, Tovey, and Trick [7].[1]

On the other hand, the computational complexity of *replacing* either candidates or voters—the control action we mostly focus on—was first studied by Loreggia et al. [40–43]. Replacement control models voting situations in which the number of candidates or voters are predefined and cannot be changed by the chair. For instance, a parliament often consists of a fixed number of seats whose occupants must be replaced if they are removed from their seats. From another viewpoint, the chair might try to veil his or her election tampering via replacement control actions by making sure that the number of participating candidates and voters is the same as before, hoping that the election might appear to be unchanged at first glance. There are also other types of electoral control, such as more natural models of control by partition introduced by Erdélyi, Hemaspaandra, and Hemaspaandra [18], but we will not consider those in this paper.

Compared with the standard control types (adding/deleting/partitioning voters or candidates), much less is known for the control action of replacing voters or candidates. It can be seen as a combination of adding and deleting them, with the additional constraint that the same number of voters/candidates must be added as have been deleted. Other types of combining standard control attacks, namely *multimode control*, have been investigated by Faliszewski, Hemaspaandra, and Hemaspaandra [23]. In their model, an external agent is allowed to perform different types of control actions at once such as deleting and/or adding voters and/or candidates. Although some types of multimode control seem to be similar to replacement control, the key difference lies in the tightly coupled control types of replacement control, whereas in multimode control the combined types of standard electoral control can often be handled separately. This leads to the interesting and subtle situation that resistances of voting rules to certain types of standard control do not transfer trivially to related types of replacement control, whereas this indeed can happen for multimode control.

The reader may ask, why do we need yet another paper on the complexity of control? That is, what is the main motivation for the research presented here? Well, the answer is twofold.

First, from a theoretical perspective, it is unsatisfactory that our knowledge about the complexity of control is still incomplete; there are several important voting rules for which we still have some unsolved open cases regarding certain control actions, especially for replacement control. *In this paper, we are filling many of these gaps (see Sect. 2 and, in particular, Table 1 for the details).*

Second, from a practical perspective, a designer of a multiagent system will have to have a careful look at which specific application of voting is planned in his or her system and which strategic scenarios the system will most likely be attacked with. Then, to make

[1] As defined by Bartholdi, Tovey, and Trick [7], for control by partition of either candidates or voters, there is a first round in which the candidates or voters are partitioned into two subgroups which separately elect winners who then may proceed to the final-round election. Hemaspaandra, Hemaspaandra, and Rothe [33] introduced two tie-handling rules, *ties eliminate* and *ties promote*, that determine which of the first-round winners proceed to the final runoff in case of a tie among two or more candidates in any of the two first-round subelections. Further, there are two variants of control by partition of candidates, one *with runoff* (where both subgroups send their winners to the final round) and one *without* (where the winners of one subgroup face *all* candidates of the other in the final round). Hemaspaandra, Hemaspaandra, and Menton [35] showed that certain destructive variants of these problems in fact are the same. In this paper, we will not consider any cases of control by partition, though.

**Table 1** Overview of results on the complexity of control by adding, deleting, and replacing either candidates or voters in various voting rules. Our results are in boldface. Previous results [7, 23, 24, 33, 39, 43, 48] are in gray. Entries "NPC" are a shorthand for "NP-completeness" and indicate resistance, "P" vulnerability, and "I" immunity results. The complexity of CCRV for 2-approval —marked by "?"—is still open

(a) Constructive control

| | CCAV | CCDV | CCRV | CCAC | CCDC | CCRC |
|---|---|---|---|---|---|---|
| Copeland$^{\alpha}$ | *NPC* | *NPC* | *NPC* | *NPC* | *NPC* | **NPC** |
| Maximin | *NPC* | *NPC* | **NPC** | *NPC* | *P* | **NPC** |
| Plurality | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| 2-Approval | *P* | *P* | ? | *NPC* | *NPC* | *NPC* |
| 3-Approval | *P* | *NPC* | *NPC* | *NPC* | *NPC* | *NPC* |
| $k$-Approval, $k \geq 4$ | *NPC* | *NPC* | *NPC* | *NPC* | *NPC* | *NPC* |
| Veto | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| 2-Veto | *P* | *P* | **P** | *NPC* | *NPC* | **NPC** |
| $k$-Veto, $k \geq 3$ | *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** |
| Plurality with runoff | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |
| Veto with runoff | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |
| Condorcet voting | *NPC* | *NPC* | **NPC** | *I* | *P* | **P** |
| Fallback voting | *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** |
| Range voting | *NPC* | *NPC* | **NPC** | *I* | *P* | **P** |
| Normalized range voting | *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** |

(b) Destructive control

| | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|
| Copeland$^{\alpha}$ | *NPC* | *NPC* | **NPC** | *P* | *P* | **P** |
| Maximin | *NPC* | *NPC* | **NPC** | *P* | *P* | **P** |
| Plurality | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| 2-Approval | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| 3-Approval | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| $k$-Approval, $k \geq 4$ | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| Veto | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| 2-Veto | *P* | *P* | *P* | *NPC* | *NPC* | **NPC** |
| $k$-Veto, $k \geq 3$ | *P* | *P* | *P* | *NPC* | *NPC* | **NPC** |
| Plurality with runoff | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |
| Veto with runoff | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |
| Condorcet voting | *P* | *P* | **P** | *P* | *I* | **P** |
| Fallback voting | *P* | *P* | **P** | *NPC* | *NPC* | **NPC** |
| Range voting | *P* | *P* | **P** | *P* | *I* | **P** |
| Normalized range voting | *P* | *P* | **P** | *NPC* | *NPC* | **NPC** |

a reasonable decision as to which voting rule to choose, the designer will have to know the computational (and other) properties of these strategic (e.g., control) actions against his or her system for the various voting rules. The more complete our knowledge is about the complexity of control scenarios for the most commonly used voting rules, the better will be the designer's decision and the better will be the multiagent system.

*Overview of the paper:*

Before diving into the technical details of our results, we give an overview of our main contributions in Sect. 2. In Sect. 3, we define the voting rules and control problems to be studied, fix our notation, and give some background on computational complexity. We then study the complexity of various control scenarios for Copeland$^\alpha$ in Sect. 4, maximin in Sect. 5, $k$-veto in Sect. 6, plurality with runoff and veto with runoff in Sect. 7, Condorcet in Sect. 8, fallback in Sect. 9, and for range voting and normalized range voting in Sect. 10. Finally, we conclude in Sect. 11.

## 2 Our main contributions

In the following, we highlight our main contributions in detail and compare them with the related work to demonstrate how our contributions have improved the state of the art in electoral control. Table 1 gives an overview of previously known and our new results on the complexity of control by replacing, adding, and deleting either candidates or voters for numerous voting rules. For the formal definition of voting rules and control scenarios mentioned and for the notation of control problems, such as CCAV, the reader is referred to Sect. 3.

- Faliszewski et al. [24] and Loreggia [40] investigated the complexity of control in Copeland$^\alpha$ elections, leaving open the case of destructive control by replacing voters for any rational $\alpha$, where $0 \leq \alpha \leq 1$. We settle this open problem.
- Faliszewski, Hemaspaandra, and Hemaspaandra [23] and Maushagen and Rothe [45, 47] investigated the complexity of control in maximin elections but focused on standard control types (i.e., on the cases of constructive and destructive control by adding, deleting, and partitioning either candidates or voters). This leaves the corresponding cases of control by replacing 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 [39] and Loreggia et al. [43] focused on control in $k$-veto (see also the work of Maushagen and Rothe [46] on control in veto elections). Open cases are constructive control by replacing voters in $k$-veto elections for $k \geq 2$. We solve these open cases, providing a dichotomy result for $k$-veto with respect to the values of $k$.
- The standard control scenarios were studied by Bartholdi, Tovey, and Trick [7] and Hemaspaandra, Hemaspaandra, and Rothe [33] for Condorcet voting, by Erdélyi et al. [16, 17, 20, 22] for fallback elections, and by Menton [48] for range voting and normalized range voting, leaving open for all these rules the cases of constructive and destructive control by replacing either candidates or voters.
- Finally, we investigate the complexity of control for two common voting rules that, somewhat surprisingly, have not been considered yet in the literature, namely plurality with runoff and veto with runoff.

## 3 Preliminaries

An *election* $E$ is given by a pair $E = (C, V)$, where $C$ is a finite set of *candidates* and $V$ is a finite multiset of *votes*. Voters typically[2] express their preferences over the candidates by linear orders over $C$, such as $c\ b\ a\ d$ for $C = \{a, b, c, d\}$, where the leftmost candidate is the most preferred one by this voter and preference (strictly) decreases from left to right. When a subset $X \subseteq C$ of candidates occurs in a vote (e.g., $c\ X\ d$ for $X = \{a, b\}$), this means that the candidates in $X$ are ranked in this vote according to a fixed order (e.g., assuming the lexicographic order, $c\ X\ d$ stands for $c\ a\ b\ d$). A *voting rule* (or, more technically, a *voting correspondence*) $\tau$ maps each election $(C, V)$ to a subset $W \subseteq C$ of the candidates, called the $\tau$ *winners* (or simply the *winners* if $\tau$ is clear from the context) of $(C, V)$.

For an election $E = (C, V)$ and 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 $X$ (e.g., of candidates or voters), let $|X|$ denote the cardinality of $X$. For ease of exposition, in this paper we exchangeably use the words vote and voter.

Letting $E = (C, V)$ be a given election, we consider the following voting rules.

**Copeland$^\alpha$** For each pairwise comparison between any two candidates, say $a$ and $b$, if $N_E(a, b) > N_E(b, a)$, $a$ receives one point and $b$ zero points. If $N_E(a, b) = N_E(b, a)$, both $a$ and $b$ receive $\alpha$ points, where $\alpha \in [0, 1]$ is a rational number. The *Copeland$^\alpha$ score of any candidate $c$* is the total number of points $c$ receives from all votes in the election, and all candidates with the highest Copeland$^\alpha$ score win.

**Maximin** The *maximin score of a candidate $a \in C$* is defined as $\min_{b \in C \setminus \{a\}} N_E(a, b)$, and all candidates with the highest maximin score wins.

***k*-Approval** Each voter gives one point to every candidate in the top-$k$ positions, and all candidates with the highest score win. In particular, 1-approval is often referred to as *plurality voting* in the literature.

***k*-Veto** A candidate gains a point from each vote in which he or she is ranked higher than in the last $k$ positions (i.e., the candidates in the last $k$ positions are vetoed), and all candidates with the highest score win. In particular, 1-veto is simply referred to as *veto*.

**Plurality with Runoff (PRun)** Each voter only approves of his or her 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 two who receive the, respectively, 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

[2] Some voting rules, such as fallback voting, require a different input format to specify votes, as will be explained below.

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 rule is defined similarly to PRun, with a slight difference in the first stage: all candidates except the two candidates who have the least and second-least vetoes are eliminated from the election (again applying a tie-breaking rule if necessary).

**Condorcet** A *Condorcet winner* is a candidate $c$ who beats all other candidates in pairwise contests, i.e., for each other candidate $d$, it holds that $N_E(c, d) > N_E(d, c)$. Note that a Condorcet winner does not always exist, but if there is one, he or she is unique.

**Fallback** In a fallback election $(C, V)$, each voter $v$ submits his or her preferences as a subset of candidates $S_v \subseteq C$ that he or she approves of and, in addition, a strict linear ordering of the approved candidates. For instance, if a voter $v$ approves of the candidates $S_v = \{c_1, ..., c_k\} \subseteq C$ and orders them lexicographically, his or her vote would be denoted as $c_1 \cdots c_k \mid C \setminus S_v$. Let $score_{(C,V)}(c) = |\{v \in V \mid c \in S_v\}|$ be the *number of approvals of* $c$ and $score^i_{(C,V)}(c)$ be the *number of level* $i$ *approvals of* $c$ (i.e., the number of voters who approve of $c$ and rank $c$ in their top $i$ positions). For convenience, let $score^0_{(C,V)}(c) = 0$ for every $c \in C$. The *fallback winner(s)* will then be determined as follows:

1. A candidate $c$ is a *level* $\ell$ *winner* if $score^{\ell}_{(C,V)}(c) > |V|/2$. Letting $i$ be the smallest integer such that there is a level $i$ winner, all candidates with the most level $i$ approvals win.
2. If there is no fallback winner on any level, all candidates with the most approvals win.

**Range Voting** Instead of a linear order over the $m$ candidates, each voter is associated with a size-$m$ vector $v \in \{0, 1, \ldots, k\}^m$ describing the points the voter gives to each candidate. The number $k$ is the maximum number of points a voter can give to a candidate, i.e., in such a *k-range election*, every voter gives at most $k$ points to a candidate. The *k-range-voting winners* are the candidates with the most points in the given $k$-range election. 1-range voting is also known as *approval voting*.

**Table 2** Special cases of the $\tau$-Constructive-Multimode-Control problem studied in this paper

| Problems | Restrictions |
|---|---|
| Adding voters | $\ell_{AC} = \ell_{DC} = \ell_{DV} = 0, D = \emptyset$ |
| Adding candidates | $\ell_{DC} = \ell_{AV} = \ell_{DV} = 0, W = \emptyset$ |
| Deleting voters | $\ell_{AC} = \ell_{DC} = \ell_{AV} = 0, D = W = \emptyset$ |
| Deleting candidates | $\ell_{AC} = \ell_{AV} = \ell_{DV} = 0, D = W = \emptyset$ |
| Replacing voters | $\|V'\| = \|W'\|, \ell_{AV} = \ell_{DV}, \ell_{AC} = \ell_{DC} = 0, D = \emptyset$ |
| Replacing candidates | $\|C'\| = \|D'\|, \ell_{AC} = \ell_{DC}, \ell_{AV} = \ell_{DV} = 0, W = \emptyset$ |

**Normalized Range Voting** Similarly to range voting, each voter is associated with a size-$m$ vector $v \in \{0, 1, \dots, k\}^m$. Additionally, each voter's vote is normalized to the range of 0 to $k$ in the following way. For each candidate $c$, let $s$ be the number of points this candidate gains from the voter and $s_{\min}$ and $s_{\max}$ be the minimal and maximal score the voter gives to any candidate. Then the normalized score that $v$ gives to $c$ is $\frac{k(s-s_{\min})}{s_{\max}-s_{\min}}$. Note that if $s_{\max} = s_{\min}$, the voter is indifferent to all candidates and can therefore be ignored. Again, the *k-normalized-range-voting winners* are the candidates with the most normalized points in the given $k$-range election.

We study various control problems that can be considered as special cases of the following problem [23], which is defined for a given voting rule $\tau$.

$\tau$-Constructive-Multimode-Control

**Input:** An election $(C \cup D, V \cup W)$ with a set $C$ of (registered) candidates,[3] a set $D$ of as yet unregistered candidates, a list $V$ of registered voters, a list $W$ of as yet unregistered voters, a distinguished candidate $c \in C$, and four nonnegative integers $\ell_{AV}$, $\ell_{DV}$, $\ell_{AC}$, and $\ell_{DC}$, with $\ell_{AV} \le |W|$, $\ell_{DV} \le |V|$, $\ell_{AC} \le |D|$, and $\ell_{DC} \le |C|$.

**Question:** Are there $V' \subseteq V$, $W' \subseteq W$, $C' \subseteq C \setminus \{c\}$, and $D' \subseteq D$ such that $|V'| \le \ell_{DV}$, $|W'| \le \ell_{AV}$, $|C'| \le \ell_{DC}$, $|D'| \le \ell_{AC}$, and $c$ is a $\tau$ winner of the election $((C \setminus C') \cup D', (V \setminus V') \cup W')$?

We may sometimes omit mentioning explicitly that these candidates are registered.

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 $\tau$ winner in $((C \setminus C') \cup D', (V \setminus V') \cup W')$.

We will study several special cases or restricted versions of multimode control, such as adding, deleting, or replacing either candidates or voters. Table 2 gives an overview of the restrictions compared to the general multimode control problem.

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 $\ell_{AV} = \ell_{DV}$.

As mentioned in the introduction, since the seminal work of Bartholdi, Tovey, and Trick [7] control by *adding* and *deleting* candidates or voters has been extensively studied in the literature (see, e.g., [11, 17, 34, 44, 49, 60, 62]). However, the complexity of control by *replacing* candidates or voters has been introduced and studied just recently by Loreggia et al. [40–43].

We remark that our proofs are based on the nonunique-winner model but can be modified to work for the unique-winner model of the control problems as well.[3]

We assume the reader to be familiar with the basics of complexity theory, such as the complexity classes P and NP and the notions of NP-hardness and NP-completeness under (polynomial-time many-one) reductions. We refer to Tovey's tutorial [58] for a concise introduction to complexity theory and to the books by Arora and Barak [2], Garey and Johnson [27], and Rothe [56] for more comprehensive discussions.

We call a voting rule *immune* to a type of control if it is never possible for the chair to reach his or her goal by this control action; otherwise, the voting rule is said to be *susceptible* to this control type. A susceptible voting rule is said to be *vulnerable* to this control type if the associated control problem is in P, and it is said to be *resistant* to it if the associated control problem is NP-hard. Note that all considered control problems are easily seen to be in NP, so any resistance result immediately implies NP-completeness, and we only provide the NP-hardness proofs since membership of these problems in NP is easy to check. Our NP-hardness results are mainly based on reductions from the Restricted-Exact-Cover-By-3-Sets (RX3C) problem [29] and the Hitting-Set problem [37]:

---

Restricted-Exact-Cover-By-3-Sets (RX3C)

---

**Input:** A set $U = \{u_1, \dots, u_{3\kappa}\}$ and a collection $\mathscr{S} = \{S_1, \dots, S_{3\kappa}\}$ of 3-element subsets of $U$ such that each $u \in U$ occurs in exactly three subsets $S \in \mathscr{S}$.

**Question:** Does $\mathscr{S}$ contain an exact 3-set cover for $U$, i.e., a subcollection $\mathscr{S}' \subseteq \mathscr{S}$ such that every element of $U$ occurs in exactly one member of $\mathscr{S}'$?

---

If we do not request every $u \in U$ to occur in exactly three elements of $\mathscr{S}$ in the RX3C problem, we obtain the generalized X3C problem.

---

Hitting-Set

---

**Input:** A set $U = \{u_1, \dots, u_s\}$ with $s \geq 1$, a family $\mathscr{S} = \{S_1, \dots, S_t\}$ of nonempty subsets $S_i \subseteq U$, and an integer $\kappa$ with $1 \leq \kappa \leq s$.

**Question:** Is there a subset $U' \subseteq U, |U'| \leq \kappa$, such that each $S_i \in \mathscr{S}$ is *hit* by $U'$ (i.e., $S_i \cap U' \neq \emptyset$ for all $S_i \in \mathscr{S}$)?

---

Note further that all voting rules considered here are susceptible to the control scenarios we study. Since the corresponding proofs can be easily obtained by appropriate examples, we will omit them in most cases. The only exceptions are Condorcet and range voting: While among the voting rules we consider these two are the only ones that are immune to some of the standard control scenarios (namely, to constructive control by

[3] In the *nonunique-winner model*, for a constructive (respectively, destructive) control action to be successful, it is enough to make the distinguished candidate $c$ a winner, possibly among others, of the resulting election (respectively, it must be ensured that $c$ is not even a winner), whereas in the *unique-winner model*, a constructive (respectively, destructive) control action is considered to be successful only when $c$ alone wins (respectively, it is enough to ensure that $c$ is not the only winner).

**Table 3** Complexity of control for Copeland$^\alpha$. Our results are in boldface. "NPC" stands for "NP-complete" and "P" stands for "polynomial-time solvable"

| CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *NPC* | *NPC* | *NPC* | *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** | *P* | *P* | **P** |

adding candidates [7, 48] and to destructive control by deleting candidates [33, 48]), we will explicitly show that susceptibility holds in these control scenarios for Condorcet (see Example 1) and range voting (see Example 2).

Assuming that the reader is familiar with graph theory (see also the books by Bang-Jensen and Gutin [4] and West [59]), we will in some proofs make use of the following problems to show membership in P.

Integral-Minimum-Cost-Flow (IMCF)

**Input:** A network $G = (V, E)$, capacity functions $b_\alpha, b_\beta : E \to \mathbb{N}_0$, a source vertex $x \in V$, a sink vertex $y \in V \setminus \{x\}$, a cost function $g : E \to N_0$, and an integer $r$.

**Task:** 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) $b_\alpha(u, v) \le f(u, v) \le b_\beta(u, v)$; and (2) for every node $v$ except $x$ and $y$, it holds that $\sum_{(u,v)\in E} f(u, v) = \sum_{(v,u)\in E} f(v, u)$.[4] 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 $\sum_{(x,v)\in E} f(x, v)$.

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

$b$-Edge-Cover ($b$-EC)

**Input:** 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 known to be polynomial-time solvable [26, 30].

## 4 Copeland$^\alpha$ voting

We start by completing our knowledge on control complexity in Copeland$^\alpha$ elections. Previously, Faliszewski et al. [24] and Loreggia [40] investigated the complexity of control in Copeland$^\alpha$ elections, leaving open the cases of destructive control by replacing voters and of constructive and destructive control by replacing candidates. In this section, we fill the gaps. We refer to Table 3 for a summary of our results in this section.

[4] 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.

**Definition 1** **(Lang, Maudet, and Polukarov** [38]) A voting rule satisfies *Insensitivity to Bottom-ranked Candidates (IBC)* if for any election with at least two candidates, the winners do not change after deleting a subset of candidates who are ranked after all other candidates in all votes.

Note that both Copeland$^\alpha$ and maximin satisfy IBC. Loreggia et al. [42, 43] established the following relationship between CCRC and CCDC, and between DCRC and DCDC.

**Lemma 1** **(Loreggia et al.** [42, 43]) *Let $\tau$ be a voting rule 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.*

By Lemma 1 and the facts that Copeland$^\alpha$ satisfies IBC and that, as shown by Faliszewski et al. [24], Copeland$^\alpha$-CCDC is NP-hard for any rational $\alpha$ with $0 \le \alpha \le 1$, we have the following result.

**Corollary 1** *For any rational $\alpha$ with $0 \le \alpha \le 1$, Copeland$^\alpha$-CCRC is* NP-*complete.*

However, for each rational $\alpha$ with $0 \le \alpha \le 1$, Copeland$^\alpha$-DCDC is *not* NP-hard but in P [24], so Lemma 1 does not imply NP-hardness of Copeland$^\alpha$-DCRC. In fact, we now show that this problem can be solved in polynomial time.

**Theorem 1** *For any rational $\alpha$ with $0 \le \alpha \le 1$, Copeland$^\alpha$-DCRC is in* P.

***Proof*** To show membership in P, we will provide an algorithm that runs in polynomial time. Given a Copeland$^\alpha$-DCRC instance $((C \cup D, V), c, \ell)$, we first check the trivial case, and immediately accept if $c$ is already not winning the election $(C, V)$. Otherwise, for any two candidates $c_1, c_2 \in C \cup D$, let $\text{Score}(c_1, c_2)$ be the number of points $c_1$ receives by $c_2$'s presence in the election (i.e., $\text{Score}(c_1, c_2) = 1$ if $N_{(C \cup D, V)}(c_1, c_2) > N_{(C \cup D, V)}(c_2, c_1)$, $\text{Score}(c_1, c_2) = \alpha$ if $N_{(C \cup D, V)}(c_1, c_2) = N_{(C \cup D, V)}(c_2, c_1)$, and $\text{Score}(c_1, c_2) = 0$ otherwise).[5] We now try to find a candidate $d \in (C \cup D) \setminus \{c\}$ and an integer $\ell'$ with $1 \le \ell' \le \ell$ so that $d$ beats $c$ by replacing $\ell'$ candidates. For a pair $(d, \ell')$, we can check if this is possible in polynomial time in the following way. Firstly, we compute $\text{Score}(c, e)$ and $\text{Score}(d, e)$ for every $e \in (C \cup D) \setminus \{c, d\}$. Then we sort $C \setminus \{c, d\}$ in decreasing order according to $\text{Score}(c, e) - \text{Score}(d, e)$ for each candidate $e \in C \setminus \{c, d\}$ and let $C' \subseteq C \setminus \{c, d\}$ contain the first $\ell'$ candidates according to this ordering. Furthermore, we sort $D \setminus \{d\}$ in decreasing order according to $\text{Score}(d, e) - \text{Score}(c, e)$ and let $D' \subseteq D \setminus \{d\}$ contain the first $\ell'$ candidates according to this ordering if $d \notin D$ and the first $\ell' - 1$ candidates according to this ordering if $d \in D$. We then check if $c$ is not winning in $((C \setminus C') \cup D' \cup \{d\}, V)$.

Correctness of the algorithm follows from the fact that we iterate over all possible candidates that can prevent $c$ from winning and all possible numbers of replacements we may need to this end, and then check whether we can be successful by adding and deleting the most optimal candidates in regards to how they affect the points balance of $c$ and the candidate that should beat $c$ after this replacement.

To see that the above algorithm runs in polynomial time, note that we can iterate over all pairs of candidates and replacements in $O(|C \cup D|\ell)$ time and checking whether a pair

[5] Note that the value of $\text{Score}(c_1, c_2)$ does not depend on any other candidates in the election.

is successful takes $O(|C|\log(|C|) + |D|\log(|D|))$ time for sorting and choosing the subsets and polynomial time for winner determination. □

It remains to handle the case of destructive control by replacing voters. We solve it in the following theorem.

**Theorem 2** *For any rational $\alpha$ with $0 \leq \alpha \leq 1$, Copeland$^\alpha$-DCRV is* NP-*complete*.

***Proof*** Our proof is a slight modification of the proof of Theorem 4.17 (showing that for every rational number $\alpha$ such that $0 \leq \alpha \leq 1$, Copeland$^\alpha$-CCAV is NP-complete) given by Faliszewski et al. [24], with the only difference that there are a number of new registered votes. In particular, from an instance $(U, \mathscr{S})$ of the RX3C problem, it is shown by Faliszewski et al. [24] that an instance of CCAV with the following property can be constructed in polynomial time.[6] Let $|U| = |\mathscr{S}| = 3\kappa$. The candidate set is

$$C = U \cup \{p, r, s\} \cup D,$$

where $D$ is a set of $t$ padding candidates with $t$ a sufficiently large integer but bounded by a polynomial in $\kappa$ (e.g., $t = 9(\kappa + 1)^3$). The multiset $V$ of registered votes are constructed so that, with respect to these registered votes, the Copeland$^\alpha$ scores of $p$ is $t$, of $r$ is $t + 3\kappa$, and of every other candidate is at most $t - 1$. Moreover, it holds that

- $N_{(C,V)}(s, p) - N_{(C,V)}(p, s) = \kappa - 1$,
- $N_{(C,V)}(r, u) - N_{(C,V)}(u, r) = \kappa - 3$ for every $u \in U$, and
- $|N_{(C,V)}(c, c') - N_{(C,V)}(c', c)| \geq \kappa + 1$ for all other pairs of candidates $c$ and $c'$ in $C$. ($|N_{(C,V)}(c, c') - N_{(C,V)}(c', c)|$ is the absolute value of $N_{(C,V)}(c, c') - N_{(C,V)}(c', c)$.)

We refer to [24] for the details of how these votes are created. In addition to the above registered votes, we add the following registered votes. First, for every two candidates $c, c' \in C$ such that $N_{(C,V)}(c, c') - N_{(C,V)}(c', c) \geq \kappa + 1$, we add $2\kappa$ registered votes, among which $\kappa$ of them are of the form $c\ c'\ \overrightarrow{C \setminus \{c, c'\}}$ and the other $\kappa$ of them are of the form $\overleftarrow{C \setminus \{c, c'\}}\ c\ c'$, where $\overleftarrow{C \setminus \{c, c'\}}$ is the reversal of $\overrightarrow{C \setminus \{c, c'\}}$. Let $V_1$ be the multiset of the above newly added votes. Then we add a multiset $V_2$ of $\kappa$ votes, each of which ranks $r$ in the top, ranks $p$ in the last place, and ranks $s$ just before $p$. (Other candidates are ranked arbitrarily between $r$ and $s$.) For notational brevity, let us redefine $V := V \cup V_1 \cup V_2$ as the multiset of all registered votes hereinafter in the proof. Then it is fairly easy to check that the following conditions hold.

- The Copeland$^\alpha$ scores of all candidates remain the same as before the creation of $V_1 \cup V_2$;
- $N_{(C,V)}(s, p) - N_{(C,V)}(p, s) = 2\kappa - 1$;
- $N_{(C,V)}(r, u) - N_{(C,V)}(u, r) = 2\kappa - 3$ for every $u \in U$; and
- $|N_{(C,V)}(c, c') - N_{(C,V)}(c', c)| \geq 2\kappa + 1$ holds for all other pairs of candidates $c$ and $c'$ not specified above.

[6] The reduction in [24] is in fact from the X3C problem, which is a generalization of RX3C where the restriction that every $u \in U$ occurs in exactly three elements of $\mathscr{S}$ is dropped.

**Table 4** Complexity of control for maximin. Our results are in boldface. "NPC" stands for "NP-complete" and "P" stands for "polynomial-time solvable"

| CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *NPC* | *NPC* | **NPC** | *NPC* | *P* | **NPC** | *NPC* | *NPC* | **NPC** | *P* | *P* | **P** |

The unregistered votes are constructed according to $\mathscr{S}$. Precisely, for every $S \in \mathscr{S}$, there is an unregistered vote with the following preference:

$$p\ (U \setminus S)\ r\ S\ (C \setminus (\{p, r, s\} \cup U))\ s.$$

Let $W$ denote the set of all unregistered votes. Additionally, we set $\ell = \kappa$. Finally, we let $r$ be the distinguished candidate (who is the current winner).

We move on to the proof for the equivalence of the two instances.

($\Rightarrow$) Assume that $U$ admits an exact set cover $\mathscr{S}' \subseteq \mathscr{S}$. Let $W' \subseteq W$ be the set of unregistered votes corresponding to $\mathscr{S}'$. We claim that after replacing $V_2$ with $W'$, $r$ is not a winner anymore. Let $E = (C, V \setminus V_2 \cup W')$. Observe that if $|N_{(C,V)}(c, c') - N_{(C,V)}(c', c)| > 2\kappa + 1$, then $c$ still beats $c'$ in $E$, as we replace at most $\kappa$ votes. As $\mathscr{S}'$ is an exact set cover of $U$, for every $u \in U$, there are exactly $\kappa - 1$ votes in $W'$ which rank $u$ above $r$. In addition, as $N_{(C,V)}(r, u) = 2\kappa - 3$ holds for every $u \in U$ and all votes in $V_2$ rank $r$ in the first place, we know that $r$ is beaten by all candidates in $U$ in the election $E$. So, the Copeland$^\alpha$ score of $r$ decreases to $t$ in $E$. Moreover, as all votes in $V_2$ rank $s$ above $p$, all votes in $W'$ rank $p$ in the top, and $N_{(C,V)}(s, p) - N_{(C,V)}(p, s) = 2\kappa - 1$, we have that $N_E(p, s) - N_E(s, p) = 1$, i.e., in the election $E$ the candidate $p$ beats $s$. Therefore, the Copeland$^\alpha$ score of $p$ in $E$ increases to $t + 1$. Clearly, $r$ is no more a winner in $E$.

($\Leftarrow$) Assume that there are $V' \subseteq V$ and $W' \subseteq W$ such that $|V'| = |W'| \leq \kappa$, and $r$ is not a winner in the election $E = (C, V \setminus V' \cup W')$. As pointed out above, if $N_{(C,V)}(c, c') - N_{(C,V)}(c', c) \geq 2\kappa + 1$, then $c$ still beats $c'$ after replacing at most $\kappa$ votes. This means that replacing at most $\kappa$ votes can only change the Copeland$^\alpha$ scores of $p$, $s$, and $r$ (see the above conditions). More importantly, between $p$ and $s$, as all unregistered votes rank $s$ in the last place, replacing at most $\kappa$ votes does not increase the score of $s$. Moreover, as $|N_{(C,V)}(r, c') - N_{(C,V)}(c', r)| \geq 2\kappa + 1$ for all other candidates $c' \in C \setminus U$, replacing at most $\kappa$ votes can only change the head-to-head comparisons between $r$ and candidates in $U$. This implies that in the election $E$, $r$ has Copeland$^\alpha$ score at least $t$. Therefore, we know that $p$ is the only candidate that prevents $r$ from winning in $E$. Then, as $|N_{(C,V)}(p, c) - N_{(C,V)}(c, p)| \geq 2\kappa - 1$ for all candidates $c \in C \setminus \{p, s\}$, the Copeland$^\alpha$ score of $p$ in $E$ can be at most $t + 1$. This implies that the Copeland$^\alpha$ score of $r$ in $E$ is exactly $t$. As the comparisons between $r$ and any of the other candidates in $C \setminus U$ do not change by replacing at most $\kappa$ votes, this is possible only when $r$ is beaten by everyone in $U$ in the election $E$. This means that for every $u \in U$, there are at least $\kappa - 1$ votes in $W'$ which rank $u$ above $r$. Due to the construction of the unregistered votes, for each $S \in \mathscr{S}$ that corresponds to an unregistered vote ranking $u$ above $r$, it holds that $u \notin S$. As this holds for all $u \in U$ and $W'$ contains at most $\kappa$ votes, we can conclude that the subcollection of $\mathscr{S}$ corresponding to $W'$ is an exact set cover of $U$. □

**Table 5** Head-to-head comparisons of candidates with respect to the registered votes in the proof of Theorem 3. * means that the value does not have any impact on the correctness of the reduction

| | $c$ | $d$ | $u \in U$ | maximin score |
|---|---|---|---|---|
| $c$ | – | $2\kappa$ | $2\kappa$ | $2\kappa$ |
| $d$ | $3\kappa + 1$ | – | $4\kappa + 1$ | $3\kappa + 1$ |
| $u' \in U$ | $3\kappa + 1$ | $\kappa$ | $*$ | $\leq \kappa$ |

## 5 Maximin voting

Let us now turn to maximin voting. Faliszewski, Hemaspaandra, and Hemaspaandra [23] have already investigated the complexity of constructive and destructive control by adding and deleting either candidates or voters. Maushagen and Rothe [45, 47] settled all cases of constructive and destructive control by partitioning either candidates or voters. We will complete the picture on control in maximin elections by providing results on constructive and destructive control by replacing either candidates or voters. Our results in this section are summarized in Table 4.

It is known that constructive control by deleting candidates for maximin is polynomial-time solvable [23]. Hence, assuming P ≠ NP, Lemma 1 cannot be used to obtain NP-hardness of Maximin-CCRC. However, as stated below, Loreggia [42] introduced another useful lemma.

**Definition 2** A voting rule is said to be *unanimous* if whenever the same candidate is ranked in the top position in all votes, this candidate wins.

**Lemma 2** **(Loreggia** [42]) *Let $\tau$ be an unanimous voting rule 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) Maximin-CCAC is NP-complete [23], we have

**Corollary 2** *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 [23]. In the following, for two subsets $A$ and $B$ of candidates and a linear order over candidates, $A$ $B$ means that $a$ $b$ for every $a \in A$ and $b \in B$.

**Theorem 3** *Maximin-CCRV and Maximin-DCRV are* NP-*complete.*

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

- there are $3\kappa + 1$ votes of the form $d$ $U$ $c$;
- there are $\kappa$ votes of the form $c$ $U$ $d$; and

- there are $\kappa$ votes of the form $c\ d\ U$.

Let $V$ denote the multiset of the above $5\kappa + 1$ registered votes. The head-to-head comparisons of candidates (i.e., $|N_{(C,V)}(c, c')|$ for all $c, c' \in C$) and their maximin scores with respect to the registered votes are summarized in Table 5.

Moreover, for each $S \in \mathscr{S}$, we create an unregistered vote in $W$ of the form

$$(U \setminus S)\ c\ S\ d.$$

We use $v(S)$ to denote this vote. Finally, we set $\ell = \kappa$, i.e., we are allowed to replace at most $\kappa$ voters.

The above Maximin-CCRV instance clearly can be constructed in polynomial time. We claim that we can make $c$ the winner of the election by replacing up to $\kappa$ voters if and only if $\mathscr{S}$ contains an exact set cover of $U$.

($\Rightarrow$) Assume that $U$ admits an exact set cover $\mathscr{S}' \subseteq \mathscr{S}$. Let $W' = \{v(S) \mid S \in \mathscr{S}'\}$ be the set of the unregistered votes corresponding to this exact set cover. Clearly, $|W'| = |\mathscr{S}'| = \kappa$. Let $V'$ be a multiset of $\kappa$ registered votes of the form $d\ U\ c$. We claim that $c$ becomes a winner in the election $E' = (C, (V \setminus V') \cup W')$. Let us now analyze the maximin scores of the candidates in $E'$. First, as all votes in $W'$ rank $c$ above $d$, and all votes in $V'$ rank $c$ in the last position, it holds that $N_{E'}(c, d) = 2\kappa - 0 + \kappa = 3\kappa$. As $\mathscr{S}'$ is an exact set cover of $U$, for every candidate $u \in U$ there is exactly one vote, namely, the vote $v(S)$ such that $u \in S$, which ranks $c$ above $u$ and is contained in $V'$. In addition, as all votes in $V'$ rank $c$ in the end, we know that $N_{E'}(c, u) = 2\kappa + 1$ for every $u \in U$. So, the maximin score of $c$ in the election $E'$ increases from $2\kappa$ to $2\kappa + 1$. Now we start the analysis for the candidate $d$. As all votes in $W'$ rank $d$ in the last position and all votes in $V'$ rank $d$ in the first position, the maximin score of $d$ in $E'$ decreases from $3\kappa + 1$ to $2\kappa + 1$. As the maximin score of every candidate $u \in U$ is at most $\kappa$ with respect to $V$, and we are allowed to replace at most $\kappa$ votes, the maximin score of $u$ in $E'$ can be at most $2\kappa$. In summary, $c$ and $d$ are the only two candidates having the maximum maximin score in $E'$, and hence $c$ is a winner in $E'$.

($\Leftarrow$) Assume that there is a subset $V' \subseteq V$ and a subset $W' \subseteq W$ such that $|V'| = |W'| \leq \kappa$ and $c$ wins the election $(C, (V \setminus V') \cup W')$. Let $\hat{E} = (C, (V \setminus V') \cup W')$, and let $\mathscr{S}' = \{S \in \mathscr{S} \mid v(S) \in W'\}$. An important observation is that the maximin score of $c$ in $\hat{E}$ can be at most $2\kappa + 1$. In fact, no matter which up to $\kappa$ unregistered votes are included in $W'$, there is at least one candidate $u \in U$ such that there is at most one unregistered vote in $W'$ which ranks $c$ above $u$, implying that $N_{\hat{E}}(c, u) \leq 2\kappa + 1$. From this observation, we know that $V'$ must consist of exactly $\kappa$ votes and, moreover, all votes in $V'$ must rank $d$ above $c$, since otherwise $d$ would have maximin score at least $3\kappa + 1 - (\kappa - 1) = 2\kappa + 2$ in $\hat{E}$, contradicting that $c$ is a winner in $\hat{E}$. This means that $V'$ consists of exactly $\kappa$ registered votes of the form $d\ U\ c$. Now the maximin score of $d$ in $\hat{E}$ is determined as $3\kappa + 1 - \kappa = 2\kappa + 1$. We claim that $\mathscr{S}'$ is an exact set cover of $U$. For the sake of contradiction, assume that this is not the case. Then there is a candidate $u \in U$ such that none of the sets in $\mathscr{S}$ contains $u$. In light of the above construction of the unregistered votes, all the $\kappa$ votes in $W'$ rank this particular candidate $u$ above $c$, resulting in the maximin score of $c$ in $\hat{E}$ being at most $2\kappa$, contradicting that $c$ is a winner in $E'$.

The destructive version works identically, except that the first group of votes (i.e., votes of the type $d\ U\ c$) consists of $3\kappa$ registered votes and the distinguished candidate is $d$. In this case, one can check that, similarly to the analysis in the above ($\Rightarrow$) direction, after replacing $\kappa$ registered votes of the form $d\ U\ c$ with $\kappa$ unregistered votes corresponding to an exact set cover of $U$, the maximin scores of $c$ and $d$ are, respectively, $2\kappa + 1$ and $2\kappa$, leading to $d$ not being a winner anymore. For the proof of the other direction, one observes that the maximin score of $d$, after replacing at most $\kappa$ votes from $V$ and by as many votes from $W$, is at least $3\kappa - \kappa = 2\kappa$, and the maximin score of every $u \in U$ can be at most $2\kappa$. This means that $c$ is the only candidate that may have maximin score at least $2\kappa + 1$ in the final election. Analogously to the analysis in the above ($\Leftarrow$) direction, we can show that the candidate $c$ achieves the maximin score $2\kappa + 1$ if and only if there exists a set of $\kappa$ unregistered votes corresponding to an exact set cover of $U$. □

It remains to show the complexity of destructive control by replacing candidates for 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 P membership of a more general problem called $\tau$-Exact-Destructive-Control-by-Adding-and-Deleting-Candidates, denoted by $\tau$-EDCAC+DC, where $\tau$ is a voting rule. In particular, this problem is a variant of $\tau$-Destructive-Multimode-Control, where $\ell_{\mathrm{AV}} = \ell_{\mathrm{DV}} = 0$, $W = \emptyset$. Moreover, it must hold that in the solution $|C'| = \ell_{\mathrm{DC}}$ and $|D'| = \ell_{\mathrm{AC}}$ (i.e., the chair deletes *exactly* $\ell_{\mathrm{DC}}$ candidates and adds *exactly* $\ell_{\mathrm{AC}}$ candidates). Note that the number of candidates added and the number of candidates deleted do not have to be the same.

**Theorem 4** *Maximin-EDCAC+DC is in* P.

***Proof*** Our input is a Maximin-EDCAC+DC instance as defined above. Suppose that the chair adds exactly $\ell_{\mathrm{AC}}$ candidates from $D$ and deletes exactly $\ell_{\mathrm{DC}}$ candidates from $C$. Note that $\ell_{\mathrm{DC}} < |C|$ since the chair must not delete the distinguished 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 $k \in \{0, 1, \dots, |V| - 1\}$. Note that we may restrict ourselves to values $k \leq \lceil |V|/2 \rceil - 1$. Otherwise, $c$ does not lose any pairwise comparison and is a weak Condorcet winner and thus a maximin winner.

In more detail, the algorithm first tries to find 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 try to find 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 $N(c', d) > k$.[7] We go to the next step.
2. Check whether $\ell_{\mathrm{DC}} \leq |C| - 1 - |C \cap \{c', d\}|$ and $\ell_{\mathrm{AC}} \geq |D \cap \{c', d\}|$. If this is the case, proceed with the next step. Otherwise, we reject because there is no way for the chair to keep both $c'$ and $d$ in (or to add them to) the final election.

[7] 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 $(c', k)$.

3. Let $C_1 = \{c'' \in C \setminus \{c, c', d\} : N(c', c'') \le k\}$. The candidates in $C_1$ must all be deleted in order to keep the maximin score of $c'$ higher than $k$. If $|C_1| > \ell_{\mathrm{DC}}$, 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 $\ell_{\mathrm{DC}}$ 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 $|D_1| < \ell_{\mathrm{AC}} - |D \cap \{c', d\}|$, 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.

If the given instance is a YES-instance, at least one such triple $(c', k, d)$ must lead to the algorithm accepting it. However, if we are given a NO-instance, the algorithm must reject. Finally, 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. □

Note that Maximin-DCRC is polynomial-time Turing-reducible to Maximin-EDCAC+DC. Then, from Theorem 4 we obtain the following result.

**Corollary 3** *Maximin-DCRC is in* P.

Theorem 4 generalizes the polynomial-time solvability results for Maximin-DCAC and Maximin-DCDC obtained by Faliszewski et al. [23]. We also point out that Faliszewski, Hemaspaandra, and Hemaspaandra [23] showed that Maximin-CCAC$_{\mathrm{u}}$+DC is polynomial-time solvable, where the subscript u refers to control by adding an *unlimited* number of candidates, as originally defined by Bartholdi, Tovey, and Trick [7]: In this case, the chair is allowed to add as many unregistered candidates as desired but can only delete a limited number of candidates.

## 6 *k*-veto

Turning now to $k$-veto and starting with control by replacing voters, it is known that Veto-CCRV and $k$-Veto-DCRV for all possible $k$ are polynomial-time solvable [43], which leaves open the complexity of $k$-Veto-CCRV for $k \ge 2$. We complement these results by showing that 2-Veto-CCRV is polynomial-time solvable and $k$-Veto-CCRV is NP-complete for $k \ge 3$, achieving a dichotomy result for constructive control by replacing voters in $k$-veto with respect to the values of $k$. Our results in this section are summarized in Table 6.

As a notation, let $V^c$ ($W^c$) be the set consisting of all voters in $V$ ($W$) vetoing $c$, and define $V^{\neg c} = V \setminus V^c$ ($W^{\neg c} = W \setminus W^c$).

**Theorem 5** 2-*Veto-CCRV is in* P.

***Proof*** Let $(C, V \cup W)$, $\ell$, and $c \in C$ be the components of a given 2-Veto-CCRV instance, as described in Sect. 3. Recall that $c$ is the distinguished candidate in the input. Our algorithm distinguishes the following cases:

**Case 1:** $|V^c| \le \min(\ell, |W| - |W^c|)$.

**Table 6** Complexity of control for $k$-veto. Our results are in boldface. "NPC" stands for "NP-complete" and "P" stands for "polynomial-time solvable"

| | CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $k = 1$ | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* | *P* | *P* | *P* | *NPC* | *NPC* | *NPC* |
| $k = 2$ | *P* | *P* | **P** | *NPC* | *NPC* | **NPC** | *P* | *P* | *P* | *NPC* | *NPC* | **NPC** |
| $k \geq 3$ | *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** | *P* | *P* | *P* | *NPC* | *NPC* | **NPC** |

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 the same number of unregistered votes not vetoing $c$.

**Case 2:** $|W| - |W^c| \leq \min(\ell, |V^c|)$.

In this case, the optimal choice for the chair is to replace $|W| - |W^c|$ 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 $\ell - |W| + |W^c|$ $V$-voters vetoing $c$ by the same number of $W$-voters vetoing $c$. Anyway, $c$ has exactly

$$v_c = |V^c| - (|W| - |W^c|) = |(V \cup W)^c| - |W|$$

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 $\max(0, |V^c| - \ell)$ and at most $|V^c| - |W| + |W^c|$ among them belong to $V^c$. We sequentially check for the exact number $\ell'$, where

$$\max(0, |V^c| - \ell) \leq \ell' \leq |V^c| - |W| + |W^c|,$$

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 checked numbers leads to a YES answer.

In the following, we transform the instance into an equivalent $b$-EC instance in polynomial time, thus providing a reduction from 2-Veto-CCRV to $b$-EC.

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 nondistinguished 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 $b_\alpha(d) = v_c - |(V^{\neg c} \cup W^{\neg c})^d|$ 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 $|(V^{\neg c} \cup W^{\neg c})^d|$ 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 $|V^c| - \ell'$ votes.

**Case 3:** $\ell \leq \min(|V^c|, |W| - |W^c|$.

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 have ensured that the final election contains all voters in $V^{\neg c}$, exactly $|V^c| - \ell$ voters in $V^c$, and exactly $\ell$ voters from $W^{\neg c}$. This observation enables us to reduce the 2-Veto-CCRV 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:

- $b_\alpha(c_V) = b_\beta(c_V) = |V^c| - \ell$. This capacity makes sure that exactly $|V^c| - \ell$ 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

$$b_\alpha(d) = \max\left(0, |V^c| - \ell - |(V^{\neg c})^d|\right).$$

The lower bound ensures that in the final election $d$ has at least the same number of vetoes as $c$. Here, $|(V^{\neg c})^d|$ is the number of 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 number of 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 $|V^c|$.

Each subcase can be done in polynomial time. Consequently, the overall algorithm terminates in polynomial time. Since we thus have a polynomial-time reduction from 2-Veto-CCRV to $b$-EC and $b$-EC can be solved in polynomial time, the theorem is proven. □

We fill the complexity gap of 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 NP-hardness proof of constructive control by adding voters for 3-veto due to Lin [39].[8]

**Theorem 6** *For every constant $k \geq 3$, $k$-Veto-CCRV is* NP-*complete.*

***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 RX3C, 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 the set

[8] We remark in passing that Loreggia et al. [43] showed NP-hardness for $k$-Approval-CCRV with $k \leq m - 3$ from which NP-hardness of $k$-Veto-CCRV with $k \geq 3$ immediately follows ($k$-veto and $(m - k)$-approval are the same for constant $m$), but their proof (given in the PhD thesis of Loreggia [42]), which reduces X3C to 3-Approval-CCRV, does not make it clear how the reduction can be adapted to $k$-approval with $k \leq m - 3$ (in particular, since the addition of dummy candidates would also increase $m$).

$\{c, d_1, d_2, d_3\}$ is disjoint from $U$. The distinguished candidate is $c$. For ease of exposition, let $n = 3\kappa$. The multiset $V$ consists of the following $2n - 2\kappa + 3\kappa n$ registered voters:

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

Note that with the registered voters, the distinguished 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 $S \in \mathscr{S}$, 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 $\mathscr{S}' \subseteq \mathscr{S}$. After replacing the $\kappa$ votes corresponding to $\mathscr{S}'$ from $W$ with $\kappa$ 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 multisets 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 NP-hardness of $k$-Veto-CCRV for $k \geq 4$, we additionally create $k - 3$ dummy candidates being vetoed by every vote. The correctness argument is analogous.

Turning now to control by replacing candidates in $k$-veto, Loreggia et al. [43] solved the two cases of constructive and destructive control by replacing candidates for veto only (i.e., for $k$-veto with $k = 1$). Note that Loreggia et al. [43] solved both cases for $k$-approval for any $k$. However, this does not solve these two cases for $k$-veto since their proofs (which again can be found in the PhD thesis of Loreggia [42]) rely on the fact that $k$-approval satisfies IBC, but $k$-veto does not.[9] We solve these two cases, CCRC and DCRC, for $k$-veto with $k \geq 2$ in Theorems 7 and 8.

**Theorem 7** *For every constant $k \geq 2$, $k$-Veto-CCRC is* NP-*complete.*

[9] Indeed, to see that $k$-veto does not satisfy IBC, consider the set $C = \{a, b, c_1, \ldots, c_k\}$ of candidates and let there be only one voter with vote $a\ b\ c_1\ \cdots\ c_k$. Then $a$ and $b$ win the election under $k$-veto, but if we remove the bottom ranked candidate $c_k$, only $a$ wins the election alone, so the set of winners can be changed by removing a bottom-ranked candidate.

***Proof*** To prove NP-hardness of $k$-Veto-CCRC for $k \geq 2$, we will modify the reduction provided by Lin [39] to prove that $k$-Veto-CCAC and $k$-Veto-CCDC are NP-hard. Since his reduction was designed so as to prove both cases at once but we only need the "adding candidates" part, we will simplify the reduction.

Let $(U, \mathscr{S}, \kappa)$ be an instance of Hitting-Set with $U = \{u_1, \dots, u_s\}$, $s \geq 1$, $\mathscr{S} = \{S_1, \dots, S_t\}$, $t \geq 1$, and integer $\kappa$, $1 \leq \kappa < s$ (without loss of generality, we may assume that $\kappa < s$ since $(U, \mathscr{S}, \kappa)$ is trivially a YES-instance if $\kappa \geq s$).

We construct an instance $((C \cup U, V), c, \kappa)$ of $k$-Veto-CCRC with candidates $C = \{c, d\} \cup C' \cup X \cup Y$, where

$$C' = \{c'_1, \dots, c'_{k-1}\},$$
$$X = \{x_1, \dots, x_{k-1}\}, \text{ and}$$
$$Y = \{y_1, \dots, y_\kappa\},$$

and unregistered candidates $U$. Let $V$ contain the following votes:

- $(t + 2s)(s - \kappa + 1)$ votes of the form $Y \cdots c\ C'$;
- $(t + 2s)(s - \kappa + 1) - s + \kappa$ votes of the form $Y \cdots d\ X$;
- for each $i$, $1 \leq i \leq t$, one vote of the form $Y \cdots c\ X\ S_i$;
- for each $i$, $1 \leq i \leq s$, one vote of the form $Y \cdots d\ X\ u_i$; and
- for each $i$, $1 \leq i \leq s$, $(t + 2s)(s - \kappa + 1) + \kappa$ votes of the form $Y \cdots c\ U \setminus \{u_i\}\ X\ u_i$.

Let $M = (t + 2s)(s - \kappa + 1)$. Without the unregistered candidates, vetoes are assigned to the other candidates as follows:

| candidates in $C$ | $c$ | $d$ | $c' \in C'$ | $x \in X$ | $y \in Y$ |
|---|---|---|---|---|---|
| number of vetoes | $M(s + 1) + s\kappa + t$ | $M + \kappa$ | $M$ | $M(s + 1) + \kappa(s + 1) + t$ | 0 |

We show that $(U, \mathscr{S}, \kappa)$ is a YES-instance of Hitting-Set if and only if $c$ can be made a $k$-veto winner of the election by replacing $\kappa$ candidates from $C$ with candidates from $U$.

($\Rightarrow$) Assume there is a hitting set $U' \subseteq U$ of $\mathscr{S}$ of size $\kappa$ (since $\kappa < s$, if $U'$ is a hitting set of size less than $\kappa$, we fill $U'$ up by adding arbitrary candidates from $U \setminus U'$ to $U'$ until $|U'| = \kappa$). We then replace the candidates from $Y$ with the candidates from $U'$. Since $c$, $d$, and candidates from $C'$ have $(t + 2s)(s - \kappa + 1)$ vetoes and candidates from $X$ and $U'$ have at least $(t + 2s)(s - \kappa + 1) + \kappa$ vetoes, $c$ is a $k$-veto winner.

($\Leftarrow$) Assume $c$ can be made a $k$-veto winner of the election by replacing $\kappa$ candidates. Since the $\kappa$ candidates from $Y$ have zero vetoes but $c$ has at least one veto, we need to remove each candidate of $Y$ (and no other candidate), and in turn we need to add $\kappa$ candidates from $U$. Note that $c$ cannot have more than $(t + 2s)(s - \kappa + 1)$ vetoes, for otherwise $c$ would lose to the candidates from $C'$. Let $U' \subseteq U$ be the set of $\kappa$ candidates from $U$ that are added to the election. Since $|U'| = \kappa > 0$, $c$ will lose all $s((t + 2s)(s - \kappa + 1) + \kappa)$ vetoes from the last group of voters. Furthermore, in order to tie the candidates in $C'$, $c$ cannot gain any vetoes from the third group of voters. Thus the $\kappa$ added candidates from $U$ need to be a hitting set of $\mathscr{S}$. Also note that with the $\kappa$ added candidates from $U$, $c$ also ties $d$ (who lost $\kappa$ vetoes from the fourth group of voters) and beats the candidates from $X$ and the added candidates from $U$. □

The same result can be shown for destructive control by replacing candidates in $k$-veto elections via a similar proof.

**Theorem 8** *For every constant* $k \geq 2$, *k-*Veto*-DCRC is* NP*-complete.*

***Proof*** As in the proof of Theorem 7, we will prove NP-hardness of $k$-Veto-DCRC, $k \geq 2$, by providing a reduction from Hitting-Set to $k$-Veto-DCRC that is a simplified and slightly modified variant of a reduction used by Lin [39] to show that $k$-Veto-DCAC and $k$-Veto-DCDC are NP-hard.

Let $(U, \mathscr{S}, \kappa)$ be an instance of Hitting-Set with $U = \{u_1, \dots, u_s\}$, $s \geq 1$, $\mathscr{S} = \{S_1, \dots, S_t\}$, $t \geq 1$, and integer $\kappa$, $1 \leq \kappa \leq s$.

We construct an instance $((C \cup U, V), c, \kappa)$ of $k$-Veto-DCRC with candidates $C = \{c, c'\} \cup X \cup Y$, where $X = \{x_1, \dots, x_{k-1}\}$ and $Y = \{y_1, \dots, y_\kappa\}$, and unregistered candidates $U$. Let $V$ contain the following votes:

- $2(s-\kappa) + 2t(\kappa+1) + 4$ votes of the form $\cdots\ c\ Y\ X\ c'$;
- $2t(\kappa+1) + 5$ votes of the form $\cdots\ c'\ X\ c$;
- for each $i$, $1 \leq i \leq t$, $2(\kappa+1)$ votes of the form $\cdots\ c'\ X\ S_i$;
- for each $i$, $1 \leq i \leq s$, two votes of the from $\cdots\ c\ Y\ X\ u_i$;
- for each $i$, $1 \leq i \leq \kappa$, $2(s-\kappa) + 2t(\kappa+1) + 6$ votes of the form $c\ c'\ \cdots\ y_i\ X$; and
- for each $i$, $1 \leq i \leq s$, $2(s-\kappa) + 2t(\kappa+1) + 6$ votes of the form $c\ c'\ \cdots\ u_i\ X$.

In $(C, V)$, $c$ wins the election with $2t(\kappa+1) + 5$ vetoes while $c'$ has $2(s-\kappa) + 4t(\kappa+1) + 4$ vetoes and every other candidate has at least $2(s-\kappa) + 2t(\kappa+1) + 6$ vetoes.

To complete the proof of Theorem 8, we will now show that $(U, \mathscr{S}, \kappa)$ is a YES-instance of Hitting-Set if and only if $c$ can be prevented from being a $k$-veto winner of the election by replacing $\kappa$ candidates from $C$ with candidates from $U$.

($\Rightarrow$) Assume there is a hitting set $U' \subseteq U$ of $\mathscr{S}$ of size $\kappa$ (since $\kappa < s$, if $U'$ is a hitting set of size less than $\kappa$, we again fill $U'$ up by adding arbitrary candidates from $U \setminus U'$ to $U'$ until $|U'| = \kappa$). Replacing the candidates from $Y$ with the candidates from $U'$, $c$ gains $2(s-\kappa)$ vetoes and now has $2(s-\kappa) + 2t(\kappa+1) + 5$ vetoes and $c'$ loses $2t(\kappa+1)$ vetoes and now has $2(s-\kappa) + 2t(\kappa+1) + 4$ vetoes, so $c$ does no longer win the election.

($\Leftarrow$) Assume $c$ can be prevented from being a $k$-veto winner of the election by replacing at most $\kappa$ candidates. We first argue why we must remove all $\kappa$ candidates from $Y$. Firstly, from removing $c'$ from the election, $c$'s strongest rival, $c$ does not gain any vetoes and then there won't be any candidate in the election that can beat $c$. Secondly, removing any candidate in $X$ from the election will lead to $c'$ gaining vetoes (which $c'$ cannot afford) while $c$ can in the best case gain the same number of vetoes as $c$ would gain by replacing candidates from $Y$. Thus removing candidates from $Y$ is the best choice. All $\kappa$ candidates from $Y$ need to be removed, for otherwise $c$ does not gain any vetoes. Then $\kappa$ candidates from $U$ need to be added to the election. Note that $c$ will always gain $2(s-\kappa)$ vetoes from those replacements, which will bring $c$ to $2(s-\kappa) + 2t(\kappa+1) + 5$ vetoes, so every candidate other than $c'$ cannot beat $c$. In order for $c'$ to beat $c$, $c'$ cannot gain any vetoes from the third group of voters. Therefore, for each $S_i \in \mathscr{S}$, at least one $u_j \in S_i$ needs to be added to the election. Thus the $\kappa$ added candidates from $U$ need to correspond to a hitting set of $\mathscr{S}$. □

**Table 7** Complexity of control for plurality with runoff. All results are ours. "NPC" stands for "NP-complete" and "P" stands for "polynomial-time solvable"

| CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **P** | **P** | **P** | **NPC** | **NPC** | **NPC** | **P** | **P** | **P** | **NPC** | **NPC** | **NPC** |

Although we do not focus on parameterized complexity [13, 51] here, we mention in passing that the proofs of Theorems 7 and 8 in fact even show W[2]-hardness of CCRC and DCRC, for $k$-veto with $k \geq 2$.

## 7 Plurality with runoff and veto with runoff

We now turn to plurality with runoff and veto with runoff, two quite common voting rules that proceed in two stages, eliminating the "weakest" candidate(s) in the first stage and then holding a runoff among the two surviving candidates for a winner to emerge. To the best of our knowledge, no results on control in plurality with runoff or veto with runoff are known to date. However, a related work has been done by Guo and Shrestha [31] who studied the complexity of control for two-stage voting rules X Then Y, where X and Y are both voting rules. Particularly, under X Then Y, the rule X is first applied and then all winning candidates under X enter a runoff election whose winners are determined by Y. Plurality (respectively, veto) with runoff can be considered as an X Then Y rule where Y is plurality (respectively, veto), and X is a rule which selects exactly two candidates with the highest plurality score (respectively, with the fewest vetoes). Nevertheless, it should be pointed out that such an X Then Y rule has not been investigated by Guo and Shrestha [31].

Our results in this section are summarized in Table 7.

We first show that the problems CCAV, CCDV, and 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. More 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, $\tau$-Exact Constructive Control by Adding and Deleting Voters, denoted by $\tau$-ECCAV+DV, is polynomial-time solvable, where $\tau$ is either plurality with runoff or veto with runoff. In this exact variant of $\tau$-Constructive-Multimode-Control, we require that the number of added voters and the number of deleted voters are *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 each of CCAV, CCDV, and CCRV is polynomial-time reducible to ECCAV+DV.

For an election $(C, V)$, a candidate $d \in C$, and $\tau \in \{\text{PRun}, \text{VRun}\}$, let $\tau_{(C,V)}(d)$ be the number of voters in $V$ approving $d$ if $\tau$ is PRun, and be the number of voters in $V$ vetoing $d$ if $\tau$ is VRun. In the proof of the following theorem we will show P membership of PRun-ECCAV+DV and VRun-ECCAV+DV by reducing them to the problem Integral-Minimum-Cost-Flow (IMCF), defined in Sect. 3, which is known to be polynomial-time solvable [1].

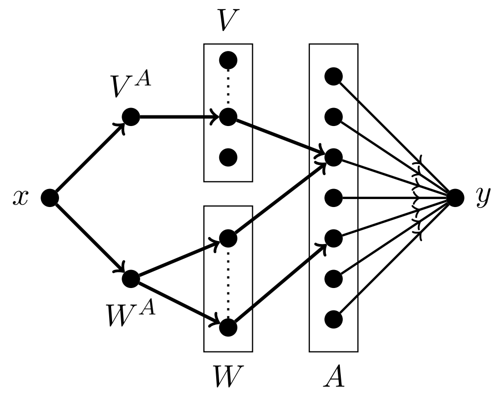


**Fig. 1** An illustration of constructing the IMCF instance in the proof of Theorem 9

**Theorem 9** *For each* $\tau \in \{\text{PRun, VRun}\}$, $\tau$ *-ECCAV+DV is in* P.

***Proof*** Let $(C, V)$, $W$, $c \in C$, $\ell_{\text{AV}}$, and $\ell_{\text{DV}}$ be the components of a given instance as described in the definition of $\tau$-ECCAV+DV. Here, $c$ is the distinguished candidate. We first give the algorithm for $\tau$ being plurality with runoff, and then we discuss how to modify the algorithm for the case where $\tau$ is veto with runoff.

$\tau =$ **PRun.** Our algorithm tries to find a candidate $d \in C \setminus \{c\}$ and four nonnegative integers $\ell^c_{\text{AV}}$, $\ell^d_{\text{AV}}$, $\ell^c_{\text{DV}}$, and $\ell^d_{\text{DV}}$ such that $\ell^c_X + \ell^d_X \leq \ell_X$ for $X \in \{\text{AV, DV}\}$. This candidate $d$ is supposed to be the one who competes with $c$ in the runoff stage. Moreover, $\ell^c_{\text{AV}}$ (respectively, $\ell^d_{\text{AV}}$) is supposed to be the number of voters added from $W$ that approve $c$ (respectively, $d$), and $\ell^c_{\text{DV}}$ (respectively, $\ell^d_{\text{DV}}$) is supposed to be the number of voters deleted from $V$ that approve $c$ (respectively, $d$). Given such a candidate and integers, we determine whether we can add exactly $\ell_{\text{AV}}$ votes from $W$ of which $\ell^c_{\text{AV}}$ (respectively, $\ell^d_{\text{AV}}$) approve $c$ (respectively, $d$), and delete exactly $\ell_{\text{DV}}$ votes from $V$ of which $\ell^c_{\text{DV}}$ (respectively, $\ell^d_{\text{DV}}$) approve $c$ (respectively, $d$). Clearly, the original instance is a YES-instance if and only if at least one of these tests leads to a YES answer. We show how to find the answer to each subinstance in polynomial time. First, we immediately discard a currently tested candidate $d$ if one of the following conditions holds:

- $\ell^c_{\text{DV}} > \tau_{(C,V)}(c)$;
- $\ell^d_{\text{DV}} > \tau_{(C,V)}(d)$;
- $\ell^c_{\text{AV}} > \tau_{(C,W)}(c)$; or
- $\ell^d_{\text{AV}} > \tau_{(C,W)}(d)$.

So let us assume that none of the above conditions holds. Then the number of voters approving $c$ and $d$ in the final election are determined. More precisely, the number of voters approving $e \in \{c, d\}$ is $\tau_{(C,V)}(e) + \ell^e_{\text{AV}} - \ell^e_{\text{DV}}$. For notational simplicity, for each $e \in \{c, d\}$, let $\tau(e) = \tau_{(C,V)}(e) + \ell^e_{\text{AV}} - \ell^e_{\text{DV}}$. Let

$$s = \min\{\tau(c), \tau(d)\}.$$

To ensure that $c$ and $d$ participate 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 $c$ is not beaten by $d$ in their pairwise comparison. Since there are $n' = |V| + \ell_{\text{AV}} - \ell_{\text{DV}}$ voters in the final election $(C, V')$, $d$ must win at most $\lfloor n'/2 \rfloor$ duels against $c$. Let $A = C \setminus \{c, d\}$ and $\tau_{(C,V)}(A) = \sum_{a \in A} \tau_{(C,V)}(a)$. Moreover, for $X \in \{\text{AV, DV}\}$, let $\ell^A_X = \ell_X - \ell^c_X - \ell^d_X$. As $d$ in turn wins $\tau(d)$ comparisons against $c$ in all votes who approve $d$, if $\lfloor n'/2 \rfloor - \tau(d) < 0$, we reject the currently tested candidate $d$ and regard the next one. Otherwise, we search for exactly

$$\underbrace{|V| - \tau_{(C,V)}(c) - \tau_{(C,V)}(d)}_{=\tau_{(C,V)}(A)} - \ell^A_{\mathrm{DV}}$$

voters in $V$ not deleted and approving candidates in $A$, and exactly $\ell^A_{\mathrm{AV}}$ voters added from $W$ and approving some $a \in A$ such that the final election contains at most $\lfloor n'/2 \rfloor - \tau(d)$ voters who approve 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. In more detail, 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

$$b_\alpha(x, V^A) = b_\beta(x, V^A) = \tau_{(C,V)}(A) - \ell^A_{\mathrm{DV}}.$$

There is another arc from $x$ to $W^A$ with lower-bound and upper-bound capacities

$$b_\alpha(x, W^A) = b_\beta(x, W^A) = \ell^A_{\mathrm{AV}}.$$

Each voter $v \in V$ who approves some candidate in $A$ yields an arc $(V^A, v)$ with upper-bound capacity 1 and lower-bound capacity 0. 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 $W$ who approve some $a \in A$. There is an arc from some $v \in V \cup W$ to some $a \in A$ with upper-bound capacity 1 and lower-bound capacity 0 if and only if $v$ approves $a$. Each $a \in A$ yields an arc $(a, y)$ with upper-bound capacity $s$ and lower-bound capacity 0.

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

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

and (minimum) cost of at most $\lfloor n'/2 \rfloor - \tau(d)$ if and only if we can find exactly $\tau_{(C,V)}(A) - \ell^A_{\mathrm{DV}}$ (remaining) voters in $V$ approving some $a \in A$ and exactly $\ell^A_{\mathrm{AV}}$ 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.

$\tau =$ **VRun**. Notice that in this case, $\tau_{(C,V)}(a)$ denotes the number of voters vetoing $a$ in the election $(C, V)$. The algorithm is similar to the above described algorithm with the following differences. First, for $X \in \{\mathrm{AV}, \mathrm{DV}\}$, $\ell^c_{\mathrm{X}}$ and $\ell^d_{\mathrm{X}}$ are defined analogously but with respect to vetoes of $c$ and $d$, respectively. Technically, this is achieved by replacing the occurrences of the word "approve" (respectively, "approves" and "approving" and "approvals") with the word "veto" (respectively, "vetoes" and "vetoing" and "vetoes") throughout the above algorithm. Second, we replace $\lfloor n'/2 \rfloor - \tau(d)$ marked above with $\lfloor n'/2 \rfloor - \tau(c)$. Recall that in the above algorithm, we use the condition $\lfloor n'/2 \rfloor - \tau(d) < 0$ to reject a tested candidate $d$, as in this case a majority of voters in the final election prefer $d$ to $c$. When the rule used is veto with runoff, a majority of voters in the final election prefer $d$ to $c$ if $\lfloor n'/2 \rfloor - \tau(c) < 0$. Finally, in the IMCF instance constructed in the above algorithm, we change the capacity of each arc from $a \in A$ to $y$ so that the lower- bound capacity is $s'$, where $s' = \max\{\tau(c), \tau(d)\}$, and the upper-bound capacity is $|V \cup W|$. The reason is that in veto with runoff, the two candidates with the least vetoes survive the first stage of the election. Therefore, if the final vetoes of $c$ and $d$ are both at most $s'$ with one of them being exactly $s'$, and $c$ and $d$ are the two candidates surviving the first stage, it must be the case that each other candidate has at least $s'$ vetoes 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.

**Theorem 10** *PRun-EDCAV+DV and VRun-EDCAV+DV are in* P.

***Proof*** To solve a PRun-EDCAV+DV or VRun-EDCAV+DV instance $I$ with the distinguished candidate $p$, we solve $m-1$ instances of the constructive exact multimode problems PRun-ECCAV+DV or VRun-ECCAV+DV, respectively, each of which takes the same input as $I$ with only the difference that the distinguished candidate is someone in $C \setminus \{p\}$, where $C$ is the set of candidates in the input and $m = |C|$. Moreover, all the $m-1$ instances have different distinguished candidates. Clearly, $I$ is a YES-instance of either of the two destructive problems if and only if at least one of these $m-1$ instances of the corresponding constructive problem is a YES-instance. Due to Theorem 9, each these $m-1$ instances can be solved in polynomial time. Therefore, $I$ can be solved in polynomial time. □

Note that for each $\text{Y} \in \{\text{CCAV}, \text{CCDV}, \text{CCRV}, \text{DCAV}, \text{DCDV}, \text{DCRV}\}$ and for each $\text{X} \in \{\text{PRun}, \text{VRun}\}$, X-Y is polynomial-time Turing-reducible to its exact version. Then, given the above results, we obtain the following corollary.

**Corollary 4** *For each* $\text{Y} \in \{\text{CCAV}, \text{CCDV}, \text{CCRV}, \text{DCAV}, \text{DCDV}, \text{DCRV}\}$, *both PRun-Y and VRun-Y are in* P.

Concerning control by adding candidates, we have the following results for plurality with runoff and veto with runoff.

**Theorem 11** *PRun-CCAC, PRun-DCAC, VRun-CCAC, and VRun-DCAC are* NP*-complete*.

***Proof*** We prove the theorem by reductions from the RX3C problem. Let $(U, \mathscr{S})$, where $|U| = |\mathscr{S}| = 3\kappa$, be an instance of the RX3C problem. We prove the theorem for the four different problems separately.

**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 distinguished candidate. Moreover, for each $S \in \mathscr{S}$, we create an unregistered candidate, denoted by the same symbol. Regarding the votes, we create $16 + 24\kappa$ votes in total defined as follows.

- First, we create nine votes with $q$ in the first position.
- Second, we create seven votes with $c$ in the first position.
- Third, for each $u \in U$, we create two votes with $u$ in the first position.

The preferences over candidates other than the top-ranked one in the above $16 + 6\kappa$ votes can be set arbitrarily.

- Finally, for each $S \in \mathscr{S}$ and each $u \in S$, we create two votes of the form $S\ u\ c\ q\ \cdots$.

We complete the construction by setting $\ell = \kappa$, i.e., we are allowed to add at most $\kappa$ candidates. It remains to prove the correctness of the reduction: There is an exact 3-set cover if and only if $c$ can be made a winner by adding up to $\kappa$ candidates.

($\Rightarrow$) If there is an exact 3-set cover $\mathscr{S}' \in \mathscr{S}$, we claim that $\mathscr{S}'$ is a solution of the PRun-CCAC instance constructed above. Clearly, after adding candidates in $\mathscr{S}'$, $q$ has 9 approvals, $c$ has 7 approvals, every $S \in \mathscr{S}'$ has 6 approvals, and every $u \in U$ has $8 - 2 = 6$ approvals. Then, according to the definition of plurality with runoff, $q$ and $c$ enter the runoff stage. Clearly, a majority of voters prefer $c$ to $q$, and hence $c$ becomes the unique winner after adding all candidates in $\mathscr{S}'$.

($\Leftarrow$) Consider now the opposite direction. Observe that to ensure $c$ to survive the first stage, at least $\kappa$ candidates must be added, since otherwise there were at least one candidate $u \in U$ which receives at least 8 approvals, resulting in $q$ and $u$ entering the runoff stage. Let $\mathscr{S}'$ be a solution of the PRun-CCAC instance. As discussed, we have $|\mathscr{S}'| = \kappa$. If $\mathscr{S}'$ is not an exact 3-set cover, again there is a candidate $u \in U$ such that $u$ is not in any subset of $\mathscr{S}'$. According to the construction of the instance, the candidate $u$ receives at least 8 approvals after adding the candidates in $\mathscr{S}'$, and hence survives the first stage with $q$. Therefore, $\mathscr{S}'$ must be an exact 3-set cover of $U$.

**PRun-DCAC.** The reduction differs from the above proof for PRun-CCAC only in that the distinguished candidate is $q$. The correctness relies on the observation that candidate $c$ is the only candidate that can preclude $q$ from winning.

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

- We create one vote of the form $\mathscr{S}\ U\ c\ q$.
- For each $u \in U$, we crate $6\kappa - 3$ votes of the form $c\ q\ \mathscr{S}\ U \setminus \{u\}\ u$.
- For each $S \in \mathscr{S}$, we create $6\kappa + 5$ votes as follows:
  - $3\kappa + 1$ votes of the form $q\ U\ c\ \mathscr{S} \setminus \{S\}\ S$;
  - $3\kappa + 1$ votes of the form $c\ U\ q\ \mathscr{S} \setminus \{S\}\ S$; and
  - three votes of the form $q\ U\ \mathscr{S} \setminus \{S\}\ c\ S$.
- For each $S = \{u_x, u_y, u_z\} \in \mathscr{S}$, we further create six votes as follows:
  - two votes of the form $c\ q\ U \setminus \{u_x\}\ \mathscr{S} \setminus \{S\}\ u_x\ S$;
  - two votes of the form $c\ q\ U \setminus \{u_y\}\ \mathscr{S} \setminus \{S\}\ u_y\ S$; and
  - two votes of the form $c\ q\ U \setminus \{u_z\}\ \mathscr{S} \setminus \{S\}\ u_z\ S$.

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.

(⇒) 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 one 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 on to the runoff stage. As more voters prefer $c$ over $q$, $c$ becomes the final winner.

(⇐) Suppose that we can add a subset $\mathcal{S}' \subseteq \mathcal{S}$ of at most $\kappa$ unregistered candidates to make $c$ 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 $u$ being the two candidates competing in the runoff stage. We can conclude that $\mathcal{S}'$ is an exact 3-set cover.

**VRun-DCAC.** The reduction differs from the one for VRun-CCAC only in that the distinguished candidate is $q$. The correctness relies on the observation that candidate $c$ is the only candidate that can preclude $q$ from winning.

Next, we study the complexity of control by deleting candidates for plurality with runoff and veto with runoff.

**Theorem 12** *PRun-CCDC, PRun-DCDC, VRun-CCDC, and VRun-DCDC are* NP*-complete.*

***Proof*** Again, letting $(U, \mathcal{S})$ with $|U| = |\mathcal{S}| = 3\kappa$ be a given RX3C instance, we separately provide our four reductions from RX3C to PRun-CCDC, PRun-DCDC, VRun-CCDC, and VRun-DCDC, respectively. Let $U = \{u_1, u_2, \ldots, u_{3\kappa}\}$. Without loss of generality, assume that $\kappa \geq 4$.

**PRun-CCDC.** From $(U, \mathcal{S})$, we create the following instance of PRun-CCDC. Let $C = \{c, q\} \cup U \cup \mathcal{S}$ be the set of candidates and $c$ the distinguished candidate. We create a multiset $V$ of $9\kappa^2 + 21\kappa + 1$ votes as follows.

- We create $2\kappa$ votes of the form $q\ u_1\ u_2\ \ldots\ u_{3\kappa}\ \mathcal{S}\ c$.
- We create $\kappa + 1$ votes of the form $q\ u_{3\kappa}\ u_{3\kappa-1}\ \ldots\ u_1\ \mathcal{S}\ c$.
- For each $u \in U$, we create $3\kappa - 3$ votes of the form $u\ U \setminus \{u\}\ \mathcal{S}\ c\ q$.
- For each $S \in \mathcal{S}$, we create three votes of the form $S\ c\ C \setminus (S \cup \{c, q\})\ q$.
- For each $S = \{u_x, u_y, u_z\} \in \mathcal{S}$, we further create six votes as follows:
  - two votes of the form $S\ u_x\ C \setminus \{c, q, u_x\}\ c\ q$;
  - two votes of the form $S\ u_y\ C \setminus \{c, q, u_y\}\ c\ q$; and
  - two votes of the form $S\ u_z\ C \setminus \{c, q, u_z\}\ c\ q$.

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

(⇒) Assume there is an exact 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 $c$ being the final winner.

(⇐) 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

**Table 8** Plurality scores of candidates in the reduction for PRun-DCDC in the proof of Theorem 12. The numbers in the equation in each row corresponding to a candidate are the plurality scores of the candidates received respectively from the four groups of votes constructed above

| | plurality scores |
|---|---|
| $q$ | $(3\kappa + 4) + 0 + 0 + 0 = 3\kappa + 4$ |
| $c$ | $0 + 0 + 0 + 0 = 0$ |
| $u \in U$ | $0 + (3\kappa - 3) + 0 + 0 = 3\kappa - 3$ |
| $S \in \mathscr{S}$ | $0 + 0 + 9 + 0 = 9$ |
| $h_i$ | $0 + 0 + 0 + 1 = 1$ |
| $a_j$ | $0 + 0 + 0 + 0 = 0$ |

$3\kappa - 3 + \kappa + 1 = 4\kappa - 2$ 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 $u$ (in the third voter group) 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 $C' \subseteq \mathscr{S}$. 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, there must be a candidate $u \in U$ who occurs in at least two subsets of $C'$. Due to the construction, 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.

**PRun-DCDC.** The candidate set is

$$C = \{c, q\} \cup U \cup \mathscr{S} \cup \{h_1, \dots, h_{9\kappa^2+15\kappa}\} \cup A,$$

where $A = \{a_1, \dots, a_\kappa\}$. For two positive integers $x$ and $y$ such that $x < y \leq 9\kappa^2$, we define

$$H[x, y] = \{h_z \mid x \leq z \leq y\}.$$

We create in total $18\kappa^2 + 36\kappa + 4$ votes classified into the following groups.

1. There are $3\kappa + 4$ votes of the form $q\ C \setminus \{q\}$.
2. For each $i \in [3\kappa]$, there are $3\kappa - 3$ votes of the form

$$u_i\ H[(i-1) \cdot \kappa, i \cdot \kappa]\ C \setminus (A \cup H[(i-1) \cdot \kappa, i \cdot \kappa] \cup \{u_i, c, q\})\ c\ q\ A.$$

3. For each $S \in \mathscr{S}$, $S = \{u_x, u_y, u_z\}$, where $\{x, y, z\} \subseteq [3\kappa]$, there are nine votes as follows:
   - three votes of the form $S\ c\ q\ C \setminus \{S, c, q\}$;
   - two votes of the form $S\ u_x\ c\ q\ C \setminus \{S, u_x, c, q\}$;
   - two votes of the form $S\ u_y\ c\ q\ C \setminus \{S, u_y, c, q\}$; and
   - two votes of the form $S\ u_z\ c\ q\ C \setminus \{S, u_z, c, q\}$.

4. There are $9\kappa^2 + 15\kappa$ votes denoted by $v_1, \dots, v_{9\kappa^2+15\kappa}$ such that for every $i \in [9\kappa^2 + 15\kappa]$), the vote $v_i$ is of the form

$$h_i\ A\ c\ q\ C \setminus (\{c, q, h_i\} \cup A).$$

Let $V$ denote the multiset of the above constructed votes. The distinguished candidate is $q$. Finally, we define $\ell = \kappa$, i.e., we are allowed to delete at most $\kappa$ candidates from $C$. The time to construct the above instance is clearly bounded by a polynomial in the size of the RX3C instance.

We are left with the proof of correctness of the reduction. It is useful to first provide a summary of the plurality scores of all candidates for a better understanding of the following arguments. We refer to Table 8 for such a summary.

Due to Table 8, $q$ survives the first stage but $c$ does not. One can check that $q$ is beaten by $c$ but beats everyone else. As a consequence, $q$ is the winner of the above constructed election.

($\Rightarrow$) Assume that there is an exact set cover $\mathcal{S}' \subseteq \mathcal{S}$ of $U$. Let $E = (C \setminus \mathcal{S}', V)$. We claim that $q$ is no longer the winner of the election $E$. With the help of Table 8 one can check easily that in the election $E$ the two candidates $q$ and $c$ receive the most approvals. Particularly, if a candidate $S \in \mathcal{S}'$ is deleted, the three votes of the form $S\ c\ q\ C \setminus \{S, c, q\}$ give three approvals to $c$. Then, as $|\mathcal{S}'| = \kappa$, after deleting the candidates in $\mathcal{S}'$, the candidate $c$ receives $3\kappa$ new approvals. In addition, as $\mathcal{S}'$ is an exact set cover, for every $u \in U$, there is exactly one $S \in \mathcal{S}'$ such that $u \in S$. Then, due to the construction of the votes in the third group, the plurality score of $u$ increases by exactly two, reaching to $3\kappa - 3 + 2 = 3\kappa - 1$. Other candidates clearly have only constant plurality scores. Therefore, $c$ and $q$ are the two candidates that survive the first stage, and this is the case no matter which tie-breaking scheme is used. As $c$ beats $q$ in the election $E$, we know that $q$ is no longer a winner.

($\Leftarrow$) Assume that there is a subset $C' \subseteq C \setminus \{q\}$ of at most $\kappa$ candidates such that $q$ is no longer a winner of $(C \setminus C', V)$. First, it is easy to verify that it is impossible to prevent $q$ from surviving the first stage by deleting at most $\kappa$ candidates. Additionally, candidate $c$ is the only one beating $q$. Due to these two observations, we know that the candidates surviving the first stage of $(C \setminus C', V)$ must be $c$ and $q$. By Table 8, there are candidates in $U$ who receive at least $3\kappa - 3$ approvals in $E$. This means that the deletion of the candidates in $C'$ increases the plurality score of $c$ to at least $3\kappa - 3$. Note that after deleting candidates in $C'$, none of the votes in the groups (1), (2), and (4) rank $c$ in the top. Therefore, the plurality score of $c$ must be from votes in the group (3). Another significant observation is that $C' \subseteq \mathcal{S}$ and, moreover, $|C'| = \kappa$, since otherwise at least one candidate in $U$ has a higher plurality score than that of $c$ in $E$. Therefore, we know that in the election $E$, $c$ has plurality score exactly $3\kappa$. Finally, we claim that $C'$ is an exact set cover of $U$. Assume for the sake of contradiction that this is not the case. Then there exists at least one candidate $u \in U$ such that there are two $S, S' \in C'$ such that $u \in S \cap S'$. By the construction of the votes in the group (3), the candidate $u$ will be ranked in the top in four votes (two of the form $S\ u\ c\ q\ C \setminus \{S, u, c, q\}$ and two of the form $S'\ u\ c\ q\ C \setminus \{S', u, c, q\}$). This means that in the election $E$, the plurality score of $u$ is at least $3\kappa - 3 + 4 = 3\kappa + 1$, which is larger than that of $c$. However, in this case, $c$ is excluded in the first stage, a contradiction.

**VRun-DCDC.** The candidate set is the same as in the reduction for PRun-CCDC. Precisely, we define

$$C = \{c, q\} \cup U \cup \mathcal{S},$$

**Table 9** Vetoes of candidates in the instance of VRun-DCDC in the proof of Theorem 12

| | vetoes |
|---|---|
| $q$ | 0 |
| $c$ | 3 |
| $u \in U$ | 2 |
| $S \in \mathscr{S}$ | 6 |

where $q$ is the distinguished candidate. We create the following votes.

- There are three votes of the form $q\ \mathscr{S}\ U\ c$.
- For each $S = \{u_x, u_y, u_z\} \in \mathscr{S}$, we create six votes as follows:
  - two votes of the form $c\ q\ U \setminus \{u_x\}\ \mathscr{S} \setminus \{S\}\ u_x\ S$;
  - two votes of the form $c\ q\ U \setminus \{u_y\}\ \mathscr{S} \setminus \{S\}\ u_y\ S$; and
  - two votes of the form $c\ q\ U \setminus \{u_z\}\ \mathscr{S} \setminus \{S\}\ u_z\ S$.
- For each $u \in U$, there are two votes of the form $c\ q\ \mathscr{S}\ U \setminus \{u\}\ u$.

Finally, we define $\ell = \kappa$, i.e., we are allowed to delete at most $\kappa$ candidates from $C \setminus \{q\}$. Clearly, the above instance of VRun-DCDC can be constructed in polynomial time. We show that there is an exact set cover of $U$ if and only if the above VRun-DCDC instance is a YES-instance. The number of vetoes of all candidates are summarized in Table 9.

From Table 9, we know that $q$ and some $u \in U$ survives the first stage of the election. In addition, it is easy to verify that $q$ beats everyone else except $c$, and hence $q$ wins the election.

($\Rightarrow$) Assume that $U$ admits an exact set cover $\mathscr{S}' \subseteq \mathscr{S}$. Let $E' = (C \setminus \mathscr{S}', V)$. We claim that $q$ is no longer a winner in the election $E'$. To this end, let us first analyze the vetoes of candidates in $E'$. Observe that deleting candidates only in $\mathscr{S}$ never changes the vetoes of $c$ and $q$. So, the vetoes of $q$ and $c$ in $E'$ are still 0 and 3, respectively. For each $u \in U$, as $\mathscr{S}'$ is an exact set cover of $U$, there is exactly one $S \in \mathscr{S}'$ such that $u \in S$. Then, after deleting $S$ from $C$, $u$ receives two more vetoes from the two votes of the form $c\ q\ U \setminus \{u\}\ \mathscr{S} \setminus \{S\}\ u\ S$, resulting in a final veto count of $2 + 2 = 4$. As this holds for all candidates in $U$, the two candidates surviving the first stage of the election are $q$ and $c$. As pointed out above, $c$ beats $q$, and hence $c$ substitutes $q$ as the new winner in $E'$.

($\Leftarrow$) Assume that there is a subset $C' \subseteq C \setminus \{q\}$ of at most $\kappa$ candidates such that $q$ is no longer a winner of $(C \setminus C', V)$ under veto with runoff. Let $E' = (C \setminus C', V)$. From Table 9, it holds that every candidate in $C \setminus C'$ except $q$ has a positive veto count in $E'$. Moreover, as in each of the above constructed votes there are more than $\kappa + 1$ candidates ranked after $q$ and $|C'| \leq \kappa$, in the election $E'$, $q$ has no vetoes. This means that $q$ survives the first stage of $E'$. Then, as $c$ is the only candidate that beats $q$, we know that $c$ is the other candidate who survives the first stage together with $q$. This implies that $c \notin C'$. As in each vote not vetoing $c$, there are more than $\kappa + 1$ candidates ranked after $c$, and it holds that $|C'| \leq \kappa$, we know that the veto count of $c$ in $E'$ is 3. Let $\mathscr{S}' = C' \cap \mathscr{S}$ and $U' = U \setminus \bigcup_{S \in \mathscr{S}'} S$. We first prove the following claims.

**Claim 1** $U' \subseteq C'$.

Assume for the sake of contradiction there exists a candidate $u \in U'$ such that $u \notin C'$. Then, due to the definition of the votes, $u$ has two vetoes in $E'$. However, this contradicts with the fact that $c$ is the candidate that survives the first stage with $q$. This proves Claim 1.

**Claim 2** $U' = \emptyset$.

Let $t = |C' \cap \mathcal{S}|$ and $t' = |C' \cap U|$. If $U' \neq \emptyset$, then we have $t < \kappa$. As $\mathcal{S}$ covers at most $3t$ elements of $U$, it holds that $t' \geq 3\kappa - 3t$. It follows that $t + t' \geq 3\kappa - 2t > \kappa$, a contradiction. This proves Claim 2.

Due to the above claim, we know that $\mathcal{S}$ covers $U$. Then, as $|\mathcal{S}| \leq \kappa$, $\mathcal{S}$ must be an exact set cover of $U$.

**VRun-CCDC.** The reduction for VRun-CCDC is similar to the above reduction for VRun-DCDC with only the difference that we set $c$ as the distinguished candidate. If $U$ admits an exact set cover, then as shown above, after deleting the candidates corresponding to this set cover, $c$ becomes the winner. For the other direction, one observes first that the above two claims still hold in this case. Then it is easy to see that if $c$ becomes a winner after deleting at most $\kappa$ candidates, the deleted candidates must correspond to an exact set cover of $U$.

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 result for PRun-CCAC studied in Theorem 11 and Lemma 2 directly yields NP-hardness of PRun-CCRC . In addition, plurality with runoff satisfies IBC when ties are broken deterministically. Hence, from Lemma 1 and the NP-hardness of PRun-DCDC stated in Theorem 12, it follows that PRun-DCRC is NP-hard when ties are broken deterministically. However, in the proof of NP-hardness of PRun-DCDC, the distinguished candidate $q$ has a strictly higher plurality score than any other candidate. So, no matter which tie-breaking scheme is used, $q$ survives the first stage. In addition, as $c$ is the candidate who replaces $q$ as the winner in the final election, it does not matter which candidate in $U$ survives the first stage with $q$ in the original election. Therefore, NP-hardness applies to all tie-breaking schemes. (Precisely, we modify the instance of PRun-DCDC by adding an additional set of $\kappa$ unregistered candidates who are ranked after all the other candidates in all votes.)

However, it is easy to check that veto with runoff is not unanimous and does not satisfy IBC either. Hence, we cannot obtain NP-hardness for VRun-CCRC and VRun-DCRC using Lemmas 1 and 2. Nevertheless, we can show NP-hardness of these problems by modifications of the proofs for VRun-CCAC and VRun-DCDC studied in Theorems 11 and 12. In particular, to obtain NP-hardness of VRun-CCRC, we modify the instance of VRun-CCAC by adding an additional set of $\kappa$ registered candidates and rank them before all the other candidates in all votes. More importantly, we rank all the $\kappa$ registered candidates in an arbitrary but fixed order so that they have to be replaced to guarantee the victory of the distinguished candidate. To obtain NP-hardness of VRun-DCRC, we modify the instance of VRun-DCDC by creating a set of $\kappa$ unregistered candidates, and rank them directly after $q$ in all votes (i.e., $q$ and these $\kappa$ candidates are ranked consecutively in all votes with $q$ being the first one among them). The relative order among these $\kappa$ candidates does not matter.

Summing up, we have the following results.

**Theorem 13** *PRun-CCRC, PRun-DCRC, VRun-CCRC, and VRun-DCRC are* NP-*complete.*

**Table 10** Complexity of control for Condorcet. Our results are in boldface. "NPC" stands for "NP-complete," "P" for "polynomial-time solvable," and "'I" for "immune"

| CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *NPC* | *NPC* | **NPC** | *I* | *P* | **P** | *P* | *P* | **P** | *P* | *I* | **P** |

Note that the NP-hardness results in the above three theorems (Theorems 11, 12, and 13) hold regardless of the tie-breaking rule used because no tie occurs in either stage of the constructed elections.

## 8 Condorcet voting

In this section, we study Condorcet voting. Our results of this section are summarized in Table 10.

For Condorcet we will show that it is vulnerable to three types of replacement control, yet resistant to the fourth one, starting with the resistance proof.

**Theorem 14** *Condorcet-CCRV is* NP-*complete.*

***Proof*** We prove NP-hardness by reducing RX3C to Condorcet-CCRV.[10] Let $(U, \mathscr{S})$ be an RX3C instance with $U = \{u_1, \ldots, u_{3\kappa}\}$, $\kappa \geq 2$ (which may be assumed, as RX3C is trivially solvable when $\kappa = 1$), and $\mathscr{S} = \{S_1, \ldots, S_{3\kappa}\}$. The set of candidates is $C = U \cup \{c\}$ with $c$ being the distinguished candidate. The votes are constructed as follows:

- There are $2\kappa - 3$ registered votes of the form $u_1 \ \cdots \ u_{3\kappa} \ c$ in $V$ and
- for each $j$, $1 \leq j \leq 3\kappa$, there is one unregistered vote of the form $S_j \ c \ U \setminus S_j$ in $W$.

The ordering of candidates in $S_j$ and $U \setminus S_j$ does not matter in any of those votes. Finally, set $\ell = \kappa$.

Analyzing the election $(C, V)$, $u_1$ is the Condorcet winner; in particular, $c$ loses against every $u_i \in U$ with a deficit of $2\kappa - 3$ votes, i.e.,

$$N_{(C,V)}(u_i, c) - N_{(C,V)}(c, u_i) = 2\kappa - 3.$$

We will now show that $(U, \mathscr{S})$ is a YES-instance of RX3C if and only if $c$ can be made the Condorcet winner of the election by replacing $\kappa$ votes from $V$ with votes from $W$.

(⇒) Assume there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of $U$. We remove $\kappa$ votes of the form $u_1 \ \cdots \ u_{3\kappa} \ c$ from the election and replace them by the votes of the form $S_j \ c \ U \setminus S_j$ for all $S_j \in \mathscr{S}'$. Let $(C, V')$ be the resulting election. Since $\mathscr{S}'$ is an exact cover of $U$, for each $u_i \in U$,

$$N_{(C,V')}(u_i, c) - N_{(C,V')}(c, u_i) = (2\kappa - 3 - \kappa + 1) - (\kappa - 1) = -1 < 0.$$

[10] A similar reduction was used by Bartholdi, Tovey, and Trick [7] to prove that Condorcet-CCAV is NP-hard.

Thus $c$ now defeats each $u_i \in U$ in pairwise comparison and, therefore, has been made the Condorcet winner of $(C, V')$.

($\Leftarrow$) Assume that $c$ can be made a Condorcet winner of the election by replacing at most $\kappa$ votes. Recall that $c$ has a deficit of

$$N_{(C,V)}(u_i, c) - N_{(C,V)}(c, u_i) = 2\kappa - 3$$

to every $u_i \in U$ in the original election. Thus *exactly* $\kappa$ votes need to be removed from the election, for otherwise $c$'s deficit of at least $\kappa - 2$ to every other candidate cannot be caught up on, since at least one other candidate is in front of $c$ in every unregistered vote. With $\kappa$ removed votes, $c$'s deficit to every other candidate is now decreased to $\kappa - 3$. However, none of the $\kappa$ votes from $W$ replacing the removed votes can rank some $u_i \in U$ in front of $c$ more than once, as otherwise we would have

$$N_{(C,V')}(u_i, c) \geq \kappa - 1 > \kappa - 2 \geq N_{(C,V')}(c, u_i)$$

for at least one $u_i \in U$ in the resulting election $(C, V')$, and $c$ would not win. Let $\mathscr{S}' \subseteq \mathscr{S}$ be the set such that each $S_j \in \mathscr{S}'$ corresponds to the vote $S_j\ c\ U \setminus S_j$ from $W$ that is added to the election to replace a removed vote. Every unregistered voter ranks three candidates of $U$ in front of $c$. By the pigeonhole principle, in order for the $\kappa$ new votes to rank each of the $3\kappa$ candidates of $U$ in front of $c$ only once, $\mathscr{S}'$ needs to be an exact cover of $U$.

By contrast, we show vulnerability to destructive control by replacing voters for Condorcet via a simple algorithm.

**Theorem 15** *Condorcet-DCRV is in* P.

***Proof*** To prove membership in P, we will provide an algorithm that solves the problem in polynomial time and outputs, if possible, which of the registered voters must be replaced by which unregistered voters for $c$ to not win.

The input to our algorithm is an election $(C, V \cup W)$, the distinguished candidate $c \in C$, and a positive integer $\ell$. The algorithm will output either a pair $(V', W')$ with $V' \subseteq V$, $W' \subseteq W$, and $|V'| = |W'| \leq \ell$ (i.e., for $c$ to not win, there are votes in $V'$ that must be removed and votes in $W'$ that must be added to the election instead), or that control is impossible.

First, the algorithm checks whether $c$ is already not winning the election $(C, V)$ and outputs $(\emptyset, \emptyset)$ if this is the case, and we are done.

Otherwise, $c$ currently wins, and the algorithm iterates over all candidates $d \in C \setminus \{c\}$ and first checks whether $N_{(C,V)}(c, d) - N_{(C,V)}(d, c) + 1 \leq 2\ell$ (if this is not the case, $d$ loses to $c$ in any case and we can skip this candidate.)

Let $V' \subseteq V$ contain at most $\ell$ votes from $V$ preferring $c$ to $d$ and let $W' \subseteq W$ contain at most $\ell$ votes from $W$ preferring $d$ to $c$. If one of them is smaller than the other, remove votes from the larger one until they are equal in size.

Then we check whether $N_E(c, d) \leq N_E(d, c)$ in the election $E = (C, (V \cup W') \setminus V'))$. If this is the case, $c$ does not beat $d$ in direct comparison, so $c$ cannot win the election. The algorithm then outputs $(V', W')$.

Otherwise, $d$ cannot beat $c$ and the algorithm proceeds to the next candidate. If, after all iterations, no candidate was found that beats or ties $c$, the algorithm outputs "control impossible." Obviously, this algorithm runs in polynomial-time and solves the problem.

Bartholdi, Tovey, and Trick [7] observed that, due to the Weak Axiom of Revealed Preference, Condorcet voting is immune to constructive control by adding candidates, and Hemaspaandra, Hemaspaandra, and Rothe [33] made the same observation regarding destructive control by deleting candidates. For control by *replacing* candidates, however, Condorcet is susceptible both in the constructive and in the destructive case, as shown in the following example.

***Example 1*** To see that Condorcet is susceptible to constructive control by replacing candidates, consider a set $C = \{b, c\}$ with two registered candidates, a set $D = \{d\}$ with just one unregistered candidate, and only one vote of the form $b\ c\ d$ over $C \cup D$. We can turn $c$ (who does not win according to $b\ c$) into a Condorcet winner by replacing $b$ with $d$ (so we now have $c\ d$).

For susceptibility in the destructive case, just consider $C' = \{c, d\}$ and $D' = \{b\}$, and replace $d$ with $b$, all else being equal.

Moreover, since in Condorcet elections the direct comparison between two candidates cannot be influenced by deleting or adding other candidates to the election, Condorcet-CCRC and Condorcet-DCRC are both easy to solve.

**Theorem 16** *Condorcet-CCRC is in* P.

***Proof*** To prove membership in P, we will provide an algorithm that solves the problem in polynomial time and outputs, if possible, which of the original candidates must be replaced by which unregistered candidates for $c$ to win.

The input to our algorithm is an election $(C \cup D, V)$, the distinguished candidate $c \in C$, and a positive integer $\ell$. The algorithm will output either a pair $(C', D')$ with $C' \subseteq C \setminus \{c\}$, $D' \subseteq D$, and $|C'| = |D'| \leq \ell$ (i.e., for $c$ to win, there are candidates in $C'$ that must be removed and candidates in $D'$ that must be added to the election instead), or that control is impossible.

First, we check whether $c$ already wins the election $(C, V)$ and output $(\emptyset, \emptyset)$ if this is the case, and we are done.

Otherwise, let $C' \subseteq C \setminus \{c\}$ be the set of candidates from $C \setminus \{c\}$ that beat or tie $c$ in direct comparison and let $D' \subseteq D$ be a set of at most $|C'|$ candidates from $D$ that $c$ beats in direct comparison.

If $|C'| \leq \ell$ and $|C'| = |D'|$, we output $(C', D')$, and otherwise we output "control impossible."

Obviously, the algorithm solves the problem and runs in polynomial time.

**Theorem 17** *Condorcet-DCRC is in* P.

***Proof*** An algorithm that solves the problem works as follows: Given an election $(C \cup D, V)$, a distinguished candidate $c \in C$, and an integer $\ell$, it checks whether $c$ is not winning the election $(C, V)$ and outputs $(\emptyset, \emptyset)$ if this is the case.

Otherwise, it checks whether there is a candidate $d \in D$ who beats or ties $c$ in direct comparison, whether there is another candidate $b \in C$ with $b \neq c$ and whether $\ell \geq 1$. If these conditions are satisfied, it outputs $(\{b\}, \{d\})$, and otherwise "control impossible."

**Table 11** Complexity of control for fallback voting. Our results are in boldface. "NPC" stands for "NP-complete" and "P" stands for "polynomial-time solvable"

| CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** | *P* | *P* | **P** | *NPC* | *NPC* | **NPC** |

This algorithm outputs either a successful pair $(C', D')$ with $C' \subseteq C \setminus \{c\}$, $D' \in D$, and $|C'| = |D'| \leq \ell$ if $c$ can be prevented from winning by replacing at most $\ell$ candidates, or else "control impossible." Obviously, the algorithm is correct and runs in polynomial time.

## 9 Fallback voting

We will now consider fallback voting and show that it is vulnerable to one type of replacement control and resistant to the others. Our results for fallback voting are summarized in Table 11.

**Theorem 18** *Fallback-CCRV is* NP-*complete*.

***Proof*** To prove NP-hardness, we will modify the reduction from X3C that Erdélyi and Rothe [22] (and Erdélyi et al. [16]) used to show NP-hardness of Fallback-CCAV. Let $(U, \mathscr{S})$ be an X3C instance with $U = \{u_1, \dots, u_{3\kappa}\}$, $\kappa \geq 2$, and $\mathscr{S} = \{S_1, \dots, S_t\}$, $t \geq 1$. The set of candidates is $C = U \cup B \cup \{c\}$ with $c$ being the distinguished candidate and $B = \{b_1, \dots, b_{t(3\kappa-4)}\}$ a set of $t(3\kappa - 4)$ dummy candidates. In $V$ (corresponding to the registered voters), there are the $3\kappa - 1$ votes (recall the input format in fallback elections described in Sect. 3):

- $2\kappa - 1$ votes of the form $U \mid B \cup \{c\}$ and
- for each $i$, $1 \leq i \leq \kappa$, one vote of the form $b_i \mid U \cup (B \setminus \{b_i\}) \cup \{c\}$.

In $W$ (corresponding to the unregistered voters), there are the following $t$ votes:

– For each $j$, $1 \leq j \leq t$, let $B_j = \{b_{(j-1)(3\kappa-4)+1}, \dots, b_{j(3\kappa-4)}\}$ and include in $W$ the vote $B_j\ S_j\ c \mid (U \setminus S_j) \cup (B \setminus B_j)$.

Finally, set $\ell = \kappa$.

Having no approvals in $(C, V)$, $c$ does not win. We will show that $(U, \mathscr{S})$ is a YES-instance of X3C if and only if $c$ can be made a fallback winner of the constructed election by replacing at most $\kappa$ votes from $V$ with as many votes from $W$.

($\Rightarrow$) Suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of $U$. Remove $\kappa$ votes $U \mid B \cup \{c\}$ from the election and add, for each $S_j \in \mathscr{S}'$, the vote $B_j\ S_j\ c \mid (U \setminus S_j) \cup (B \setminus B_j)$ instead. Let $(C, \widehat{V})$ be the resulting election. It follows that

- $score_{(C,\widehat{V})}(b_i) \leq 2$ for every $b_i \in B$,
- $score_{(C,\widehat{V})}(u_i) = \kappa$ for every $u_i \in U$ ($\kappa - 1$ approvals from the remaining registered voters and one approval from the added voters since $\mathscr{S}'$ is an exact cover of $U$), and

- $score_{(C,\hat{V})}(c) = \kappa$.

Thus no candidate has a majority on any level and $c$ is one of the winners since he or she ties all candidates of $U$ for the most approvals overall.

($\Leftarrow$) Suppose $c$ can be made a fallback winner of the election by replacing at most $\kappa$ votes from $V$ with as many votes from $W$. Since $c$ has no approvals in $(C, V)$ and we can only add at most $\kappa$ approvals for $c$, the only chance for $c$ to win is to have the most approvals in the last stage of the election. Regardless of which votes we remove or add to the election, every dummy candidate can have at most two approvals, which will at least be tied by $c$ if we add $\kappa \geq 2$ unregistered votes to the election. We need to remove $\kappa$ votes $U \mid B \cup \{c\}$ from the election; otherwise, some $u_i \in U$ would have at least $s$ approvals, whereas $c$ could gain no more than $\kappa - 1$ approvals from adding unregistered votes. Each $u_i \in U$ receives $\kappa - 1$ approvals from the remaining registered votes of the original election and $c$ receives $\kappa$ approvals from the added votes. Additionally, every added voter approves of three candidates from $U$. Hence, in order for $c$ to at least tie every candidate from $U$, each $u_i \in U$ can only be approved by at most one of the added votes. Since there are $\kappa$ added votes, there must be an exact cover of $U$.

By contrast, we establish vulnerability of the destructive case of control by replacing voters for fallback voting. The proof employs a rather involved polynomial-time algorithm solving this problem.

**Theorem 19** *Fallback-DCRV is in* P.

***Proof*** We provide a polynomial-time algorithm that solves the problem and computes which voters need to be removed and which need to be added to prevent the distinguished candidate from being a fallback winner. The algorithm is inspired by an algorithm designed by Erdélyi and Rothe [22] (see also Erdélyi et al. [16]) to prove membership of fallback-DCAV in P.

For an election $(C, V)$, let $maj(V) = \lfloor |V|/2 \rfloor + 1$ and let

$$def^i_{(C,V)}(d) = maj(V) - score^i_{(C,V)}(d)$$

be the deficit of candidate $d \in C$ to a strict majority in $(C, V)$ on level $i$, $1 \leq i \leq |C|$. Note that the number of voters is always the same, namely $|V|$, and so we will use $maj(V)$ even after we have replaced some voters.

The input of the algorithm is an election $(C, V \cup W)$, a distinguished candidate $c \in C$, and an integer $\ell$. The algorithm will output either a pair $(V', W')$ with $V' \subseteq V$, $W' \subseteq W$, and $|V'| = |W'| \leq \ell$ (i.e., for $c$ to not win, there are votes in $V'$ that must be removed and votes in $W'$ that must be added to the election instead), or that control is impossible.

The algorithm runs through $n = \max_{v \in V \cup W} |S_v|$ stages which we call the *majority stages* and one final stage which we call the *approval stage*. In the majority stages the algorithm checks whether $c$ can be beaten in the first $n$ levels of the fallback election by replacing at most $\ell$ voters, and in the approval stage it checks whether $c$ can be dethroned in the last stage of the fallback election by this control action.

The algorithm works as follows: If $c$ is already not winning in $(C, V)$, we output $(\emptyset, \emptyset)$ and are done.

Majority Stage $i$, $1 \leq i \leq n$: For $i > 1$, this stage is reached if we could not successfully control the election in majority stages 1 through $i - 1$. Note that in each majority stage $i$

we assume that a candidate that is approved by a voter on level $j > i$ is disapproved by this voter. Now, for every candidate $d \in C \setminus \{c\}$, we check whether $d$ can beat $c$ on level $i$ of the fallback election. First, we check if the following two conditions hold:

$$def^{i}_{(C,V)}(d) \leq \ell; \tag{1}$$

$$score^{i}_{(C,V)}(d) > score^{i}_{(C,V)}(c) - 2\ell. \tag{2}$$

If at least one of (1) and (2) does not hold, $d$ cannot have a strict majority on level $i$ or cannot beat $c$ on this level, no matter which at most $\ell$ votes we replace, and we skip $d$ and proceed to the next candidate (or the next stage if all candidates failed to beat $c$ in this stage).

Otherwise (i.e., if both (1) and (2) hold), we determine the largest $W_d \subseteq W$ such that $|W_d| \leq \ell$ and all votes of $W_d$ approve of $d$ and disapprove of $c$ on the first $i$ levels. Furthermore, we determine the largest $V_d \subseteq V$ such that $|V_d| \leq \ell$ and all votes of $V_d$ approve of $c$ and disapprove of $d$ on the first $i$ levels. Again, if $|V_d| \neq |W_d|$, we fill up the smaller vote list with votes as follows until they are equal in size:

- If $|V_d| < |W_d|$, we fill up $V_d$ with votes of $V \setminus V_d$ who approve of neither $c$ nor $d$ until we either have $|V_d| = |W_d|$ or run out of those votes, and in the latter case we now keep adding to $V_d$ those votes of $V \setminus V_d$ who approve of both $c$ and $d$ while prioritizing those votes that approve of $c$ on levels up to $i - 1$ over votes that approve of $c$ on level $i$. Only if this is still not enough to make these two vote lists equal in size, we remove votes from $W_d$ until both lists are equally large.
- If $|V_d| > |W_d|$, we fill up $W_d$ with votes of $W \setminus W_d$ that approve of both $c$ and $d$ on the first $i$ levels while prioritizing those votes that approve of $c$ on level $i$ over votes that approve of $c$ on levels up to $i - 1$, and if this is not enough to make these two vote lists equal in size, we add those votes from $W \setminus W_d$ to $W_d$ that disapprove of both $c$ and $d$. Again, only if this is still not enough to make them both equal in size, we will remove votes from $V_d$ (while prioritizing votes that approve of $c$ on level $i$) until both lists are equally large.

Now, knowing that the resulting lists $V_d$ and $W_d$ are equal in size, we check the following condition:

$$score^{i}_{(C,(V\setminus V_d)\cup W_d)}(d) \geq maj(V); \tag{3}$$

$$score^{i}_{(C,(V\setminus V_d)\cup W_d)}(d) > score^{i}_{(C,(V\setminus V_d)\cup W_d)}(c). \tag{4}$$

If (3) or (4) does not hold, $d$ cannot beat $c$ and win on level $i$, and we skip $d$ and proceed to the next candidate or the next stage.

Otherwise, we check the following condition:

$$score^{i-1}_{(C,(V\setminus V_d)\cup W_d)}(c) \geq maj(V). \tag{5}$$

If (5) does not hold, we output $(V_d, W_d)$, as $d$ wins on the $i$th level and so prevents $c$ from winning. Note that for $i = 1$ condition (5) always fails to hold, so the following steps are only done in majority stages 2 through $n$. If (5) does hold, then $c$ wins on an earlier level and we failed to control the election. We will try to fix this, if at all possible, in two steps.

Firstly, if there are votes in $W_d$ that approve of $c$ on levels up to $i-1$ and of $d$ on the first $i$ levels (this would mean that all votes in $V_d$ approve of $c$ and disapprove of $d$ on the first $i$ levels), then we remove, by taking turns, one of them from $W_d$ and one from $V_d$ that approve of $c$ on level $i$ as long as possible and as long as

$$score^i_{(C,(V\setminus V_d)\cup W_d)}(d) \geq maj(V)$$

and (4) still hold. Note that we can skip this step if $W_d$ was not filled up with votes in earlier steps to bring $W_d$ and $V_d$ to the same size.

Secondly, we find the largest vote lists $W_{cd} \subseteq (W \setminus W_d)$ and $V_{cd} \subseteq (V \setminus V_d)$ such that:

(a) $|V_d \cup V_{cd}| \leq \ell$,
(b) $|V_{cd}| = |W_{cd}|$,
(c) all votes in $V_{cd}$ approve of $c$ on the first $i-1$ levels,
(d) all votes in $W_{cd}$ approve of $c$ on level $i$ or disapprove of $c$, and
(e) we have

$$score^i_{(C,(V\setminus(V_d\cup V_{cd}))\cup W_d\cup W_{cd})}(d) \geq \max\{maj(V), score^i_{(C,(V\setminus(V_d\cup V_{cd}))\cup W_d\cup W_{cd})}(c)+1\}.$$

Items (a), (b), and (e) make sure that we still have a valid replacement action and items (c) and (d) find the best votes to be added and removed such that $c$ loses approvals on the first $i-1$ levels.

Then we check the following condition:

$$score^{i-1}_{(C,(V\setminus(V_d\cup V_{cd}))\cup W_d\cup W_{cd})}(c) \geq maj(V). \tag{6}$$

If (6) holds, $c$ cannot be prevented from reaching a strict majority in the first $i-1$ levels without $d$ not reaching a strict majority or failing to beat $c$ on level $i$ as well.

Otherwise, $d$ still has a strict majority on level $i$ and $c$ cannot beat $d$ with a strict majority on earlier levels, so we output $(V_d \cup V_{cd}, W_d \cup W_{cd})$ as a successful pair.

*ApprovalStage*: This stage will only be reached if it was not possible to find a successful control action in majority stages 1 through $n$.

We first check whether the following holds:

$$score_{(C,V)}(c) - \ell < maj(V). \tag{7}$$

If (7) does not hold, we output "control impossible" since, after replacing at most $\ell$ suitable votes, (1) we could not find a candidate that beats $c$ in the majority stages and reaches a strict majority and (2) $c$ cannot be prevented from reaching a strict majority in overall approvals; so $c$ must win, no matter which at most $\ell$ votes are replaced.

Otherwise (i.e., if (7) holds), we iterate over all candidates $d \in C \setminus \{c\}$ and check whether

$$score_{(C,V)}(c) - 2\ell > score_{(C,V)}(d).$$

If this is the case, we skip $d$ and proceed to the next candidate or, if none is left, we output "control impossible" since then $d$ cannot catch up on his or her deficit to $c$.

Otherwise, we will try to make $d$ overtake $c$ in overall approvals while decreasing $c$'s overall approvals as much as possible in order to prevent $c$ from reaching a strict majority. We again determine the largest $W_d \subseteq W$ such that $|W_d| \leq \ell$ and all votes of $W_d$ approve of $d$ and disapprove of $c$. Furthermore, we again determine the largest $V_d \subseteq V$ such that

$|V_d| \leq \ell$ and all votes of $V_d$ approve of $c$ and disapprove of $d$. Once more, if $|V_d| \neq |W_d|$, we fill up the smaller vote list with votes as follows until they are equal in size:

- If $|V_d| < |W_d|$, we fill up $V_d$ with votes of $V \setminus V_d$ who approve of both $c$ and $d$ until we either have $|V_d| = |W_d|$ or run out of those votes, and in the latter case we now keep adding to $V_d$ those votes of $V \setminus V_d$ who approve of neither $c$ nor $d$. Only if this is still not enough to make the two lists equal in size, we remove votes from $W_d$ until both lists are equally large.
- If $|V_d| > |W_d|$, we fill up $W_d$ with votes of $W \setminus W_d$ that disapprove of both $c$ and $d$ until we either have $|V_d| = |W_d|$ or run out of those votes, and in the latter case we now keep adding to $W_d$ those votes of $W \setminus W_d$ that approve of both $c$ and $d$. We prefer adding votes disapproving both $c$ and $d$ over votes approving both $c$ and $d$ since the former type of votes keep $c$'s score as low as possible. Again, only if this is still not enough to make both vote lists equal in size, we remove votes from $V_d$ until both lists are equally large. Afterwards, if there are votes in $V \setminus V_d$ that approve of both $c$ and $d$ and votes in $W \setminus W_d$ that disapprove of both $c$ and $d$, we add as many as possible of them to $V_d$ and $W_d$, respectively, always ensuring that $|V_d| = |W_d|$ still holds. By doing this, we further reduce $c$'s score without changing the score balance of $c$ and $d$.

Then we check the following conditions:

$$score_{(C,(V \setminus V_d) \cup W_d)}(d) > score_{(C,(V \setminus V_d) \cup W_d)}(c), \tag{8}$$

$$score_{(C,(V \setminus V_d) \cup W_d)}(c) < maj(V). \tag{9}$$

If (8) and (9) are true, output $(V_d, W_d)$ since we have successfully prevented $c$ from reaching a strict majority and found a candidate $d$ that beats $c$ by approval score.

Otherwise, we proceed to the next candidate or, if none is left, output "control impossible."

Correctness of the algorithm follows from the explanations given during its description: The algorithm takes the safest way possible to guarantee that a YES-instance is verified. Clearly, the algorithm runs in polynomial time.

Turning to control by replacing candidates, fallback is resistant in both the constructive and the destructive case.

**Theorem 20** *Fallback-CCRC and Fallback-DCRC are* NP-*complete.*

***Proof*** Erdélyi and Rothe [22] (see also the subsequent journal version by Erdélyi et al. [16]) showed that fallback is resistant to constructive and destructive control by deleting candidates. Recall that in the former problem (denoted by Fallback-CCDC), we are given a fallback election $(C, V)$, a distinguished candidate $c \in C$, and an integer $\ell$, and we ask whether $c$ can be made a fallback winner by deleting at most $\ell$ votes, whereas in the destructive variant (denoted by Fallback-DCDC), for the same input we ask whether we can prevent $c$ from winning by deleting at most $\ell$ votes. To prove the theorem, we will reduce

**Table 12** Complexity of control for range voting (second row) and for normalized range voting (the third row). Our results are in boldface. "NPC" stands for "NP-complete," "P" for "polynomial-time solvable," and "I" for "immune"

| CCAV | CCDV | CCRV | CCAC | CCDC | CCRC | DCAV | DCDV | DCRV | DCAC | DCDC | DCRC |
|---|---|---|---|---|---|---|---|---|---|---|---|
| *NPC* | *NPC* | **NPC** | *I* | *P* | **P** | *P* | *P* | **P** | *P* | *I* | **P** |
| *NPC* | *NPC* | **NPC** | *NPC* | *NPC* | **NPC** | *P* | *P* | **P** | *NPC* | *NPC* | **NPC** |

– Fallback-CCDC to Fallback-CCRC and
– Fallback-DCDC to Fallback-DCRC, respectively.

Let $((C, V), c, \ell)$ be an instance of Fallback-CCDC (or Fallback-DCDC). We construct from $(C, V)$ a fallback election $(C \cup D, V')$ with (dummy) unregistered candidates $D = \{d_1, \ldots, d_\ell\}$, $D \cap C = \emptyset$, where we extend the votes of $V$ to the set of candidates $C \cup D$ by letting all voters disapprove of all candidates in $D$, thus obtaining $V'$. Our distinguished candidate remains $c$, and the deletion bound $\ell$ now becomes the limit on the number of candidates that may be replaced.

Since all candidates from $D$ are irrelevant to the election and can be added to the election without changing the winner(s), it is clear that $c$ can be made a fallback winner of $(C, V)$ by deleting up to $\ell$ candidates from $C$ if and only if $c$ can be made a fallback winner of $(C \cup D, V')$ by deleting up to $\ell$ candidates from $C$ and adding the same number of dummy unregistered candidates from $D$. This gives the desired reduction in both the constructive and the destructive case.

## 10 Range voting and normalized range voting

Now we study range voting and normalized range voting. Our results in this section are summarized in Table 12.

We first solve the cases in which range voting and normalized range voting have the same complexity and can be solved at one go starting with constructive control by replacing voters that follows from a result by Menton [48] that makes use of the fact that approval voting is a special case of range voting and normalized range voting.

**Theorem 21** **(Menton** [48]) *If approval voting is resistant to a case of control, range voting and normalized range voting will also be resistant for any scoring range.*

**Corollary 5** *Range-Voting-CCRV and Normalized-Range-Voting-CCRV are* NP*-complete.*

The destructive variant can be solved by a simple algorithm for range voting and normalized range voting.

**Theorem 22** *Range-Voting-DCRV and Normalized-Range-Voting-DCRV are in* P.

***Proof*** To prove membership in P of both problems, we will provide an algorithm that solves the problems in polynomial time and outputs, if possible, which of the registered voters must be replaced by which unregistered voters for $c$ to not win. The input to our algorithm is a $k$-range election $(C, V \cup W)$, the distinguished candidate $c \in C$, and an integer $\ell$. The algorithm will output either a pair $(V', W')$ with $V' \subseteq V$, $W' \subseteq W$, and $|V'| = |W'| \leq \ell$ (i.e., for $c$ to not win, there are votes in $V'$ that must be removed and votes in $W'$ that must be added to the election instead), or that control is impossible.

First, the algorithm checks whether $c$ is already not winning the election $(C, V)$ and outputs $(\emptyset, \emptyset)$ if this is the case, and we are done.

Otherwise (i.e., if $c$ is initially winning), we will try to find a candidate $d \in C \setminus \{c\}$ who can beat the distinguished candidate $c$ if voters are replaced. Since removing voters from or adding voters to the election does not affect the number of points (normalized or not) other voters give to the candidates, we can compute the change of the points balance (for range voting and normalized range voting, respectively) of $c$ and $d$ for each voter in $V \cup W$. Formally, let $v \in V \cup W$ and $s_c^v$ and $s_d^v$ be the (normalized) points given to $c$ and $d$ by voter $v$. Let $dist_{(C,\{v\})}(c, d) = s_c^v - s_d^v$ be the points difference that $c$ and $d$ would gain if $v$ were part of the election. Order the voters in $V$ and $W$, respectively, according to those values. Let $V' = \emptyset$ and $W' = \emptyset$. Then, in at most $\ell$ rounds, choose one vote $v \in V$ to remove from the election that maximizes the points balance in favor of $c$ (i.e., $v = \arg\max_{v \in V} dist_{(C,\{v\})}(c, d)$) and one vote from $w \in W$ to add to the election that maximizes the points balance in favor of $d$ (i.e., $w = \arg\min_{v \in V} dist_{(C,\{v\})}(c, d)$). If the replacement of $v$ with $w$ changes the points balance of $c$ and $d$ in favor of $d$ (i.e., if $dist_{(C,\{w\})}(c, d) - dist_{(C,\{v\})}(c, d) < 0$), set $V = V \setminus \{v\}$, $V' = V' \cup \{v\}$, $W = W \setminus \{w\}$, and $W' = W' \cup \{w\}$.

Afterwards, check whether $c$ is beaten by $d$ in $(C, (V \setminus V') \cup W')$ and output $(V', W')$ if this is the case. If there is no such candidate $d$, output that control is impossible. The algorithm solves the problems and runs in polynomial-time.

Turning now to control by replacing candidates, we start by examining constructive and destructive control for range voting and show that these problems are easy to solve. First note that Menton [48] showed that range voting (just like its special variant approval voting [33]) is immune to constructive control by adding candidates and to destructive control by deleting candidates. For control by *replacing* candidates, however, range voting is susceptible both in the constructive and in the destructive case, as shown in the following example.

***Example 2*** Consider a set $C = \{c, d\}$ of registered candidates, a set $D = \{e\}$ with only one unregistered candidate, and one voter $v$ with points vector (1, 2, 0), where $C \cup D$ is ordered lexicographically (i.e., $c$ gets one point, $d$ two, and $e$ zero points). If we are allowed to replace one candidate, $c$ loses in the 2-range election $(C, V)$ under range voting, but wins if $d$ is replaced by $e$. This shows that range voting is susceptible to constructive control by replacing candidates.

We can use the same candidate sets $C$ and $D$ and the points vector (1, 0, 2) for $v$ to show susceptibility of range voting for destructive control by replacing candidates analogously.

**Theorem 23** *Range-Voting-CCRC and Range-Voting-DCRC are in* P.

***Proof*** For range voting, adding or removing candidates does not affect the points given to other candidates. Therefore, for an input of Range-Voting-CCRC and Range-Voting-DCRC, respectively, we do the following after checking whether the chair's constructive or destructive goal is reached trivially (and accepting in this case).

In the constructive case, we need to check whether the number of registered candidates that beat the distinguished candidate $c$ is at most $\ell$ and whether there are enough unregistered candidates that do not beat $c$ so that each of them can replace one registered candidate beating $c$. If so, we accept; otherwise, control is impossible.

In the destructive case, we check if there exists an unregistered candidate $d$ that beats $c$; if so, we choose an arbitrary registered candidate that is not $c$ and replace this candidate by $d$; otherwise, control is impossible.

In contrast to range voting, we now show that normalized range voting is resistant to constructive and destructive control by replacing candidates. Starting with constructive control, we adapt a reduction by Menton [48] to reduce Hitting-Set to Normalized-Range-Voting-CCRC.

**Theorem 24** *Normalized-Range-Voting-CCRC is* NP-*complete.*

***Proof*** The reduction is a simple modification of the reduction that Menton [48] used to show that normalized range voting is resistant to constructive control by adding candidates.

Given a Hitting-Set instance $(U, \mathscr{S}, \kappa)$, construct a Normalized-Range-Voting-CCRC instance as follows. Let $C = E \cup \{c, w\}$ with $E = \{e_1, \dots, e_\kappa\}$ be the set of registered candidates and $D = U$ the set of unregistered candidates.

- $2t(\kappa + 1) + 4s$ voters give a score of 2 to $c$ and each $e_i \in E$, and a score of 0 to all other candidates;
- $3t(\kappa + 1) + 2\kappa$ voters give a score of 2 to $w$ and each $e_i \in E$, and a score of 0 to all other candidates;
- for each $b \in U$, 4 voters give a score of 2 to $b$ and each $e_i \in E$, a score of 1 to $w$, and a score of 0 to all other candidates; and
- for each $S_i \in \mathscr{S}$, $2(\kappa + 1)$ voters give a score of 2 to each $b \in S_i$ and each $e_i \in E$, a score of 1 to $c$, and a score of 0 to all other candidates.

The voters are exactly the same as in the reduction for Normalized-Range-Voting-CCAC of Menton [48] (the number of voters in the second group are adjusted to the nonunique-winner model) except that every voter gives the candidates from $E$ the maximum number of points. Since $w$ gains zero points from the second group of voters in order for $w$ to have a chance of winning, all candidates from $E$ need to be removed. Together with the fact that we can pad every solution of the Hitting-Set instance to contain exactly $\kappa$ elements of $U$ we can conclude that $(U, \mathscr{S}, \kappa)$ is a YES-instance of Hitting-Set if and only if $((C \cup D, V), w, \kappa)$ is a YES-instance of Normalized-Range-Voting-CCRC.

For the destructive variant we can use the NP-hardness of Normalized-Range-Voting-DCDC proven by Menton [48].

**Theorem 25** *Normalized-Range-Voting-DCRC is* NP-*complete.*

***Proof*** To show NP-hardness we will reduce Normalized-Range-Voting-DCDC to Normalized-Range-Voting-DCRC. Given a Normalized-Range-Voting-DCDC instance $((C, V), c, \ell)$, construct a set of unregistered candidates $D$ with $|D| = \ell$ and let every voter $v \in V$ give every candidate from $D$ as many points as he or she gives to $c$. Therefore, $c$ and every candidate from $D$ will always have the same number of points. Since $c$ is always part of the election (removing $c$ would trivially achieve the destructive goal), adding any candidate of $D$ never affects the number of points given to other candidates. Therefore, if at most $\ell$ candidates from $C \setminus \{c\}$ can be removed from the election $(C, V)$ to make $c$ not win (i.e., $((C, V), c, \ell)$ is a YES-instance of Normalized-Range-Voting-DCDC), we can add the same number of candidates from $D$ to the election without changing the winners, so $((C \cup D, V), c, \ell)$ is a YES-instance of Normalized-Range-Voting-DCRC. For the converse direction, if we cannot make $c$ be beaten in $(C, V)$ by removing at most $\ell$ candidates, we cannot do so by adding candidates from $D$. Menton [48] showed that Normalized-Range-Voting-DCDC is NP-hard. Thus the theorem is proven.

## 11 Conclusions and open problems

We have investigated the computational complexity of control for Copeland$^\alpha$, maximin, $k$-veto, plurality with runoff, veto with runoff, Condorcet, fallback, range voting, and normalized range voting, closing a number of gaps in the literature. Table 1 on page 5 in Sect. 2 gives an overview of our and previously known results on the complexity of control by replacing, adding, and deleting either candidates or voters for the voting rules mentioned above.

Our proofs are based on the nonunique-winner model but can be modified to work for the unique-winner model of the control problems as well. Notice that the complexity of CCRV for 2-approval remains the only open problem in Table 1. The polynomial-time algorithm for 2-Veto-CCRV from the proof of Theorem 5 cannot be trivially extended to 2-approval. In 2-veto, any optimal solution only replaces registered voters in $V$ that veto the distinguished candidate. However, this is not the case in 2-approval. In a worst case, we need to replace registered votes in $V$ that do not approve of $c$ with some unregistered votes in $W$ that also do not approve of $c$. It is not clear how to reduce such a worst-case instance to an equivalent $b$-EC instance.

We point out that the complexity of partitioning either candidates or voters (in the various scenarios due to Bartholdi, Tovey, and Trick [7] and Hemaspaandra, Hemaspaandra, and Rothe [33]) is still open for plurality with runoff and veto with runoff. In addition, it would also be 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 provide purely a worst-case analysis and whether these problems are hard to solve in practice needs to be further investigated. Finally, our polynomial-time algorithm in Theorem 9 relies on that ties are broken in favor of the chair. It would be 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, 52, 63].

**Acknowledgements** We thank the anonymous JAAMAS, AAMAS'19, CSR'20, and ISAIM'20 reviewers for their helpful comments. This work was supported in part by DFG Grants RO-1202/14-2 and RO-1202/21-1.

**Funding** Open Access funding enabled and organized by Projekt DEAL.

# References


1. Ahuja, R., Magnanti, T., & Orlin, J. (1993). *Network Flows: Theory, Algorithms, and Applications*. New Jersey: Prentice-Hall.
2. Arora, S., & Barak, B. (2009). *Computational Complexity: A Modern Approach*. Cambridge: Cambridge University Press.
3. Aziz, H., Gaspers, S., Mattei, N., Narodytska, N., & Walsh, T. (2013) Ties matter: Complexity of manipulation when tie-breaking with a random vote. In: Proceedings of the 27th AAAI Conference on Artificial Intelligence, pp. 74–80
4. Bang-Jensen, J., & Gutin, G. (2008). *Digraphs: Theory*. Berlin: Springer-Verlag.
5. Bartholdi, J., & III., & Orlin, J. . (1991). Single transferable vote resists strategic voting. *Social Choice and Welfare, 8*(4), 341–354.
6. Bartholdi, J., & III., Tovey, C., & Trick, M. . (1989). The computational difficulty of manipulating an election. *Social Choice and Welfare, 6*(3), 227–241.
7. Bartholdi, J., & III., Tovey, C., & Trick, M. . (1992). How hard is it to control an election? *Mathematical and Computer Modelling, 16*(8/9), 27–40.
8. Baumeister, D., & Rothe, J. (2015). Preference aggregation by voting. In: J. Rothe (ed.) Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, Springer Texts in Business and Economics, chap. 4, pp. 197–325. Springer-Verlag
9. Betzler, N., & Uhlmann, J. (2009). Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science, 410*(52), 5425–5442.
10. Brams, S., & Sanver, R. (2009). Voting systems that combine approval and preference. In: S. Brams, W. Gehrlein, F. Roberts (eds.) The Mathematics of Preference, Choice, and Order: Essays in Honor of Peter C. Fishburn, pp. 215–237. Springer
11. Chen, J., Faliszewski, P., Niedermeier, R., & Talmon, N. (2017). Elections with few voters: Candidate control can be easy. *Journal of Artificial Intelligence Research, 60,* 937–1002.
12. Conitzer, V., & Walsh, T. (2016). Barriers to manipulation in voting. In: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (eds.) Handbook of Computational Social Choice, chap. 6, pp. 127–145. Cambridge University Press
13. Downey, R., & Fellows, M. (2013). *Parameterized Complexity* (2nd ed.). Springer-Verlag.
14. Dwork, C., Kumar, R., Naor, M., & Sivakumar, D. (2001). Rank aggregation methods for the web. In: Proceedings of the 10th International World Wide Web Conference, pp. 613–622
15. Ephrati, E., & Rosenschein, J. (1997). A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence, 20*(1–4), 13–67.
16. Erdélyi, G., Fellows, M., Rothe, J., & Schend, L. (2015). Control complexity in Bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences, 81*(4), 632–660.
17. Erdélyi, G., Fellows, M., Rothe, J., & Schend, L. (2015). Control complexity in Bucklin and fallback voting: An experimental analysis. *Journal of Computer and System Sciences, 81*(4), 661–670.
18. Erdélyi, G., Hemaspaandra, E., & Hemaspaandra, L. (2015). More natural models of electoral control by partition. In: Proceedings of the 4th International Conference on Algorithmic Decision Theory, pp. 396–413
19. Erdélyi, G., Nowak, M., & 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.
20. Erdélyi, G., Piras, L., & Rothe, J. (2011). The complexity of voter partition in Bucklin and fallback voting: Solving three open problems. In: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems, pp. 837–844
21. Erdélyi, G., Reger, C., & Yang, Y. (2019). Towards completing the puzzle: Solving open problems for control in elections. In: Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, pp. 846–854
22. Erdélyi, G., & Rothe, J. (2010). Control complexity in fallback voting. In: Proceedings of Computing: the 16th Australasian Theory Symposium, pp. 39–48
23. Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2011). Multimode control attacks on elections. *Journal of Artificial Intelligence Research, 40,* 305–351.
24. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009). Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research, 35,* 275–341.
25. Faliszewski, P., & Rothe, J. (2016). Control and bribery in voting. In: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (eds.) Handbook of Computational Social Choice, chap. 7, pp. 146–168. Cambridge University Press

26. Gabow, H. (1983). An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In: Proceedings of the 15th ACM Symposium on Theory of Computing, pp. 448–456
27. Garey, M., & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman and Company: W. H.
28. Ghosh, S., Mundhe, M., Hernandez, K., & Sen, S. (1999). Voting for movies: The anatomy of recommender systems. In: Proceedings of the 3rd Annual Conference on Autonomous Agents, pp. 434–435
29. Gonzalez, T. (1985). Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science, 38,* 293–306.
30. Grötschel, M., Lovász, L., & Schrijver, A. (1988). *Geometric Algorithms and Combinatorial Optimization*. Berlin: Springer.
31. Guo, J., & Shrestha, Y.R. (2014). Controlling two-stage voting rules. In: Proceedings of the 21st European Conference on Artificial Intelligence, pp. 411–416
32. Haynes, T., Sen, S., Arora, N., & Nadella, R. (1997). An automated meeting scheduling system that utilizes user preferences. In: Proceedings of the 1st International Conference on Autonomous Agents, pp. 308–315
33. Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2007). Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence, 171*(5–6), 255–285.
34. Hemaspaandra, E., Hemaspaandra, L., & Schnoor, H. (2014). A control dichotomy for pure scoring rules. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 712–720
35. Hemaspaandra, E., Hemaspaandra, L. A., & Menton, C. (2020). Search versus decision for election manipulation problems. *ACM Transactions on Computation Theory, 12*(1), 3:1-3:42.
36. Hemaspaandra, L. (2018). Computational social choice and computational complexity: BFFs? In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence, pp. 7971–7977
37. Karp, R. (1972). Reducibility among combinatorial problems. In: R. Miller, J. Thatcher (eds.) Complexity of Computer Computations, pp. 85–103
38. Lang, J., Maudet, N., & Polukarov, M. (2013). New results on equilibria in strategic candidacy. In: Proceedings of the 6th International Symposium on Algorithmic Game Theory, pp. 13–25
39. Lin, A. (2011). The complexity of manipulating *k*-approval elections. In: Proceedings of the 3rd International Conference on Agents and Artificial Intelligence, pp. 212–218
40. Loreggia, A. (2012). Iterative voting and multi-mode control in preference aggregation. Master's thesis, University of Padova
41. Loreggia, A. (2014). Iterative voting and multi-mode control in preference aggregation. *Intelligenza Artificiale, 8*(1), 39–51.
42. Loreggia, A. (2016). Iterative voting, control and sentiment analysis. Ph.D. thesis, University of Padova
43. Loreggia, A., Narodytska, N., Rossi, F., Venable, B., & Walsh, T. (2015). Controlling elections by replacing candidates or votes (extended abstract). In: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, pp. 1737–1738
44. Magiera, K., & Faliszewski, P. (2017). How hard is control in single-crossing elections? *Journal of Autonomous Agents and Multi-Agent Systems, 31*(3), 606–627.
45. Maushagen, C., & 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, pp. 277–285
46. Maushagen, C., & Rothe, J. (2018). Complexity of control by partitioning veto elections and of control by adding candidates to plurality elections. *Annals of Mathematics and Artificial Intelligence, 82*(4), 219–244.
47. Maushagen, C., & Rothe, J. (2020). The last voting rule is home: Complexity of control by partition of candidates or voters in maximin elections. In: Proceedings of the 24th European Conference on Artificial Intelligence, pp. 163–170
48. Menton, C. (2013). Normalized range voting broadly resists control. *Theory of Computing Systems, 53*(4), 507–531.
49. Menton, C., & Singh, P. (2013). Control complexity of Schulze voting. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 286–292
50. Neveling, M., Rothe, J., & Zorn, R. (2020). The complexity of controlling Condorcet, fallback, and *k*-veto elections by replacing candidates or voters. In: Proceedings of the 15th International Computer Science Symposium in Russia, pp. 314–327
51. Niedermeier, R. (2006). *Invitation to Fixed-Parameter Algorithms*. Oxford: Oxford University Press.

52. Obraztsova, S., Elkind, E., & Hazon, N. (2011). Ties matter: Complexity of voting manipulation revisited. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 2698–2703
53. Oflazer, K., & Tür, G. (1997). Morphological disambiguation by voting constraints. In: Proceedings of the *8th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 222–229
54. Parkes, D., & Xia, L. (2012). A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In: Proceedings of the 26th AAAI Conference on Artificial Intelligence, pp. 1429–1435
55. Pennock, D., Horvitz, E., & Giles, C. (2000). Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In: Proceedings of the 17th National Conference on Artificial Intelligence, pp. 729–734
56. Rothe, J. (2005). Complexity Theory and Cryptology. An Introduction to Cryptocomplexity. EATCS Texts in Theoretical Computer Science. Springer-Verlag
57. Sigletos, G., Paliouras, G., Spyropoulos, C., & Hatzopoulos, M. (2005). Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research, 6,* 1751–1782.
58. Tovey, C. (2002). Tutorial on computational complexity. *Interfaces, 32*(3), 30–61.
59. West, D. (2000). *Introduction to Graph Theory*. New Jersey: Prentice-Hall.
60. Yang, Y. (2017). The complexity of control and bribery in majority judgment. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, pp. 1169–1177
61. Yang, Y., & Guo, J. (2014).Controlling elections with bounded single-peaked width. In: Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems, pp. 629–636
62. Yang, Y., & Guo, J. (2017). The control complexity of $r$-approval: From the single-peaked case to the general case. *Journal of Computer and System Sciences, 89,* 432–449.
63. Yang, Y., & Wang, J. (2017). Anyone but them: The complexity challenge for a resolute election controller. In: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, pp. 1133–1141




## Authors and Affiliations


**Gábor Erdélyi[1] · Marc Neveling[2] · Christian Reger[3] · Jörg Rothe[2] · Yongjie Yang[4] · Roman Zorn[2]**

Gábor Erdélyi
gabor.erdelyi@canterbury.ac.nz

Marc Neveling
marc.neveling@hhu.de

Christian Reger
christian.reger@ymail.com

Jörg Rothe
rothe@hhu.de

Roman Zorn
roman.zorn@hhu.de

1 School of Mathematics and Statistics, University of Canterbury, Christchurch, New Zealand

2 Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany

3 School of Economic Disciplines, University of Siegen, Siegen, Germany

4 Chair of Economic Theory, Saarland University, Saarbrücken, Germany

# Chapter 4

# Complexity of Control for Single Nontransferable Vote and Bloc Voting

## 4.1 Summary

While previous literature on electoral control focused mostly on single-winner elections, this work concentrated on electoral control in multiwinner elections. Although some results in this area were already known, they often relied on the complexity of the single-winner variant of the multiwinner voting rule. Since for a committee size of one, SNTV is equivalent to Plurality and Bloc voting is equivalent to $k$-approval, the complexity results from those single-winner voting rules would also apply to their multiwinner variant (as we are dealing with worst-case complexities).

The idea of this work was to exclude the single-winner case from the definition of multiwinner voting rules, as one would only use a multiwinner voting rule if one intended to find a winning committee with multiple members, not just a single winner. Using this revised definition of multiwinner voting rules, where $k \geq 2$, we investigated SNTV and Bloc voting with regard to their resistance or vulnerability to control by all types of adding, deleting or replacing both voters and candidates. In doing so, we not only solidified the already existing results based on single-winner voting rules but also found new results for control by replacing candidates and voters – two control problems that had not previously been studied in the context of multiwinner voting.

## 4.2 Personal Contribution

The writing of this work was carried out primarily by me, with finalization and polishing by my co-authors Jörg Rothe and Garo Karh Bet. The initial technical results for all cases of candidate control were established by Garo Kar Bet, under my supervision and with my feedback. The results for the cases of replacing voters were developed solely by me.

## 4.3 Publication

G. Karh Bet, J. Rothe, and R. Zorn. "Complexity of Control for Single Nontransferable Vote and Bloc Voting". In: *Annals of Mathematics and Artificial Intelligence* (Submitted)

A preliminary version of this work has been submitted to and accepted at the *8th International Conference on Algorithmic Decision Theory*:

G. Karh Bet, J. Rothe, and R. Zorn. "Complexity of Candidate Control for Single Nontransferable Vote and Bloc Voting". In: *8th International Conference on Algorithmic Decision Theory (ADT 2024)*. Cham: Springer Nature Switzerland, 2024.

# Complexity of Control for Single Nontransferable Vote and Bloc Voting

Garo Karh Bet[1], Jörg Rothe[1*], Roman Zorn[1]

[1]Institut für Informatik, MNF, Heinrich-Heine-Universität Düsseldorf, Universitätsstr. 1, Düsseldorf, 40225, Germany.

*Corresponding author(s). E-mail(s): rothe@hhu.de;
Contributing authors: garo.karh.bet@hhu.de; roman.zorn@hhu.de;

**Abstract**

Electoral control is a scenario where an election chair changes the structure of an election by actions such as adding or deleting either candidates or voters with the goal of either making a favorite candidate win or precluding a despised candidate's victory. Much work has been done on the computational complexity of controlling elections for single-winner voting rules, yet much less work on the control complexity for multiwinner voting rules which aim at electing not only a single winner but a winning committee of candidates. Meir et al. [1] initiated the investigation of electoral control for multiwinner voting rules, including single nontransferable vote (SNTV) and bloc voting. We study these two rules with a focus on their multiwinner aspect and with respect to control by adding, deleting, and replacing candidates or voters.

**Keywords:** multiwinner voting, electoral control, single nontransferable vote, bloc voting

## 1 Introduction

Computational social choice [2, 3] is an interdisciplinary field at the interface of artificial intelligence and theoretical computer science on the one hand and economics and social choice theory on the other, which focuses on the computational aspects of collective decision-making mechanisms such as voting. Application scenarios range over a wide spectrum and include, for instance, electing a leader of a group of people or an organisation, or selecting a meeting time and place for a group of people, or when judges or referees short-list the finalists of a competition based on their performance [4, 5]. Moreover, voting has been used as a decision-making mechanism in various computational settings such as planning [6, 7] and collaborative filtering [8], and also in several large-scale computer settings, including web-page rank aggregation and the related spam reduction and similarity-search problems [9, 10]. In such scenarios, particular attention has been paid to ways of tampering with the outcome of elections by manipulation, control, and bribery [11–13]. We focus on electoral control—a scenario where the structure of an election can be changed by an election chair via control actions such as adding or deleting either candidates or voters. Bartholdi et al. [14] introduced and studied the constructive variant of the problem where the chair's goal is to make a favorite candidate win. Hemaspaandra et al. [15] introduced the destructive counterpart that aims at precluding a despised candidate's victory. We investigate these scenarios when applied to the multiwinner voting rules single nontransferable vote and bloc voting, extending previous work by Meir et al. [1].

***Related work.***

Much work has been done on studying the complexity of electoral control for single-winner voting rules, especially for the *standard* control actions of adding, deleting, or partitioning either the candidates or the voters (see the book chapters by Faliszewski and Rothe [12] and Baumeister and Rothe [13] for an overview). As mentioned above, Bartholdi et al. [14] introduced the constructive variants of control and were the first to study them for the plurality and Condorcet voting rules. Hemaspaandra et al. [15] complemented their results by considering the destructive counterparts of these scenarios, and they also studied constructive and destructive control for approval voting [16–18]. Since then, a wide variety of other voting rules have also been studied in terms of their control complexity: Copeland and Llull voting by Faliszewski et al. [19]; Borda by, e.g., Elkind et al. [20] and Neveling and Rothe [21]; maximin voting (a.k.a. the Simpson–Kramer rule) and veto by Faliszewski et al. [22] and Maushagen and Rothe [23–25]; $k$-approval (which, unlike approval voting but like plurality, veto, or Borda, is a positional scoring protocol) by Lin [26, 27]; range voting and normalized range voting by Menton [28]; Bucklin and fallback voting by Erdélyi et al. [29, 30]; and Schulze and ranked-pairs voting by Menton and Singh [31], Hemaspaandra et al. [32], and Maushagen et al. [33].

However, much less is known about the complexity of control for multiwinner voting rules where not only a single winner is to be elected but a winning committee of candidates [4, 5]. Our main goal is to study the complexity of control by *replacing* candidates or voters for two fundamental multiwinner voting rules: *single nontransferable vote (SNTV)* and *bloc voting*, the multiwinner analogues of plurality and $k$-approval voting, respectively. Control by replacing candidates or voters was introduced and first studied by Loreggia et al. [34, 35]. This control type—which combines adding and deleting either candidates or voters but with the constraint that the same number of candidates or voters must be added as has been deleted—has only been studied for single-winner voting so far, see, e.g., the recent work of Erdélyi et al. [36] and Maushagen et al. [33]. Meir et al. [1] were the first to explicitly address the complexity of control of *multiwinner* elections, including SNTV and bloc voting. Subsequently, Yang [37] investigated the complexity of manipulation and control as well as the parameterized complexity of control for various approval-based multiwinner voting rules.

Although one can consider some multiwinner voting rules to be generalizations of their single-winner variants, they have been studied much less and only very recently. As stated above, almost all previous work that has studied the complexity of electoral control focused on *single-winner* elections and considered the constructive and destructive cases of the standard control types; the scenarios of *replacing* either candidates or voters have been studied only since 2015 when this control type was introduced [35]. Bearing all this in mind, one can conclude that there is an obvious gap in the literature which we aim to fill: the study of constructive and destructive control of *multiwinner* elections by *replacing* either candidates or voters. To this end, we will show the NP-hardness of various control problems in the multiwinner setting for SNTV and bloc voting. Hopefully, this will pave the way toward further investigation and exploration of multiwinner electoral control.

We will use a general result by Loreggia [34] (stated here as Theorem 1) that links control by replacing candidates to control by adding or deleting candidates, provided that the voting rule is insensitive to bottom-ranked candidates, a property introduced by Lang et al. [38]. Therefore, we also need to explore the complexity of control by adding or deleting candidates for SNTV and bloc voting. Note that Meir et al. [1] in particular observed such results as a simple consequence of the corresponding results in single-winner elections. However, they model these control problems differently than it was done in the original papers on control [14, 15], using additive utility functions that assign integers to the candidates. That is why we formalize these standard control problems in the original model and adapt the original proofs in the single-winner setting to the multiwinner case. Another difference is that we exclude the committee size one (which is the single-winner case), so unlike the work of Meir et al. [1], our results in the multiwinner setting are not immediate consequences of the corresponding single-winner results—we only consider elections with strictly multiple winners to be members of a winning committee.

### *Our contributions.*

Our results are summarized in Table 1, where "R" stands for *resistant* (indicating that the corresponding control problem is NP-hard) and "V" stands for *vulnerable* (indicating that the corresponding control problem is in P).

**Table 1** Complexity results for SNTV and bloc voting, for a target committee size $k$ with $2 \leq k \leq |C| - 1$

| Problem | SNTV | Bloc voting |
|---|---|---|
| CCAC | R (Theorem 4) | R (Theorem 7) |
| DCAC | R (Theorem 3) | R (Theorem 6) |
| CCDC | R (Theorem 2) | R (Theorem 5) |
| DCDC | R (Theorem 4) | R (Theorem 7) |
| CCRC | R (Corollary 1) | R (Corollary 3) |
| DCRC | R (Corollary 2) | R (Corollary 4) |
| CCAV | V (Meir et al. [1]) | R (Meir et al. [1]) |
| DCAV | V (Meir et al. [1]) | R (Meir et al. [1]) |
| CCDV | V (Meir et al. [1]) | R (Meir et al. [1]) |
| DCDV | V (Meir et al. [1]) | R (Meir et al. [1]) |
| CCRV | V (Theorem 8) | R (Theorem 10) |
| DCRV | V (Theorem 9) | R (Theorem 11) |

A preliminary version of this paper has been presented at the *8th International Conference on Algorithmic Decision Theory* [39]. The current version substantially extends this conference version by the results on control by replacing voters in Theorems 8–11.

# 2 Preliminaries

In this section, we first introduce some background from social choice theory—in particular, multiwinner elections and those multiwinner voting rules we are going to study here (for more background on multiwinner voting, see, e.g., the book chapters by Baumeister et al. [4] and Faliszewski et al. [5]) and then formally define the problems whose complexity we will study here.

## 2.1 Multiwinner Elections and Voting Rules

A *multiwinner election* is a triple $E = (C,V,k)$, where $C$ and $V$ stand for the set of candidates and the list of votes (or voters), respectively, and $k$, $2 \leq k \leq |C| - 1$, refers to the (fixed) target committee size. Mathematically speaking, $V$ is expressed by a (strict) linear order on $C$, satisfying the properties of connectivity, transitivity, and asymmetry. Since for $k = 1$ one would be back to the case of single-winner elections, and for $k = |C|$ the solution is trivial, we are going to consider only target committee size values of $1 < k < |C|$.

A *multiwinner voting rule* $\mathcal{R}$ is a function that maps a given multiwinner election $E = (C,V,k)$ to a set $\mathcal{R}(E)$ of $k$-element subsets of $C$, which are referred to as *winning committees*. This is denoted by $\mathcal{R}(E) = W = \{W_1, \dots, W_\ell\}$, where $|W_i| = k$, $1 \leq i \leq \ell$. A multiwinner voting rule outputs all the winning committees that could end up winning for some way of resolving ties that may occur while executing the rule; we call them *possible winning committees*.

Let $E = (C,V,k)$ be a multiwinner election and $W = \{W_1, \dots, W_\ell\}$ the set of all possible winning committees with $|W_i| = k$. We call a candidate $c \in C$ a *certain winner* if $c$ is a member of every possible winning committee, i.e., $c \in W_i$ for each $W_i \in W$. Note that for a target committee size $k$, there can be at most $k$ certain winners in a given multiwinner election. We call $c$ an *uncertain winner* if $c$ is in some, yet not in all possible winning committees, i.e., $c \in \bigcup_{1 \leq i \leq \ell} W_i$ and $c \notin W_i$ for some $i$, $1 \leq i \leq \ell$. Similarly, we call $c$ a *certain nonwinner* if there is no possible winning committee $W_i \in W$ such that $c \in W_i$, i.e., $c \notin W_i$ for any $W_i \in W$.

We consider the following two multiwinner voting rules:

- *Single nontransferable vote (SNTV)* is the multiwinner variant of plurality, since for committee size $k$, it returns the $k$ candidates with the highest plurality scores, where the *plurality score* $\sigma(c)$ *of a candidate* $c \in C$ is the number of votes in which $c$ is ranked first.[1]
- *Bloc voting* is the multiwinner variant of $k$-approval: For committee size $k$, bloc voting returns the $k$ candidates with the highest $k$-approval scores, where the *$k$-approval score* $\alpha_k(c)$ *of a candidate* $c \in C$ is the number of votes in which $c$ is ranked among the first $k$ positions.

[1] Occasionally (e.g., in the proof of Theorem 8), we will explicitly specify the election the plurality score $\sigma(c)$ refers to by writing, e.g., $\sigma_{(C,V)}(c)$. However, whenever this election is clear from the context, we omit this subscript.

## 2.2 Control Problems

Meir et al. [1] model control for multiwinner voting rules via utility functions. In contrast, we follow the original model of Bartholdi et al. [14] and Hemaspaandra et al. [15] for control in single-winner voting when defining the following four multiwinner control problems for any multiwinner voting rule $\mathcal{R}$ and fixed committee size $k \geq 2$. Note that in the multiwinner elections considered in these problems, all votes are tacitly assumed to contain only those candidates participating in these elections. For example, if $C = \{a,b,c\}$, $V$ contains two votes, $a\ b\ c$ and $b\ c\ a$, and $C' = \{b\}$, then we write $(C \setminus C', V, k)$ in $\mathcal{R}$-CCDC to mean the election $(\{a,c\},(a\ c, c\ a),k)$.

| $\mathcal{R}$-CONSTRUCTIVE-CONTROL-BY-ADDING-CANDIDATES ($\mathcal{R}$-CCAC) | |
|---|---|
| **Given:** | Two sets of candidates, $C$ and $D$ with $C \cap D = \emptyset$, a list $V$ of votes over $C \cup D$, committee size $k$, a distinguished candidate $c \in C$, and a positive integer $r \leq \|D\|$. |
| **Question:** | Is it possible to add at most $r$ candidates from $D$ to $C$ such that $c$ is a certain $\mathcal{R}$ winner of the resulting election? That is, is there a subset $D' \subseteq D$ with $\|D'\| \leq r$ such that $c$ is a certain $\mathcal{R}$ winner of the election $(C \cup D', V, k)$? |

| $\mathcal{R}$-CONSTRUCTIVE-CONTROL-BY-DELETING-CANDIDATES ($\mathcal{R}$-CCDC) | |
|---|---|
| **Given:** | A set of candidates $C$, a list $V$ of votes over $C$, committee size $k$, a distinguished candidate $c \in C$, and a positive integer $r \leq \|C\|$. |
| **Question:** | Is it possible to delete at most $r$ candidates from $C$ such that $c$ is a certain $\mathcal{R}$ winner of the resulting election? That is, is there a subset $C' \subseteq C$ with $\|C'\| \leq r$ such that $c$ is a certain $\mathcal{R}$ winner of the election $(C \setminus C', V, k)$? |

| $\mathcal{R}$-CONSTRUCTIVE-CONTROL-BY-REPLACING-CANDIDATES ($\mathcal{R}$-CCRC) | |
|---|---|
| **Given:** | Two sets of candidates, $C$ and $D$ with $C \cap D = \emptyset$, a list $V$ of votes over $C \cup D$, committee size $k$, a distinguished candidate $c \in C$, and a positive integer $r \leq \|C\|$. |
| **Question:** | Are there subsets $C' \subseteq C$ and $D' \subseteq D$ such that $\|C'\| = \|D'\| \leq r$ and $c$ is a certain $\mathcal{R}$ winner of the election $((C \setminus C') \cup D', V, k)$? |

| $\mathcal{R}$-CONSTRUCTIVE-CONTROL-BY-REPLACING-VOTERS ($\mathcal{R}$-CCRV) | |
|---|---|
| **Given:** | A set of candidates $C$, two lists $V$ and $U$ of votes over $C$, committee size $k$, a distinguished candidate $c \in C$, and a positive integer $r \leq \|V\|$. |
| **Question:** | Are there sublists $V' \subseteq V$ and $U' \subseteq U$ such that $\|V'\| = \|U'\| \leq r$ and $c$ is a certain $\mathcal{R}$ winner of the election $(C, (V \setminus V') \cup U', k)$? |

The destructive variants of these four problems, denoted by $\mathcal{R}$-DESTRUCTIVE-CONTROL-BY-ADDING-CANDIDATES ($\mathcal{R}$-DCAC) etc., are defined analogously, except that we replace "certain $\mathcal{R}$ winner" in the question field by "certain $\mathcal{R}$ nonwinner."

A (multiwinner) voting rule $\mathcal{R}$ is said to be *susceptible* to some control type if the chair can successfully exert control for at least some instance of the corresponding control problem; otherwise, $\mathcal{R}$ is said to be *immune* to it. If $\mathcal{R}$ is susceptible to some control type, it is said to be *vulnerable* to it if the corresponding control problem is in P (i.e., solvable in *deterministic polynomial time*), and $\mathcal{R}$ is said to be *resistant* to this control type if the corresponding control problem is NP-hard (where NP is the complexity class *nondeterministic polynomial time*).

In our proofs, we will use the following standard NP-complete problems (see, e.g., the textbooks by Garey and Johnson [40], Papadimitriou [41], or Rothe [42]):

| HITTING-SET | |
|---|---|
| **Given:** | A set $B = \{b_1, \dots, b_m\}$, a family $\mathcal{S} = \{S_1, \dots, S_n\}$ of subsets $S_i \subseteq B$, and a positive integer $r$. |
| **Question:** | Is there a subset $B' \subseteq B$, $\|B'\| \leq r$, such that each $S_i \in \mathcal{S}$ is *hit* by $B'$, i.e., $S_i \cap B' \neq \emptyset$ for all $S_i \in \mathcal{S}$? |

| EXACT-COVER-BY-THREE-SETS (X3C) | |
|---|---|
| **Given:** | A set $B = \{b_1, \dots, b_m\}$ with $m = 3r$ and $r \geq 1$ and a family $\mathcal{S} = \{S_1, \dots, S_n\}$ of subsets $S_i \subseteq B$ with $\|S_i\| = 3$, for each $i$, $1 \leq i \leq n$. |
| **Question:** | Is there a subfamily $\mathcal{S}' \subseteq \mathcal{S}$ such that every element of $B$ appears in exactly one subset of $\mathcal{S}'$? |

In our resistance proofs for constructive and destructive control by replacing candidates, we will make use of voting rules being *insensitive to bottom-ranked candidates (IBC)*, a notion due to Lang et al. [38].

**Definition 1** *A voting rule $\mathcal{R}$ is* IBC *if its set of winners does not change after adding or deleting a subset of candidates at the bottom of the preference profile.*

Loreggia [34] shows how this property can be used to relate control by adding or deleting candidates to control by replacing candidates.

**Theorem 1 (Theorems 3.3.3 and 3.3.4 in the PhD thesis of Loreggia [34])**

1. *Every voting rule that is IBC and resistant to constructive control by deleting candidates is also resistant to constructive control by replacing candidates.*
2. *Every voting rule that is IBC and resistant to destructive control by adding or by deleting candidates is also resistant to destructive control by replacing candidates.*

## 3 Control by Adding, Deleting, or Replacing Candidates

We begin with the cases for candidate control, specifically for SNTV. To prove its resistance to constructive and destructive control by replacing candidates, Theorem 1 will be used. To this end, SNTV's insensitivity to bottom-ranked candidates will be shown first.

**Lemma 1** *SNTV is insensitive to bottom-ranked candidates.*

**Proof.** This is clear from the definition, as SNTV only considers the first rank of any votes. Adding or deleting candidates at the bottom of a vote has no influence on the winning committees. ❑

Additionally, we now need to show that SNTV is resistant to constructive control by deleting candidates and destructive control by adding candidates. Meir et al. [1] already observed that these two resistances immediately follow from the corresponding resistance results in the single-winner case [14, 15]. However, we exclude the case of $k = 1$ from our definition, so this argument does not apply here; further, based on the original definitions, we have modeled our control problems somewhat differently. We now show that SNTV is resistant to constructive control by deleting candidates.

**Theorem 2** *SNTV is resistant to constructive control by deleting candidates.*

**Proof.** Our proof modifies the construction by which Bartholdi et al. [14] show that plurality is resistant to constructive control by deleting candidates via a (polynomial-time many-one) reduction from X3C. Let $(B,\mathcal{S})$ be a given instance of X3C, with $B=\{b_1,\dots,b_m\}$, $m=3r$ for $r\geq 5$ (which can be assumed, without loss of generality, because excluding instances with $r<5$ does not change the complexity of the problem), and a family $\mathcal{S}=\{S_1,\dots,S_n\}$ of three-element subsets of $B$. Let $b_i^1, b_i^2, b_i^3$ denote the three elements of $S_i$. Construct a multiwinner election $E=(C,V,k)$ as follows. Let $C=\{c\}\cup A\cup B\cup D$ be the candidate set, where $c$ is the distinguished candidate and $A=\{a_1,\dots,a_k\}$ and $D=\{d_1,\dots,d_r\}$ are auxiliary candidates. For each $i$, $1\leq i\leq n$, there is one candidate $s_i$ corresponding to the set $S_i\in\mathcal{S}$. The list $V$ of votes is divided into five voter groups as shown in Table 2.

**Table 2** Voter groups in the SNTV-CCDC instance contructed in the proof of Theorem 2

| Voter group | Number of votes | Preference |
|---|---|---|
| $V_1$ | 1 for each $i$, $1\leq i\leq n$ | $s_i\ c\ \cdots$ |
| $V_2$ | 1 for each $i$, $1\leq i\leq n$<br>1 for each $i$, $1\leq i\leq n$<br>1 for each $i$, $1\leq i\leq n$ | $s_i\ b_i^1\ D\ \cdots$<br>$s_i\ b_i^2\ D\ \cdots$<br>$s_i\ b_i^3\ D\ \cdots$ |
| $V_3$ | $r$ | $a_1\ D\ \cdots$ |
| $V_4$ | $r-1$ for each $q$, $2\leq q\leq k$ | $a_q\ D\ \cdots$ |
| $V_5$ | $r-2$ for each $j$, $1\leq j\leq m$ | $b_j\ D\ \cdots$ |

Here, a set $D$ in a preference indicates that the candidates from $D$ occur in an arbitrary order and "$\cdots$" indicates that the remaining candidates from $C$ follow in some arbitrary order. Now, based on those votes, the plurality score of each of the candidates can be calculated:

$$\begin{aligned}\sigma(s_i) &= 4 \text{ for } 1\leq i\leq n,\\ \sigma(a_1) &= r,\\ \sigma(a_q) &= r-1 \text{ for } 2\leq q\leq k,\\ \sigma(b_j) &= r-2 \text{ for } b_j\in B, \text{ and}\\ \sigma(c) &= 0.\end{aligned}$$

Since there are at least $k$ candidates who have a higher plurality score than $c$, $c$ can never be a member of any winning committee and thus is a certain nonwinner of the election.

We now show that $(B,\mathcal{S})$ is a yes-instance of X3C if and only if $c$ can be made a certain winner of the election resulting from $E$ by deleting at most $r$ candidates from it.

Suppose that $(B,\mathcal{S})$ is a yes-instance of X3C, so there is an exact 3-cover $\mathcal{S}'\subseteq\mathcal{S}$ of $B$. Delete the candidates $s_i$ corresponding to the sets $S_i\in\mathcal{S}'$. Since the exact 3-cover has a cardinality of $r$, the plurality score of the candidates will be altered as follows (we denote these scores by $\sigma'$ in the modified election after $|\mathcal{S}'|=r$ candidates $s_i$ were removed from it):

$$\sigma'(s_i) = \sigma(s_i)=4 \text{ for } i,\ 1\leq i\leq n, \text{ with } S_i\notin\mathcal{S}',$$

$$\sigma'(a_1) = \sigma(a_1) = r,$$
$$\sigma'(a_q) = \sigma(a_q) = r-1 \text{ for } 2 \leq q \leq k,$$
$$\sigma'(b_j) = r-1 \text{ for } b_j \in B, \text{ after receiving one vote each in voter group } V_2, \text{ and}$$
$$\sigma'(c) = r.$$

Note that candidates $c$ and $a_1$ have the highest plurality score among the remaining candidates, so they are now members of every winning committee of the election, i.e., certain winners, since $k \geq 2$.

Conversely, assume that there exists a subset $C'$ of no more than $r$ candidates, whose deletion would make $c$ a certain winner of election $E = (C, V, k)$. Based on our construction, $c$ can get votes only from voter group $V_1$, because $c$'s position in the other groups is lower than $r$. That being said, only candidates $s_i$ can be deleted to reach our goal and $c$ can get *no more* than $r$ votes. However, $c$ should also receive *no less* than $r$ votes, since that would tie him with the $a_q$'s, which will result in $c$ being an *uncertain* winner at best (and not a certain winner who is in all winning committees of size $k$). Thus $c$ must receive *exactly* $r$ votes, which can only be achieved by deleting $r$ candidates $s_i$ in group $V_1$.

Moreover, the sets $S_i \in \mathcal{S}'$ which the candidates $s_i$ correspond to must comprise an exact 3-cover for the X3C instance. To show this, for a contradiction assume otherwise. After deleting the candidates and since $|S_i| = 3$ for all $i$, $1 \leq i \leq n$, there would be some $b_j$ who receives two (instead of one) additional votes, giving $b_j$ a plurality score of $r-2+2 = r$. This would tie $b_j$ with $c$ and $a_1$, which leads to $c$ being an uncertain winner only, as $c$ would not be included in all winning committees of size $k = 2$. This means that the initial assumption was wrong and the sets $S_i$ that the deleted candidates $s_i$ correspond to must form an exact 3-cover for the instance $(B, \mathcal{S})$. ❑

**Corollary 1** *SNTV is resistant to constructive control by replacing candidates.*
**Proof.** This follows directly from Theorem 1, as SNTV is IBC according to Lemma 1 and resistant to constructive control by deleting candidates according to Theorem 2. ❑

Now, to show that SNTV is resistant to destructive control by replacing candidates, we start by showing that it is resistant to destructive control by adding candidates.
**Theorem 3** *SNTV is resistant to destructive control by adding candidates.*
**Proof.** For this proof, we modify the construction of Hemaspaandra et al. [15] from their proof that plurality is resistant to destructive control by adding candidates by a reduction from HITTING-SET. Given an instance $(B, \mathcal{S}, r)$ of HITTING-SET, where $B = \{b_1, b_2, \ldots, b_m\}$ is a set, $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ is a family of subsets $S_i$ of $B$, and $r \leq m$ is a positive integer, we construct an instance of SNTV-DCAC as follows. Let $E = (C, V, k)$ be a multiwinner election with registered candidates $C = \{c\} \cup A$, where $A = \{a_1, a_2, \ldots, a_k\}$, unregistered candidates $B$, and the list $V$ of votes that is divided into four voter groups as shown in Table 3.

It is important to note here that in voter groups $V_2$ and $V_4$, the candidates $b_j \in B$ are initially not registered. Hence, the sets $S_1, \ldots, S_n$ are initially empty and candidates $c$ and $a_q$ are ranked first in $V_2$ and $V_4$, respectively. Now, based on the votes in Table 3, the plurality score of each registered candidate can be calculated:

$$\sigma(c) = 2(m-r) + 2n(r+1) + 3k + 2n(r+1) = 2(m-r) + 4n(r+1) + 3k,$$

**Table 3** Voter groups in the SNTV-DCAC instance contructed in the proof of Theorem 3

| Voter group | Number of votes | Preference |
|---|---|---|
| $V_1$ | $2(m-r)+2n(r+1)+3k$ | $c\ a_1\ \cdots$ |
| $V_2$ | $2(r+1)$ for each $i,\ 1\leq i\leq n$ | $S_i\ c\ \cdots$ |
| $V_3$ | $2n(r+1)+3(k+1)$ for each $q, 1\leq q\leq k$ | $a_q\ c\ \cdots$ |
| $V_4$ | 2 for each $j,\ 1\leq j\leq m$, and each $q,\ 1\leq q\leq k$ | $b_j\ a_q\ \cdots$ |

$$\sigma(a_q) \;=\; 2m+2n(r+1)+3(k+1) \text{ for } q,\ 1\leq q\leq k.$$

Based on the above plurality scores, it can be seen that initially candidate $c$ is a certain winner of the election $(C,V,k)$.

We now show that $(B,\mathcal{S},r)$ is a yes-instance of HITTING-SET if and only if $c$ can be made a certain nonwinner of the election resulting from $E$ by adding at most $r$ unregistered candidates.

Indeed, if $B'$ is a hitting set of size $r$ for $\mathcal{S}$, then registering the candidates of $B'$ would cause the plurality scores to change as follows (again, we denote these scores by $\sigma'$ in the modified election):

$$\begin{aligned}
\sigma'(c) &= 2(m-r)+2n(r+1)+3k, \text{ after losing } 2n(r+1) \text{ votes from } V_2,\\
\sigma'(a_q) &= 2(m-|B'|)+2n(r+1)+3(k+1) \text{ for } q,\ 1\leq q\leq k, \text{ and} \qquad (1)\\
\sigma(b_j) &\leq 2k+2n(r+1) \text{ for } b_j\in B', \qquad (2)
\end{aligned}$$

(1) holds since each $a_q$ loses votes whenever some $b_j\in B'$ was added in voter group $V_4$, and (2) holds since $b_j\in B'$ may not necessarily be the first element in the $S_i$ in voter group $V_2$, yet every such $b_j$ has exactly $2k$ votes from group $V_4$.

Note now that $\sigma'(c)<\sigma'(a_q)$ for $q,\ 1\leq q\leq k$, i.e., there are $k$ candidates who have a plurality score higher than $c$, so any winning committee of size $k$ would consist of candidates in $A$ only. This means that $c$ can never be a member of any winning committee. Thus $c$ has been turned into a certain nonwinner of the election $(C\cup B',V,k)$.

Conversely, suppose that $c$ is a certain nonwinner (not a member of any winning committee) of the election $(C\cup B',V,k)$ for any subset $B'\subseteq B$ of unregistered candidates. This means that there are at least $k$ candidates in $(C\cup B')\setminus\{c\}$ who have a higher plurality score than $c$. In $(C\cup B',V,k)$, we have:

$$\begin{aligned}
\sigma'(c) &= 2(m-r)+2n(r+1)+3k+2(r+1)\ell,\\
\sigma'(a_q) &= 2(m-|B'|)+2n(r+1)+3(k+1) \text{ for } q,\ 1\leq q\leq k, \text{ and}\\
\sigma'(b_j) &\leq 2k+2n(r+1) \text{ for } j,\ 1\leq j\leq m,
\end{aligned}$$

where $\ell$ is the number of sets in $\mathcal{S}$ that have not been hit by $B'$. That is, if there exists a set $S_i\in\mathcal{S}$ not hit by $B'$ (i.e., $S_i\cap B'=\emptyset$), $c$ gains $2(r+1)$ additional votes from voter group $V_2$ because $S_i=\emptyset$ in the vote "$S_i\ c\ \cdots$," so it takes the form "$c\ \cdots$." In order to guarantee that $c$ is a certain nonwinner of the election $(C\cup B',V,k)$, $\ell=0$ needs to hold. Recall that there are at least $k$ candidates who have a higher plurality score than $c$. Based on the plurality scores above, notice that these candidates must be all $a_q \in A$, and that $\sigma'(c) < \sigma'(a_q)$ holds only when $\ell = 0$. However, we have

$$\begin{aligned}
2(m-r)+2n(r+1)+3k+2(r+1)\ell &< 2(m-|B'|)+3(k+1)+2n(r+1) \\
2(m-r)+3k+2(r+1)\ell &< 2(m-|B'|)+3k+3 \\
2m-2r+2(r+1)\ell &< 2m-2|B'|+3 \\
2r-2(r+1)\ell &> 2|B'|-3
\end{aligned}$$

and for the smallest $\ell$ such that $\ell \neq 0$, we have $2r-2r-2 > 2|B'|-3$, which is equivalent to $1/2 > |B'|$, a contradiction. This means that the first equation is valid only for the value $\ell = 0$, which implies that $B'$ is a hitting set of size at most $r$.

Summing up, we have shown that $\mathcal{S}$ has a hitting set of size less than or equal to $r$ if and only if destructive control by adding candidates can be executed for the constructed election $(C,V,k)$ with unregistered candidates $B$. ❑

It can now easily be concluded that SNTV is also resistant to destructive control by replacing candidates.

**Corollary 2** *SNTV is resistant to destructive control by replacing candidates.*

**Proof.** This follows directly from Theorem 1, as SNTV is IBC according to Lemma 1 and resistant to destructive control by adding candidates according to Theorem 3. ❑

Having shown the results for replacement control, we now present the complexity results for the last two problems in a shorter way.

**Theorem 4** *SNTV is resistant to both constructive control by adding candidates and destructive control by deleting candidates.*

**Proof.** To show this, we will provide a short sketch of a modification of the proof of NP-hardness of plurality-CCAC due to Bartholdi et al. [14]. To make it work for SNTV elections with a committee size of at least two we need to add $k-1$ candidates to the original election. Also, for each of those candidates, we add $3n$ voters preferring that candidate over everyone else to the original election. Every other voter from the original election has the new candidates ranked last in their vote. Thus those newly added candidates are all certain winners of the election $(C,V,k)$ and they cannot be overtaken by anyone else no matter which candidates get added later. From there on, we can use the same construction and argument that Bartholdi et al. [14] used to show that $c$ can only become the last possible certain winner of this election if the given instance is indeed a hitting set.

In a very similar way, one can modify the proof of NP-hardness for plurality-DCDC due to Hemaspaandra et al. [15] by adding $k-1$ candidates with an insurmountable amount of points to the original election (such that deleting those candidates would only help $c$ win) and thus resulting in a reduction showing that SNTV is also resistant to destructive control by deleting candidates. ❑

We will now turn our attention toward bloc voting and will again start by showing that bloc voting is also IBC.

**Lemma 2** *Bloc voting is insensitive to bottom-ranked candidates.*

**Proof.** Just like SNTV, this is clear from the definition, as for a target committee size of $k$, bloc voting considers the $k$ candidates ranked in the first $k$ positions of each vote. Adding or deleting candidates at the bottom of the profile has no influence on the winning committees, because target committee size values of only $2 \leq k \leq |C|-1$ are considered. Hence, the addition or deletion would take place at the $(|C|+1)$-st position. ❑

**Theorem 5** *Bloc voting is resistant to constructive control by deleting candidates.*
**Proof.** This proof uses a similar approach as that of Theorem 2 to show resistance of bloc voting to constructive control by deleting candidates, again by modifying the approach of Bartholdi et al. [14] showing that plurality is resistant to constructive control by deleting candidates by a reduction from X3C. Let $(B,\mathcal{S})$ be a given instance of X3C, where $B = \{b_1,\dots,b_m\}$ is a set with $m = 3r$ elements (again assuming $r \geq 5$, without loss of generality) and $\mathcal{S} = \{S_1,\dots,S_n\}$ is a family of three-element subsets of $B$. Let $b_i^1, b_i^2, b_i^3$ denote the elements of $S_i$. Construct a multiwinner election $E = (C,V,k)$ as follows. Define the set of candidates by

$$C = \{c\} \cup \left(\bigcup_{i=1}^{n} A_i\right) \cup B \cup D \cup \left(\bigcup_{j=1}^{m} E_j\right) \cup G \cup H \cup \left(\bigcup_{i=1}^{n} \{s_i\}\right),$$

where $c$ is the distinguished candidate, the candidates $s_i$ correspond to the sets $S_i \in \mathcal{S}$, and

$$\begin{aligned} A_i &= \{a_i^1,\dots,a_i^{k-1}\}, \quad 1 \leq i \leq n, \\ D &= \{d_1,\dots,d_k\}, \\ E_j &= \{e_i^1,\dots,e_j^{k-1}\}, \quad 1 \leq j \leq m, \\ G &= \{g_1,\dots,g_{k-1}\}, \text{ and} \\ H &= \{h_1,\dots,h_r\}. \end{aligned}$$

The list $V$ of votes is divided into five voter groups as shown in Table 4, where an arrow sitting on top of a set of candidates indicates that the candidates in this set are ordered by increasing indices.

**Table 4** Voter groups in the bloc-CCDC instance contructed in the proof of Theorem 5

| Voter group | Number of votes | Preference |
|---|---|---|
| $V_1$ | 1 for each $i$, $1 \leq i \leq n$ | $s_i\ \overrightarrow{A_i}\ c\ \cdots$ |
| $V_2$ | $r-1$ | $\overrightarrow{D}\ H\ \cdots$ |
| $V_3$ | 1 | $d_1\ \overrightarrow{G}\ H\ \cdots$ |
| $V_4$ | 1 for each $i$, $1 \leq i \leq n$<br>1 for each $i$, $1 \leq i \leq n$<br>1 for each $i$, $1 \leq i \leq n$ | $s_i\ \overrightarrow{A_i}\ b_i^1\ H\ \cdots$<br>$s_i\ \overrightarrow{A_i}\ b_i^2\ H\ \cdots$<br>$s_i\ \overrightarrow{A_i}\ b_i^3\ H\ \cdots$ |
| $V_5$ | $r-2$ for each $j$, $1 \leq j \leq m$ | $b_j\ \overrightarrow{E_j}\ H\ \cdots$ |

Based on these votes, the $k$-approval scores of the candidates can now be calculated:

$$
\begin{aligned}
\alpha_k(s_i) &= 4 \text{ for } 1 \leq i \leq n, \\
\alpha_k(a_i^t) &= 4 \text{ for } 1 \leq i \leq n \text{ and } 1 \leq t \leq k-1, \\
\alpha_k(d_1) &= r \text{ and } \alpha_k(d_t) = r-1 \text{ for } 2 \leq t \leq k, \\
\alpha_k(b_j) &= r-2 \text{ for } 1 \leq j \leq m, \\
\alpha_k(e_j^t) &= r-2 \text{ for } 1 \leq j \leq m \text{ and } 1 \leq t \leq k-1, \\
\alpha_k(g_t) &= 1 \text{ for } 1 \leq t \leq k-1 \text{ and } \alpha_k(h_t) = 0 \text{ for } 1 \leq t \leq r, \text{ and} \\
\alpha_k(c) &= 0.
\end{aligned}
$$

Note again that there are at least $k$ candidates who have a higher $k$-approval score than the distinguished candidate $c$, so $c$ can never be a member of any winning committee and thus is a certain nonwinner of the election.

Now, we show that $c$ can be made a certain winner by deleting at most $r$ candidates if and only if the X3C instance $(B, \mathcal{S})$ has a solution.

Assume that $(B, \mathcal{S})$ is a yes-instance of X3C, and let $\mathcal{S}' \subseteq \mathcal{S}$ be an exact 3-cover of $B$. Delete the candidates $s_i$ corresponding to the sets $S_i \in \mathcal{S}'$. Since $|\mathcal{S}'| = r$, the $k$-approval scores of the candidates change as follows:

- The distinguished candidate $c$ gets $r$ votes in total from voter group $V_1$.
- Candidates $b_j$ will each get one additional vote from voter group $V_4$ due to the deletion of candidates $s_i$, changing their $k$-approval score to $r-1$ for $1 \leq j \leq m$.
- There are $n-r$ candidates $s_i$ who keep their positions as well as their $k$-approval score of four votes each, since they are not deleted.
- All other candidates also keep their $k$-approval scores.

Note that candidates $c$ and $d_1$ now both have the highest $k$-approval scores. Hence, $c$ has been turned into a certain winner of the election, since $k \geq 2$.

Conversely, assume again that there exists a subset $C' \subseteq C$ of no more than $r$ candidates whose deletion would make candidate $c$ a certain winner of the election $E' = (C \setminus C', V, k)$. Based on the above construction, $c$ would need to get votes from voter group $V_1$, because $c$'s position in the other groups is lower than $r+k$. That being said, only candidates $s_i$ or elements of the corresponding set $A_i$ can be deleted to reach our goal and in that way, $c$ would get *no more* than $r$ votes. This would have no effect on the $k$-approval scores of the $k-1$ candidates in $A_i$ that are not deleted. However, $c$ should also receive *no less* than $r$ votes, since that would tie $c$ with the candidates in $D \setminus \{d_1\}$ and in $B$, which will result in $c$ not being a certain winner anymore but only an uncertain winner or even a certain nonwinner. This is because in the first case, candidate $d_1$ would have $r$ votes, and there would be more than $k-1$ candidates with $r-1$ votes, and in the second case, there would be more than $k$ candidates who each have a $k$-approval score higher than $c$'s. Thus $c$ must receive *exactly* $r$ votes, which can only be achieved by deleting exactly $r$ candidates $s_i$.

Moreover, the sets $S_i \in \mathcal{S}'$ which the candidates $s_i \in C'$ correspond to, must comprise an exact 3-cover for the X3C instance. To see this, for a contradiction assume otherwise. After deleting the candidates and since $|S_i| = 3$ for all $i$, $1 \leq i \leq n$, there would be some $b_j$ who receives two (instead of one) additional votes, giving $b_j$ a $k$-approval score of $r-2+2 = r$.

This would tie $b_j$ with $c$ and $d_1$, which leads to $c$ being an uncertain winner only, as $c$ would not be included in all winning committees of size $k = 2$. This means that the initial assumption was wrong and the sets $S_i$ that the deleted candidates $s_i$ correspond to must form an exact 3-cover for the instance $(B, \mathcal{S})$. ❑

It can now be easily concluded that bloc voting is also resistant to constructive control by replacing candidates.

**Corollary 3** *Bloc voting is resistant to constructive control by replacing candidates.*

**Proof.** This follows immediately from Theorem 1, as bloc voting is IBC according to Lemma 2 and resistant to constructive control by deleting candidates according to Theorem 5. ❑

Now, a similar approach will show that bloc voting is resistant to destructive control by replacing candidates. Again, we start with showing resistance of bloc voting to destructive control by adding candidates.

**Theorem 6** *Bloc voting is resistant to destructive control by adding candidates.*

**Proof.** As in the proof of Theorem 3, we modify the construction of Hemaspaandra et al. [15] from their proof that plurality is resistant to destructive control by adding candidates by a reduction from HITTING-SET. Given an instance $(B, \mathcal{S}, r)$ of HITTING-SET, with a set $B = \{b_1, b_2, \ldots, b_m\}$ and a family $\mathcal{S} = \{S_1, S_2, \ldots, S_n\}$ of subsets $S_i$ of $B$, and $r \leq m$ is a positive integer, construct an instance of our control problem as follows.

Let $E = (C, V, k)$ be a multiwinner election with registered candidates

$$C = \{c\} \cup A \cup G \cup \bigcup_{i=1}^{n} E_i,$$

where $A = \{a_1, \ldots, a_{k-1}\}$, $G = \{g_1, \ldots, g_k\}$, and $E_i = \{e_i^1, \ldots, e_i^{k-1}\}$, with unregistered candidates $B$, and with the list $V$ of votes that is divided into four voter groups as shown in Table 5.

**Table 5** Voter groups in the bloc-DCAC instance contructed in the proof of Theorem 6

| Voter group | Number of votes | Preference |
|---|---|---|
| $V_1$ | $2(m-r) + 2n(r+1) + 3$ | $c \ \overrightarrow{A} \ \cdots$ |
| $V_2$ | $2(r+1)$ for each $i$, $1 \leq i \leq n$ | $\overrightarrow{E_i} \ S_i \ c \ \cdots$ |
| $V_3$ | $2n(r+1) + 4$ | $\overrightarrow{G} \ \cdots$ |
| $V_4$ | $2$ for each $j$, $1 \leq j \leq m$ | $b_j \ \overrightarrow{G} \ \cdots$ |

Note again that in voter groups $V_2$ and $V_4$, the candidates $b_j \in B$ are initially not registered. Hence, the sets $S_1, \ldots, S_n$ are initially empty and the distinguished candidate $c$ is ranked in the $k$-th position in $V_2$ and $g_1$ is ranked first in $V_4$. Based on the votes in Table 5, the $k$-approval scores of each registered candidate can now be calculated:

$$\begin{aligned}
\alpha_k(c) &= 2(m-r)+2n(r+1)+3+2n(r+1) = 2(m-r)+4n(r+1)+3,\\
\alpha_k(a_t) &= 2(m-r)+2n(r+1)+3 \text{ for all } a_t \in A,\\
\alpha_k(e_i^t) &= 2(r+1) \text{ for all } e_i^t \in E_i,\ 1 \le i \le n, \text{ and}\\
\alpha_k(g_t) &= 2m+2n(r+1)+4 \text{ for all } g_t \in G.
\end{aligned}$$

Note that $c$ has the highest $k$-approval score. This makes $c$ a certain winner of the election $E=(C,V,k)$, that is, without registering any candidates from $B$.

Now, we show that $(B,\mathcal{S},r)$ is a yes-instance of HITTING-SET if and only if $c$ can be made a certain nonwinner by adding at most $r$ unregistered candidates.

Indeed, if $B'$ is a hitting set of size $r$ for $\mathcal{S}$, then registering the candidates from $B'$ will alter the $k$-approval scores of the candidates as follows (denoting their $k$-approval scores by $\alpha'_k$ in the modified election):

- Since $B' \cap S_i \neq \emptyset$ for all $i$, $1 \le i \le n$, $c$ is pushed beyond the first $k$ positions of the votes in $V_2$. Thus $c$ loses a total of $2n(r+1)$ votes in $V_2$, so $\alpha'_k(c) = 2(m-r)+2n(r+1)+3$.
- For each registered $b_j \in B'$, some votes in $V_4$ change, and we have $\alpha'_k(g_k) = 2(m-|B'|)+2n(r+1)+4$.
- It may not always be the case that a registered $b_j \in B'$ is ranked in the $k$-th position of the votes in $V_2$. However, every $b_j \in B'$ has at least two votes from voter group $V_4$, so we have $\alpha'_k(b_j) \le 2+2n(r+1)$ for each $b_j \in B'$.
- The candidates in $A$ and $E_i$, $1 \le i \le n$, keep their previous $k$-approval scores.
- The candidates in $G \setminus \{g_k\}$ keep their $k$-approval score of $2m+2n(r+1)+4$ as well.

Since $|B'| \le r$, we have $\alpha'_k(c) < \alpha'_k(g_k) < \alpha'_k(g_1) \le \cdots \le \alpha'_k(g_{k-1})$. Hence, there are $k$ candidates—namely, those in $G$—who have $k$-approval scores higher than $c$'s after the candidates from $B'$ have been added. Thus $c$ cannot be a member of the (unique) winning committee $G$. This means that adding the candidates from $B'$ to the election has turned $c$ into a certain nonwinner.

Conversely, let $B' \subseteq B$ and assume that $c$ is not a member of any winning committee, i.e., a certain nonwinner of the election $(C \cup B', V, k)$. This means there are at least $k$ candidates who definitely have higher $k$-approval scores than $c$. The candidates with higher $k$-approval scores than $c$ have to be the candidates in $G$, since no candidate from a group other than $G$ is close enough in points to $c$.

Now recall from the construction above that after registering the candidates from $B'$, we have the following $k$-approval scores in the modified election (again denoting them by $\alpha'_k$ and letting $\ell$ be the number of sets in $\mathcal{S}$ that have not been hit by $B'$):

$$\begin{aligned}
\alpha'_k(c) &= 2(m-r)+2n(r+1)+3+2(r+1)\ell,\\
\alpha'_k(g_k) &= 2(m-|B'|)+2n(r+1)+4,\\
\alpha'_k(g_t) &= 2m+2n(r+1)+4 \text{ for } t,\ 1 \le t \le k-1,\\
\alpha'_k(b_j) &\le 2+2n(r+1) \text{ for } b_j \in B',\\
\alpha'_k(a_t) &= 2(m-r)+2n(r+1)+3 \text{ for } t,\ 1 \le t \le k-1, \text{ and}\\
\alpha'_k(e_i^t) &= 2(r+1) \text{ for } e_i^t \in E_i,\ 1 \le i \le n.
\end{aligned}$$

Based on the above scores, if the $k$ candidates in the winning committee are the candidates in $G$, then $\alpha'_k(c) < \alpha'_k(g_t)$ for all $g_t \in G$. However, in order for $\alpha'_k(c) < \alpha'_k(g_k)$ to hold, $\ell = 0$ must hold as well. We have the following:

$$2(m-r)+2n(r+1)+3+2(r+1)\ell < 2(m-|B'|)+2n(r+1)+4$$
$$2m-2r+2(r+1)\ell < 2m-2|B'|+1$$
$$2r-2(r+1)\ell > 2|B'|-1$$

and for the smallest $\ell$ such that $\ell \neq 0$, we have $2r-2r-2 > 2|B'|-1$, which is eqivalent to $-1/2 > |B'|$, a contradiction. This means that $\alpha'_k(c) < \alpha'_k(g_k)$ can only hold for the value $\ell = 0$. Hence, all sets in $\mathcal{S}$ are hit by $B'$, so $B' \subseteq B$ is a hitting set of size less than or equal to $r$ for $\mathcal{S}$. ❑

Finally, we can easily conclude that bloc voting is also resistant to destructive control by replacing candidates.

**Corollary 4** *Bloc voting is resistant to destructive control by replacing candidates.*

**Proof.** This follows directly from Theorem 1, because bloc voting is IBC according to Lemma 2 and resistant to destructive control by adding candidates according to Theorem 6. ❑

Again, we will now present the complexity results for the last two problems in a shorter way.

**Theorem 7** *Bloc voting is resistant to both constructive control by adding candidates and destructive control by deleting candidates.*

**Proof.** To show this we will once again provide a short sketch of a modification of the proof of NP-hardness of plurality-CCAC due to Bartholdi et al. [14]. To make it work for bloc voting with a committee size of at least two, we need to add $k-1$ candidates to the original election. All voters from the original election have those new voters ranked on the first $k-1$ positions of their votes. Thus those newly added candidates are all certain winners of the election $(C,V,k)$ and they cannot be overtaken by anyone else no matter which candidates get added later. From there on we can use the same construction and argument that Bartholdi et al. [14] used to show that $c$ can only become the last possible certain winner of this election if the given instance is indeed a hitting set.

In a very similar way, one can modify the proof of NP-hardness for plurality-DCDC due to Hemaspaandra et al. [15] by adding $k-1$ candidates who are ranked first by every voter to the original election (such that deleting those candidates would only help $c$ win) and thus resulting in a reduction showing that bloc voting is also resistant to destructive control by deleting candidates. ❑

# 4 Control by Replacing Voters

We now turn toward control by replacing voters. We will again begin by presenting our results for SNTV. Meir et al. [1] have already provided an algorithm that solves the problem for adding voters in polynomial time. This algorithm also works for our definition of the problem since they specifically designed it to work for multiple winners—unlike as for the candidate control problems where they simply adapted the proofs from the single-winner variants of the voting rules. Also, we can easily see that we can get an equivalent to our model of the problem: In the constructive case, we modify the utilities in their model such that the distinguished candidate has a utility of 1 and each other candidate has a utility of 0 and the target utility is 1; while in the destructive case, the distinguished candidate has a utility of 0 and all other candidates have a utility of 1 and the target utility is $k$. Further, they also provided proofs showing that the cases for deleting voters are also in P. Therefore, we will now only focus on the problem of replacing voters. In fact, it is not trivial to adapt their algorithm to also work for the problems of replacing voters, and thus we will present new algorithms solving these problems in polynomial time. We start with the constructive case in Theorem 8; the destructive case in Theorem 9 then works very similarly.

**Theorem 8** *SNTV is vulnerable to constructive control by replacing voters.*

**Proof.** We describe a polynomial-time algorithm that solves this problem. Let $(C,V,U,k,c,r)$ be a given input of SNTV-CCRV. First, compute the maximum score $\sigma_{\max}(c)$ that our distinguished candidate $c$ can be pushed to by adding at most $r$ voters from $U$. We can do this by adding the number of voters in $U$ that rank $c$ first to its score in the original election (but at most $r$). Thus

$$\sigma_{\max}(c) = \sigma_{(C,V)}(c) + \min(r, \sigma_{(C,U)}(c)).$$

Let $A = \min(r, \sigma_{(C,U)}(c))$ be the number of voters in $U$ ranking $c$ first that we can add. In order for $c$ to become a certain winner of the election, $c$ needs to have more points than at least $|C| - k$ other candidates. If there are fewer than that number of candidates in the original election with at least $\sigma_{\max}(c)$ points, we can make $c$ a certain winner by replacing $A$ voters not ranking $c$ first in $V$ with $A$ voters from $U$ ranking $c$ first, and we are done.

Otherwise, for each candidate $f \in C \setminus \{c\}$ with

$$\sigma_{(C,V)}(f) \geq \sigma_{\max}(c), \tag{3}$$

we now check whether there are at least

$$\mathrm{diff}_f = \sigma_{(C,V)}(f) + 1 - \sigma_{\max}(c) \tag{4}$$

voters in $U$ that do not rank $f$ first. Let $F$ denote the set of candidates fulfilling both these conditions, (3) and (4). Also, define

$$g = 1 + |\{f \in C \setminus \{c\} \mid \sigma_{(C,V)}(f) \geq \sigma_{\max}(c)\}| - k.$$

That is, $g$ is the number of candidates whose score we need to reduce below $\sigma_{\max}(c)$ in order to make $c$ a certain winner. Sort the candidates in $F$ in ascending order such that $\mathrm{diff}_{f_1} \leq \mathrm{diff}_{f_2} \leq \cdots \leq \mathrm{diff}_{f_{|F|}}$. Now, control is possible only if $\sum_{i=1}^{g} \mathrm{diff}_{f_i} \leq r$; so if not, the algorithm rejects right here. If this sum is less than $A$, we delete $A - \sum_{i=1}^{g} \mathrm{diff}_{f_i}$ voters not ranking $c$ first, and the algorithm accepts. If the sum is greater than $A$, we need to add more voters not ranking $c$ first. To this end, we first add voters that rank the candidates $f_j \in F$ first, $j > g$. If enough of those voters exist, the algorithm accepts. Else we add voters ranking the candidates with the lowest score first. If that makes $c$ a certain winner of the election, the algorithm accepts; otherwise, $c$ cannot be made a certain winner. ❑

**Theorem 9** *SNTV is vulnerable to destructive control by replacing voters.*

**Proof.** A very similar algorithm can be used for the destructive case. Let $(C,V,U,k,c,r)$ be a given input of SNTV-DCRV. First, compute the minimum score $\sigma_{\min}(c)$ our distinguished candidate $c$ can be pushed to by deleting at most $r$ voters from $V$. We can do this by counting the number of voters in $U$ that do not rank $c$ first (but at most $r$) and subtract it from $c$'s score in the original election. Thus

$$\sigma_{\min}(c) = \sigma_{(C,V)}(c) - \min(r, |U| - \sigma_{(C,U)}(c), \sigma_{(C,V)}(c)).$$

Also, let $D = \min(r, |U| - \sigma_{(C,U)}(c), \sigma_{(C,V)}(c))$ be the number of voters ranking $c$ first that we can delete from $V$.

In order for $c$ to become a certain nonwinner of the election, $c$ needs to have fewer points than at least $k$ other candidates. If there are at least $k$ candidates in the original election with more than $\sigma_{\min}(c)$ points, we are done and can make $c$ a certain nonwinner by replacing $D$ voters ranking $c$ first in $V$ with $D$ voters from $U$ ranking anyone else first.

Otherwise, for each candidate $f \in C \setminus \{c\}$ with

$$\sigma_{(C,V)}(f) \leq \sigma_{\min}(c), \tag{5}$$

we now check whether there are at least

$$\text{diff}_f = \sigma_{\min}(c) - \sigma_{(C,V)}(f) + 1 \tag{6}$$

voters in $U$ that do rank $f$ first. Let $F$ denote the set of candidates fulfilling both these conditions, (5) and (6). Also, define

$$g = k - |\{f \in C \setminus \{c\} \mid \sigma_{(C,V)}(f) > \sigma_{\min}(c)\}|.$$

That is, $g$ is the number of candidates whose score we need to increase above $\sigma_{\min}(c)$ in order to make $c$ a certain nonwinner. Sort the candidates in $F$ in ascending order such that $\text{diff}_{f_1} \leq \text{diff}_{f_2} \leq \cdots \leq \text{diff}_{f_{|F|}}$. Now, control is possible only if $\sum_{i=1}^{g} \text{diff}_{f_i} \leq r$; so if not, the algorithm rejects right here. If this sum is less than $D$, we add $D - \sum_{i=1}^{g} \text{diff}_{f_i}$ voters from $U$ not ranking $c$ first, and the algorithm accepts. If the sum is greater than $D$, we need to delete more voters not ranking $c$ first. To this end, we first delete voters ranking the candidates with the lowest plurality scores. If that makes $c$ a certain nonwinner of the election, the algorithm accepts; otherwise, $c$ cannot be made a certain nonwinner. ❑

Next, we turn our attention back toward bloc voting. Again, Meir et al. [1] have already shown NP-hardness for all cases of adding and deleting voters. Since for these problems their proof is a full reduction and not simply a reference to the single-winner case of $k$-approval voting, it is easy to see that their result will also hold for committee size $k > 1$. Therefore, we only need to consider control by replacing voters.

**Theorem 10** *Bloc voting is resistant to constructive control by replacing voters.*

**Proof.** We show NP-hardness of bloc-CCRV by reducing from bloc-CCDV. Let $(C', V', k, c, r)$ be an instance of bloc-CCDV. Without loss of generality, we may assume that $\sigma(c)_{(C',V')} \geq 2$. From this instance, we construct our bloc-CCRV instance $(C,V,U,k,c,r)$ as follows. Define the candidate set $C = C' \cup D$ with $D = \{d_1, \dots, d_{k \cdot r}\}$. The votes in $V$ result from those in $V'$ by adding the candidates from $D$ in any order at the bottom of each vote. Therefore, no candidate from $D$ scores any points in $(C,V,k)$. Additionally, $U$ contains $r$ votes that each rank a $k$-element subset of candidates from $D$ first such that these $r$ subsets of $D$ are pairwise disjoint. Therefore, no candidate from $C'$ would benefit from any of the votes in $U$ being added to the election.

We claim that $(C,V,U,k,c,r)$ is a yes-instance of bloc-CCRV if and only if $(C',V',k,c,r)$ is a yes-instance of bloc-CCDV.

From right to left, suppose it is possible to make $c$ a certain winner in $(C',V',k)$ by deleting at most $r$ votes from $V'$. Then we can also make $c$ a certain winner by deleting the corresponding votes from $V$ and adding the same number of votes from $U$. Since the newly added votes give the candidates from $D$ only up to one point each (which is less than the score of $c$), $c$ has been turned into a certain winner of the election by replacing at most $r$ votes.

From left to right, assume that $(C,V,U,k,c,r)$ is a yes-instance of bloc-CCRV. Then, to make $c$ a certain winner in $(C',V',k)$, we can simply delete the votes from $V'$ that correspond to those votes we have replaced in $V$. This works because, no matter which votes from $U$ were added to the election to replace votes deleted from $V$, the candidates that received points from those votes are all still certain nonwinners and thus not adding them does not prevent $c$ from becoming a certain winner of the election. ❑

**Theorem 11** *Bloc voting is resistant to destructive control by replacing voters.*

**Proof.** We show NP-hardness of bloc-DCRV in a very similar way by reducing from bloc-DCDV. Let $(C',V',k,c,r)$ be an instance of bloc-DCDV. Without loss of generality, we may assume that $\sigma(c)_{(C',V')} \geq r+2$. From this instance, we construct our bloc-DCRV instance $(C,V,U,k,c,r)$ as follows. Define the set $C = C' \cup D$ with $D = \{d_1, \dots, d_{k \cdot r}\}$. The votes in $V$ result from those in $V'$ by adding the candidates from $D$ in any order at the bottom of each vote. Therefore, no candidate from $D$ scores any points in $(C,V,k)$. Additionally, $U$ contains $r$ votes that each rank a $k$-element subset of candidates from $D$ first such that these $r$ subsets of $D$ are pairwise disjoint. Therefore, no candidate from $C'$ would benefit from any of the votes in $U$ being added to the election.

We claim that $(C,V,U,k,c,r)$ is a yes-instance of bloc-DCRV if and only if $(C',V',k,c,r)$ is a yes-instance of bloc-DCDV.

From right to left, suppose it is possible to make $c$ a certain nonwinner in $(C',V',k)$ by deleting at most $r$ votes from $V'$. Then we can also make $c$ a certain nonwinner by deleting the corresponding votes from $V$ and adding the same number of votes from $U$. Since the newly added votes do not add any points to the score of $c$, $c$ will still be a certain nonwinner.

From left to right, assume that $(C,V,U,k,c,r)$ is a yes-instance of bloc-DCRV. Then, to make $c$ a certain nonwinner in $(C',V',k)$, we can simply delete the votes from $V'$ that correspond to those votes we have replaced in $V$. This works because, no matter which votes from $U$ were added to the election to replace votes deleted from $V$, the candidates that received points from those votes now all have a maximum score of 1, since no candidate appears more than once in the first $k$ positions of the votes in $U$, and even if we delete the maximum of $r$ votes that rank $c$ among their first $k$ positions, the score of $c$ is still at least 2. So adding these points to the candidates from $D$ was not what made $c$ become a certain nonwinner. ❑

## 5 Conclusions and Future Work

We have studied the complexity of control of two of the most popular multiwinner voting rules: single nontransferable vote and bloc voting. We have shown that these two rules are resistant to constructive and destructive control by adding, by deleting, and by replacing candidates. This complements previous results by Meir et al. [1] who use a somewhat different framework to model control via utility functions. We have also extended their results on voter control by providing new results for both constructive and destructive control by replacing voters for both multiwinner voting rules mentioned above. Note further that we excluded the committee size $k = 1$ (i.e., the single-winner case), so our results do not immediately follow from the corresponding single-winner results. We have shown them by using the notion of IBC and applying a general result of Loreggia [34] by appropriately modifying previous reductions of Bartholdi et al. [14] and Hemaspaandra et al. [15] adapted to our setting.

One of the key aims of this paper is drawing more attention to the fact that the complexity of replacement control has not been much explored yet for multiwinner elections. Arguably, multiwinner voting is similarly important as single-winner voting in practice. We hope that our work may encourage further investigation of this interesting and significant topic.

For future work, we propose to study control by replacing candidates or voters (and further control types, e.g., control by partitioning them) also for other prominent multiwinner voting rules such as the Chamberlin–Courant rule [43].

**Acknowledgments.** We thank the anonymous ADT'24 reviewers for helpful comments. We gratefully acknowledge that this work was supported in part by DFG grant RO 1202/21-2 (project number 438204498).

## Declarations

**Funding:** This work was supported in part by Deutsche Forschungsgemeinschaft under grant RO 1202/21-2 (project number 438204498).

**Non-financial interests:** Author Jörg Rothe currently is or has been on the following editorial boards of scientific journals:

- *Annals of Mathematics and Artificial Intelligence* (AMAI), Associate Editor, since 01/2020,
- *Journal of Artificial Intelligence Research* (JAIR), Associate Editor, 09/2017–08/2023,
- *Mathematical Logic Quarterly* (MLQ – Wiley), Editorial Board, 01/2008–12/2019, and
- *Journal of Universal Computer Science* (J.UCS), Editorial Board, since 01/2005.

**Availability of data and materials:** Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

**Conflict of Interest:** The authors declare that they have no conflict of interest.

## References

[1] Meir, R., Procaccia, A., Rosenschein, J., Zohar, A.: Complexity of strategic behavior in multi-winner elections. Journal of Artificial Intelligence Research **33**, 149–178 (2008)

[2] Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.): Handbook of Computational Social Choice. Cambridge University Press, Cambridge, UK (2016)

[3] Rothe, J. (ed.): Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, 2nd edn. Classroom Companion: Economics. Springer, Cham, Switzerland (2024)

[4] Baumeister, D., Faliszewski, P., Rothe, J., Skowron, P.: Multiwinner voting. In: Rothe, J. (ed.) Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, 2nd edn. Classroom Companion: Economics, pp. 403–465. Springer, Cham, Switzerland (2024). Chap. 6

[5] Faliszewski, P., Skowron, P., Slinko, A., Talmon, N.: Multiwinner voting: A new challenge for social choice theory. In: Endriss, U. (ed.) Trends in Computational Social Choice, pp. 27–47. AI Access Foundation, aiaccess.org (2017). Chap. 2

[6] Ephrati, E., Rosenschein, J.: The Clarke Tax as a consensus mechanism among automated agents. In: Proceedings of the 9th National Conference on Artificial Intelligence, pp. 173–178. AAAI Press, aaai.org (1991)

[7] Ephrati, E., Rosenschein, J.: A heuristic technique for multi-agent planning. Annals of Mathematics and Artificial Intelligence **20**(1–4), 13–67 (1997)

[8] Pennock, D., Horvitz, E., Giles, C.: Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In: Proceedings of the 17th National Conference on Artificial Intelligence, pp. 729–734. AAAI Press, aaai.org (2000)

[9] Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: Proceedings of the 10th International World Wide Web Conference, pp. 613–622. ACM Press, dl.acm.org (2001)

[10] Fagin, R., Kumar, R., Sivakumar, D.: Efficient similarity search and classification via rank aggregation. In: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, pp. 301–312. ACM Press, dl.acm.org (2003). Corrigendum: dl.acm.org/doi/abs/10.1145/1376616.1376778

[11] Conitzer, V., Walsh, T.: Barriers to manipulation in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.) Handbook of Computational Social Choice, pp. 127–145. Cambridge University Press, Cambridge, UK (2016). Chap. 6

[12] Faliszewski, P., Rothe, J.: Control and bribery in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.) Handbook of Computational Social Choice, pp. 146–168. Cambridge University Press, Cambridge, UK (2016). Chap. 7

[13] Baumeister, D., Rothe, J.: Preference aggregation by voting. In: Rothe, J. (ed.) Economics and Computation. An Introduction to Algorithmic Game Theory, Computational

Social Choice, and Fair Division, 2nd edn. Classroom Companion: Economics, pp. 233–367. Springer, Cham, Switzerland (2024). Chap. 4

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

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

[16] Brams, S.: Approval voting in multicandidate elections. Policy Studies Journal **9**(1), 102–108 (1980)

[17] Brams, S., Fishburn, P.: Approval voting, Condorcet's principle, and runoff elections. Public Choice **36**(1), 89–114 (1981)

[18] Brams, S., Fishburn, P.: Approval Voting. Birkhäuser, Boston, MA, USA (1983)

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

[20] Elkind, E., Faliszewski, P., Slinko, A.: Cloning in elections: Finding the possible winners. Journal of Artificial Intelligence Research **42**, 529–573 (2011)

[21] Neveling, M., Rothe, J.: Control complexity in Borda elections: Solving all open cases of offline control and some cases of online control. Artificial Intelligence **298**, 103508 (2021)

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

[23] Maushagen, C., Rothe, J.: The last voting rule is home: Complexity of control by partition of candidates or voters in maximin elections. In: Proceedings of the 24th European Conference on Artificial Intelligence. Frontiers in Artificial Intelligence and Applications, vol. 325, pp. 163–170. IOS Press, Amsterdam, The Netherlands (2020)

[24] Maushagen, C., Rothe, J.: 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, pp. 277–285. IOS Press, Amsterdam, The Netherlands (2016)

[25] Maushagen, C., Rothe, J.: Complexity of control by partitioning veto elections and of control by adding candidates to plurality elections. Annals of Mathematics and Artificial Intelligence **82**(4), 219–244 (2018)

[26] Lin, A.: Solving hard problems in election systems. PhD thesis, Rochester Institute of Technology, Rochester, NY, USA (March 2012)

[27] Lin, A.: The complexity of manipulating $k$-approval elections. In: Proceedings of the 3rd International Conference on Agents and Artificial Intelligence, pp. 212–218. SciTePress, Setúbal, Portugal (2011)

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

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

[30] Erdélyi, G., Fellows, M., Rothe, J., Schend, L.: Control complexity in Bucklin and fallback voting: An experimental analysis. Journal of Computer and System Sciences **81**(4), 661–670 (2015)

[31] Menton, C., Singh, P.: Control complexity of Schulze voting. In: Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 286–292. AAAI Press/IJCAI, ijcai.org (2013)

[32] Hemaspaandra, L., Lavaee, R., Menton, C.: Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. Annals of Mathematics and Artificial Intelligence **77**(3–4), 191–223 (2016)

[33] Maushagen, C., Niclaus, D., Nüsken, P., Rothe, J., Seeger, T.: Toward completing the picture of control in Schulze and ranked pairs elections. In: Proceedings of the 33rd International Joint Conference on Artificial Intelligence, pp. 2940–2948. IJCAI, ijcai.org (2024)

[34] Loreggia, A.: Iterative voting, control and sentiment analysis. PhD thesis, University of Padova, Padova, Italy (2016)

[35] Loreggia, A., Narodytska, N., Rossi, F., Venable, B., Walsh, T.: Controlling elections by replacing candidates or votes (extended abstract). In: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, pp. 1737–1738. IFAAMAS, www.ifaamas.org (2015)

[36] Erdélyi, G., Neveling, M., Reger, C., Rothe, J., Yang, Y., Zorn, R.: Towards completing the puzzle: Complexity of control by replacing, adding, and deleting candidates or voters. Journal of Autonomous Agents and Multi-Agent Systems **35**(2), 41–14148 (2021)

[37] Yang, Y.: Complexity of manipulating and controlling approval-based multiwinner voting. In: Proceedings of the 28th International Joint Conference on Artificial Intelligence, pp. 637–643. IJCAI, ijcai.org (2019)

[38] Lang, J., Maudet, N., Polukarov, M.: New results on equilibria in strategic candidacy. In: Proceedings of the 6th International Symposium on Algorithmic Game Theory. Lecture

Notes in Computer Science, vol. 8146, pp. 13–25. Springer, Heidelberg and Berlin, Germany (2013)

[39] Karh Bet, G., Rothe, J., Zorn, R.: Complexity of candidate control for single nontransferable vote and bloc voting. In: Proceedings of the 8th International Conference on Algorithmic Decision Theory. Lecture Notes in Artificial Intelligence, vol. 15248, pp. 3–17. Springer, Heidelberg and Berlin, Germany (2024)

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

[41] Papadimitriou, C.: Computational Complexity, 2nd edn. Addison-Wesley, Reading, MA, USA (1995)

[42] Rothe, J.: Complexity Theory and Cryptology. An Introduction to Cryptocomplexity. EATCS Texts in Theoretical Computer Science. Springer, Heidelberg and Berlin, Germany (2005)

[43] Chamberlin, J., Courant, P.: Representative deliberations and representative decisions: Proportional representation and the Borda rule. The American Political Science Review **77**(3), 718–733 (1983)

# Chapter 5

# The Complexity of Cloning Candidates in Multiwinner Elections

## 5.1 Summary

This work initiated the study of cloning in multiwinner elections. To this end, we adapted the model of cloning by Elkind, Faliszewski, and Slinko [9] from single-winner to multiwinner elections. More details on our model of cloning in multiwinner elections can be found towards the end of Chapter 2.

For multiwinner voting rules, we established new results for STV, SNTV, Bloc voting, $k$-Borda, $t$-Approval-CC, and Borda-CC. We considered all possible control problems stemming from our model for both necessary cloning and possible cloning, as well as for all three presented cost models: general-cost, unit-cost and zero-cost. We provided proofs for the computational complexity for all these control problems for STV, SNTV, Bloc voting, and $k$-Borda. To do this, we provided some polynomial-time algorithms to show membership in P, and used polynomial-time many-one reductions to show NP-hardness. For the remaining two voting rules, we showed some results regarding their parameterized complexity.

## 5.2 Personal Contribution

I had no parts in finding any of the results of the original shorter version of this paper by M. Neveling and Jörg Rothe, nor in the writing thereof.

The extension of the original conference version to the journal version, and the writing of it, was done by me, with finalization and polishing by Jörg Rothe.

I found and corrected a mistake in a proof sketch from the appendix of the original shorter work for Bloc-POSSIBLE-CLONING. This resulted in two new technical results by me, stated in Theorems 4.4 and 4.5 of the journal version.

## 5.3 Publication

M. Neveling, J. Rothe, and R. Zorn. "The Complexity of Cloning Candidates in Multiwinner Elections". In: *Journal of Autonomous Agents and Multi-Agent Systems* (Submitted)

A preliminary version of this work without my involvement has been submitted to and accepted at the *International Conference on Autonomous Agents and Multi-Agent Systems 2020*:

M. Neveling, and J. Rothe. "The Complexity of Cloning Candidates in Multiwinner Elections". In: *International Conference on Autonomous Agents and Multi-Agent Systems 2020 (AAMAS 2020)*.

# The Complexity of Cloning Candidates in Multiwinner Elections

Marc Neveling[1], Jörg Rothe[1*], Roman Zorn[1]

[1]Institut für Informatik, MNF, Heinrich-Heine-Universität Düsseldorf, Universitätsstr. 1, Düsseldorf, 40225, Germany.

*Corresponding author(s). E-mail(s): rothe@hhu.de;
Contributing authors: marc.neveling@hhu.de; roman.zorn@hhu.de;

**Abstract**

We initiate the study of cloning in multiwinner elections, focusing on single-transferable vote (STV), single-nontransferable vote (SNTV), bloc voting, $k$-Borda, $t$-approval-CC, and Borda-CC. Transferring the model of cloning due to Elkind et al. [1] from single-winner to multiwinner elections, we consider decision problems describing possible and necessary cloning in the zero-cost, the unit-cost, and the general-cost model and study their computational complexity. In this model, a manipulator can add clones of candidates to an election, where clones are so similar to the original candidate that each vote simply ranks all of them as a block in their preference order. The manipulator is assumed to be restricted by a varying cost per clone and a budget. We show that, depending on the multiwinner voting rule and on the cost model chosen, some of these cloning problems are in P, some are NP-hard, and some of the latter (for which, in fact, already winner determination is NP-hard) are fixed-parameter tractable.

**Keywords:** Computational social choice, Multiwinner elections, Cloning, STV, SNTV, bloc voting, $k$-Borda, $t$-approval-CC, and Borda-CC

## 1 Introduction

A common thread in computational social choice—see, e.g., the books edited by Brandt et al. [2] and Rothe [3]—is to study how the outcome of elections can be tampered with and how resistant voting rules are against such attempts in terms of computational complexity. The most thoroughly studied types of attack are *manipulation* (see, e.g., the book chapters by Conitzer and Walsh [4] and Baumeister and Rothe [5, Section 4.3.3]), *electoral control* (see, e.g., the book chapters by Faliszewski and Rothe [6] and Baumeister and Rothe [5, Section

4.3.4]), and *bribery* (see, e.g., the book chapters by Faliszewski and Rothe [6] and Baumeister and Rothe [5, Section 4.3.5]). On the other hand, relatively few papers have studied attacks by cloning candidates (see the related work below), and they are typically concerned with cloning in single-winner voting rules. We initiate the study of cloning in *multiwinner* elections, where the goal is not to elect a winner but to elect a *winning committee* of a certain size (see, e.g., the book chapters by Baumeister et al. [7] and Faliszewski et al. [8]). Multiwinner elections have various applications ranging from parliament elections over short-listing possible employees to selecting items to offer to a group of people (see the work of Lu and Boutilier [9], Elkind et al. [10], and Skowron et al. [11] for more detailed descriptions of the mentioned settings).

In each of those settings, cloning candidates might be beneficial for some given candidate to be voted into a resulting winning committee. For instance, in a parliament election the candidates of a party may look like clones of each other to ignorant voters, so the campaign manager of this party might be inclined to nominate only a strategically chosen number of candidates to represent the party. Another application of cloning in multiwinner elections are movie recommender systems [12] in which a set of movies is recommended depending on the users' preferences: To influence the election result by spreading out and diminishing the support of a particular disliked movie, one might add to the election additional, very similar movies (e.g., other movies of the same genre or with a similar cast or by the same director).

In social choice theory, Tideman [13] introduced the notion of cloning and studied the *independence of clones* property for various voting rules. In particular, he showed that the single-winner variant of single-transferable vote (STV) is independent of clones. In a follow-up paper, Zavist and Tideman [14] studied independence of clones for the ranked pairs rule and presented a variant of ranked pairs that is even "completely independent of clones." Schulze voting is another widely used voting rule that is independent of clones [15]. In anonymous settings, such as the internet, voters may be tempted and able to cast their vote twice (or more often). This was the motivation for Conitzer [16] to introduce *false-name manipulation* as some kind of "cloning of voters" instead of candidates.[1] More recently, Ayadi et al. [21] studied the independence of clones property for the single-winner variant of STV with top-truncated votes.

The paper by far most closely related to our work is due to Elkind et al. [1] (see also their follow-up paper [22]). They were the first to study how resistant single-winner voting rules are against cloning in terms of computational complexity. Adapting their model of cloning to multiwinner (rather than single-winner) elections, we consider decision problems describing *possible* cloning (where we ask whether a given candidate can become a member of a winning committee in at least one cloned multiwinner election, i.e., for at least one ordering of the clones) and *necessary* cloning (where we ask the same question for all cloned multiwinner elections, i.e., for all orderings of the clones), where the cloning costs are specified according to three cost models: zero cost, unit cost, and general cost. We study these problems in terms of their computational complexity and show that, depending on the multiwinner voting rule and on the cost model chosen, some of these cloning problems are in P, some are NP-hard, and some of the latter (for which, in fact, already winner determination is NP-hard) are in FPT, i.e., they are fixed-parameter tractable.

[1] False-name manipulation [17, 18] has also been studied in cooperative game theory and the related property of duplication monotonicity [19, 20] has also been studied in fair division.

*Organization.* In Section 2, we present some background from social choice theory and multiwinner voting rules. In Section 3, we describe our model and define the problems to be studied in terms of their complexity. Section 4 contains our results for a variety of well-known multiwinner voting rules, and Section 5 presents our conclusions and some open problems.

## 2 Preliminaries

A *multiwinner election* $E = (C,V,k)$ is defined by a set $C = \{c_1,\dots,c_m\}$ of candidates, a list $V = (v_1,\dots,v_n)$ of votes over $C$, and a desired committee size $k$. Votes are assumed to be (strict) linear orders over the candidates and we write them each as a sequence of candidates, with the voter's preference strictly decreasing from left to right, so the leftmost (rightmost) candidate in a vote is most (least) preferred by this voter. For instance, if $C = \{a,b,c,d\}$, a vote $b\ a\ c\ d$ means that $b$ is preferred to $a$, $a$ to $c$, and $c$ to $d$.

Given a multiwinner election $(C,V,k)$, a *multiwinner voting rule* returns a nonempty family of size-$k$ subsets of $C$, referred to as the *winning committees*. Given $(C,V,k)$ and a fixed $t \geq 1$, the *$t$-approval score of a candidate $c \in C$* is the number of votes in which $c$ is ranked in the first $t$ positions, and $c$'s Borda score is the total number of points $c$ scores in all votes of $V$, where $c$ is rewarded with $m-i$ points whenever $c$ is ranked in the $i$-th position of a vote for $m$ candidates. Note that 1-approval is also known as *plurality* and the 1-approval score as the *plurality score of a candidate $c$*. We consider the following multiwinner voting rules (each with $n$ voters and committee size $k$):[2]

**Single transferable vote (STV):** Let $q = \lfloor n/(k+1) \rfloor + 1$ be the quota. Iteratively, if a candidate $c$ is ranked first in at least $q$ votes, add $c$ to the winning committee and remove both $c$ and $q$ votes that rank $c$ first from the multiwinner election, or else eliminate a candidate from the multiwinner election that is ranked first in the smallest number of votes. The iteration halts as soon as $k$ candidates have been selected. Ties between votes (i.e., when a candidate is ranked first in more than $q$ votes but only $q$ of those votes will be removed) are broken by an arbitrarily chosen but fixed order.[3]
**Single nontransferable vote (SNTV):** Choose $k$ candidates with highest 1-approval score.
**Bloc voting:** Choose $k$ candidates with highest $k$-approval score.
***k*-Borda:** Choose $k$ candidates with highest Borda score.
***t*-approval-CC:** A voter $v$ approves a committee if $v$ ranks a committee member in the first $t$ positions and disapproves it otherwise. The committee(s) with the most approvals from the voters win(s). Here, CC stands for "Chamberlin–Courant" [24].
**Borda-CC:** Works similarly as $t$-approval-CC except that the voters assign to each committee the Borda score of its highest ranked member in their preferences.

Note that $t$-approval-CC and Borda-CC have an NP-hard winner determination problem [9, 25], though they are in FPT if parameterized by the number of candidates or voters [26].

For a multiwinner election $(C,V,k)$ and candidates $c,d \in C$, let $score_{(C,V,k)}(c)$ denote the number of points $c$ scores (according to either $t$-approval or Borda, which will always be clear

[2] To break ties between candidates in these rules, we will use a predefined lexicographic tie-breaking order.
[3] We cannot use "parallel-universe tie-breaking" [23] for STV since winner determination would then already be NP-hard.

from the context), and let

$$dist_{(C,V,k)}(c,d) = score_{(C,V,k)}(c) - score_{(C,V,k)}(d)$$

denote the difference between the scores of $c$ and $d$ in $(C,V,k)$. We sometimes omit the subscript $(C,V,k)$ if it is clear from the context. If $S \subseteq C$ is a subset of the candidates, $\overrightarrow{S}$ in a vote denotes a ranking of these candidates in an arbitrary but fixed order and $\overleftarrow{S}$ denotes this ranking in reverse order. For example, for $C = \{a,b,c,d\}$ and $S = \{a,d\}$ and assuming the lexicographic order of candidates, $c\ \overleftarrow{S}\ b$ denotes the vote $c\ a\ d\ b$ and the vote $c\ \overleftarrow{S}\ b$ denotes $c\ d\ a\ b$. Occasionally, we omit the candidates in a vote whose order does not matter for the argument; for example, the vote $c\ d\ \cdots$ stands for either of $c\ d\ a\ b$ or $c\ d\ b\ a$.

# 3 Model and Problem Definitions

In this section, we will formalize how cloning is modeled for multiwinner elections. Let $E = (C,V,k)$ be a multiwinner election with $C = \{c_1,\dots,c_m\}$ and $V = (v_1,\dots,v_n)$. Let $K = (K_1,\dots,K_m)$ with $K_i \geq 0$ being a vector, called a *cloning vector*. Intuitively, $K_i$ means that the candidate $c_i$ is cloned $K_i$ times and $c_i$ is replaced by her clones in the multiwinner election. If $K_i = 0$, the candidate $c_i$ is not cloned and simply remains in the multiwinner election. Note that Elkind et al. [1] require that every candidate is cloned at least once, which is equivalent to our definition, but we feel it may be more natural and convenient if one can choose not to clone a candidate.

A multiwinner election $E_K = (C',V',k)$ is *created by cloning from $E = (C,V,k)$ via the cloning vector $K$* if

$$C' = (C \setminus \{c_i \in C \mid K_i \geq 1\}) \cup \{c_i^{(j)} \mid 1 \leq j \leq K_i\}$$

and $V' = (v'_1,\dots,v'_n)$ with each $v'_i \in V'$ being a total order over $C'$ that results from $v_i \in V$ by replacing the cloned candidates in the vote $v_i$ with their clones (i.e., for each clone $c'_i$ of $c_i$, it holds that $c'_i$ is preferred to $c_j \in C'$ in $v'_i$ if and only if $c_i$ is preferred to $c_j$—or $c_j$'s original candidate if $c_j$ is a clone—in $v_i$).

Note that there can be several possible cloned multiwinner elections depending on how the clones of the same candidate are ordered in the votes. The goal of cloning a multiwinner election is to make a distinguished candidate (always called $p$) or one of $p$'s clones a member of at least one winning committee. Regarding the ordering of clones of the same candidate in the votes, we use an optimistic and a pessimistic approach:

- In the *optimistic setting*, cloning via a cloning vector $K$ is considered to be successful if and only if the distinguished candidate (or one of their clones) is a member of a winning committee for *at least one* cloned multiwinner election via $K$.
- In the *pessimistic setting*, cloning via a cloning vector $K$ is considered to be successful if and only if the distinguished candidate (or one of their clones) is a member of a winning committee in *all* cloned multiwinner elections via $K$.

Additionally, as is common in the literature, we adopt the so-called *nonunique-winner model* in which we assume a cloning action to be successful if and only if the distinguished candidate is part of *at least one* winning committee (as opposed to in *all* winning committees, which would be required in the so-called *unique-winner model*).

Furthermore, we consider the cost of cloning candidates. In the *general-cost model* (GC), for every candidate $c_i \in C$ there is a cost function $\rho_i : \mathbb{N} \to \mathbb{N}$ with $\rho_i(0) = \rho_i(1) = 0$ and for each $j, j' \in \mathbb{N}$ with $j < j'$ it holds that $\rho_i(j) \leq \rho_i(j')$. Here, $\rho_i(j)$ is the cost of cloning the $i$-th candidate $j$ times and replacing this candidate in all votes with these clones. There also is an integer $B$, called the *budget*. Additionally, we study two natural special cases of the general-cost model: The *unit-cost model* (UC) in which $\rho_i(j) = j - 1$ for all $i$ and $j \geq 1$ (i.e., every additional clone has unit cost and there is a maximum number of additional clones), and a special case of the unit-cost model, the *zero-cost model* (ZC) in which either the budget is set to infinity, or $\rho_i(j) = 0$ for all $i$ and $j \geq 1$. In the latter cost model, since the budget is not a concern in this case, we seek to find out whether a successful cloning is even possible at all.

We can now define the decision problems we will consider. Let $\mathcal{R}$ be a multiwinner voting rule. In the problem $\mathcal{R}$-POSSIBLE-CLONING-GC, we are given a multiwinner election $E = (C, V, k)$, a cost function $\rho_i : \mathbb{N} \to \mathbb{N}$ for every $c_i \in C$, a distinguished candidate $p \in C$, and a budget $B$, and we ask whether there is a cloning vector $K = (K_1, \ldots, K_m)$ with $\sum_{c_i \in C} \rho_i(K_i) \leq B$ such that $p$ (or one of its clones) is in a winning committee under $\mathcal{R}$ in at least one cloned multiwinner election $E_K$ resulting from $E$ via $K$.

The problem $\mathcal{R}$-NECESSARY-CLONING-GC is defined analogously, except that we ask whether $p$ ends up in a winning committee under $\mathcal{R}$ for all multiwinner elections $E_K$ obtained from $E$ by cloning via $K$. If we use the unit-cost or the zero-cost model in this definition, we replace GC in the problem name by UC or ZC and omit the cost functions in the problem instances, and in the case of the zero-cost model we also omit the budget.

We assume the reader to be familiar with the basic notions of computational complexity theory, both in the classical branch and the parameterized branch. In particular, we will use the classical complexity classes P (*"deterministic polynomial time"*), NP (*"nondeterministic polynomial time"*), and coNP (the class of complements of NP problems); and we will use the notions of *hardness* and *completeness* for NP and coNP based on the *polynomial-time many-one reducibility* (see, e.g., the text books by Papadimitriou [27] and Rothe [28] for formal details). Further, we will use the parameterized complexity class FPT (*"fixed-parameter tractability"*) which, in some sense, is the parameterized pendant of P; we will use the parameterized complexity class $W[1]$ which, in some sense, is the parameterized pendant of NP; and we will use the notion of $W[1]$-*hardness* based on *parameterized reductions* (see, e.g., the text books by Downey and Fellows [29] and Niedermeier [30]).

Since the zero-cost model is a special case of the unit-cost model, which in turn is a special case of the general-cost model, it holds that: $\mathcal{R}$-$\star$-CLONING-ZC reduces to $\mathcal{R}$-$\star$-CLONING-UC, which in turn reduces to $\mathcal{R}$-$\star$-CLONING-GC, where $\star \in \{$POSSIBLE, NECESSARY$\}$.

## 4 Complexity Results for Cloning in Multiwinner Elections

In this section, we present our results on the complexity of cloning in various multiwinner voting rules; see Table 1 for an overview. Question marks in this table indicate open problems and "—" means that influencing the outcome of a multiwinner election via this type of cloning and under this multiwinner voting rule is impossible.

**Table 1** Overview of complexity results for cloning problems in multiwinner voting rules

| | | POSSIBLE-CLONING | | | NECESSARY-CLONING | | |
|---|---|---|---|---|---|---|---|
| voting rule | parameter | ZC | UC | GC | ZC | UC | GC |
| STV | | NP-hard | | | coNP-hard | | |
| SNTV | | P | | | — | | |
| Bloc ($k \geq 2$) | | P | NP-hard | NP-hard | NP-hard | | |
| $k$-Borda | | P | P | NP-hard | NP-hard | | |
| $t$-Approval-CC | #candidates | ? | | | FPT | | |
| | #voters | FPT | | | FPT | | |
| Borda-CC | #voters | ? | ? | $W[1]$-hard | ? | ? | $W[1]$-hard |

## 4.1 STV

We start by showing that possible cloning with zero cost is NP-hard for STV, even if the committee size is fixed to two.

**Theorem 4.1.** *STV-POSSIBLE-CLONING-ZC is NP-hard for instances with committee size $k \geq 2$, even if $k = 2$.*

To prove this theorem, we will need upcoming Lemmas 4.1 and 4.2 and the following easy observation.

**Observation 4.1.** *Cloning a candidate does not change the plurality score of any other candidates or their clones.*

**Lemma 4.1.** *In an STV multiwinner election, the order in which candidates (or their last standing clones) are deleted from the multiwinner election in rounds where the quota is not reached cannot be changed by cloning those candidates.*

*Proof.* Let $(C,V,k)$ be a multiwinner election and let $C' = \{c_1,\dots,c_l\} \subseteq C$ be the set of those candidates that are deleted during the election process. First assume that we can clone candidates from $C'$ at any point during the multiwinner election. In every round, there is one of three possible outcomes:

1. A candidate reaches the quota and is added to the winning committee while $q$ of her first-place votes are removed from the multiwinner election,
2. no one reaches the quota and exactly one candidate has the lowest score and is eliminated, or
3. no one reaches the quota and there are several candidates with the lowest score from which we have to select (with some tie-breaking rule) one candidate that will be eliminated.

Assuming that a candidate is eliminated when her last clone is eliminated, we will show that for all three cases we will arrive at the same outcome if candidates from $C'$ are cloned. We consider the same three cases as above.

1. We cannot clone the candidate that reached the quota since she is part of a winning committee. By Observation 4.1, we cannot prevent that the candidate reaches the quota.
2. Let $c_i \in C'$ be the unique candidate with the lowest score $s$ in this round. If a candidate is cloned, the sum of points of her clones is equal to the score of the original candidate.

Therefore, if candidates are cloned and for some order there are candidates with a score less than $s$, they will be eliminated until the lowest score is $s$ again. Eliminated clones will transfer their points to other clones of their original candidate until only one clone is left who will have the same score as the original candidate. If there is more than one candidate with score $s$, the other candidates distinct from the last clone of $c_i$ with this score are not last clones of their candidates, as this would contradict the assumption that $c_i$ was the only candidate with score $s$ in this round. Therefore, either the last clone of $c_i$ is eliminated or clones with score $s$ are eliminated (without eliminating a last clone of a candidate other than $c_i$) until the last clone of $c_i$ is the only candidate with score $s$.

3. Again, let $s$ be the lowest score in this round and let $C_s \subseteq C'$ be the set of candidates with this score. The sum of scores of the clones of a candidate from $C_s$ equals $s$ and the sum of scores of the clones of a candidate not from $C_s$ is greater than $s$. If there are candidates with scores lower than $s$, they are not last clones of candidates and can be eliminated while transferring their points to other clones of their original candidate. Then, if there are candidates with score $s$ that are not part of $C_s$, they are not last clones of their original candidate. If they are eliminated, no last clone of a candidate not in $C_s$ was eliminated and the candidates with score $s$ are exactly the candidates (or last clones of candidates) from $C_s$ and $s$ is the lowest score.

With the constraint that we can only clone candidates and order clones in the first round and assuming that clones who survived the earlier rounds are only created and ordered in the current round, we have shown the lemma. ❑

**Lemma 4.2.** *In an STV multiwinner election, candidates in winning committees that are always added last to their winning committees can be cloned without changing the outcome of the multiwinner election.*

*Proof.* Let $(C,V,k)$ be a multiwinner election and let $c_i \in C$ be a candidate that is in at least one winning committee but is always added last. We will only consider the first round in which $k-1$ candidates were added to the committee and the sum of scores of the clones of $c_i$ reaches the quota, since it follows from Lemma 4.1 that cloning $c_i$ will have no effect on the outcome in earlier rounds. We will show that one of the clones of $c_i$ will be added to the winning committee. Note that after $k-1$ candidates were added to the winning committee, there cannot be two candidates reaching the quota in the same round.[4] Therefore, if no clone of $c_i$ reaches the quota then nobody does, and candidates will be eliminated in the following rounds until a clone of $c_i$ reaches the quota by gaining points from the elimination of other clones of $c_i$. It does not matter that other candidates may have been eliminated between those rounds, since the multiwinner election ends after a clone of $c_i$ was added to the winning committee. ❑

Now, having shown these two lemmas, we are ready to present the proof of Theorem 4.1.

*Proof of Theorem 4.1.* To show NP-hardness of STV-POSSIBLE-CLONING-ZC, we reduce from the well-known NP-complete problem EXACT-COVER-BY-3-SETS [31] (see also, e.g., [27, 28, 32]): Given a pair $(X,\mathcal{S})$ with $X = \{x_1,\dots,x_{3s}\}$ and $\mathcal{S} = \{S_1,\dots,S_{3s}\}$, where each

[4] To show that, assume for a contradiction that there were two candidates who reach the quota in the same round. Since $k-1$ candidates were added to the winning committee and for each of them $q$ votes ranking them first were removed from the multiwinner election, we have $|V| \geq (k-1)q+2q = (k+1)q$. If $|V|$ is divisible by $k+1$, it holds that $(k+1)q = |V|+k+1$, and if $|V|$ is not divisible by $k+1$, we have $(k+1)q = (k+1)(\lfloor |V|/(k+1) \rfloor + 1) = (k+1)(\lceil |V|/(k+1) \rceil) > |V|$, which is a contradiction in both cases.

**Table 2** Votes $V$ for the proof of Theorem 4.1

| number | vote | for |
|---|---|---|
| $9\frac{s^2}{2}+49\frac{s}{2}+13$ | $d\ e\ f\ p\ \cdots$ | |
| $1$ | $d\ S_i\ c\ p\ \cdots$ | $1 \le i \le 3s$ |
| $\frac{s}{2}+1$ | $S_i\ e\ p\ \cdots$ | $1 \le i \le 3s$ |
| $\frac{s}{2}+2$ | $S_i\ f\ p\ \cdots$ | $1 \le i \le 3s$ |
| $s+5$ | $x_i\ \mathcal{S}_{x_i}\ c\ p\ \cdots$ | $1 \le i \le 3s$ |
| $\frac{s}{2}+2$ | $b_i\ S_i\ e\ p\ \cdots$ | $1 \le i \le 3s$ |
| $\frac{s}{2}+2$ | $b_i\ S_i\ f\ p\ \cdots$ | $1 \le i \le 3s$ |
| $4s+8$ | $p\ c\ \cdots$ | |
| $4s+7$ | $c\ p\ e\ \cdots$ | |
| $4s+4$ | $e\ p\ c\ \cdots$ | |
| $4s+4$ | $f\ p\ c\ \cdots$ | |

$S_i \subseteq X$ has exactly three elements, does there exist an *exact cover of* $X$, i.e., a subfamily $\mathcal{S}' \subseteq \mathcal{S}$, $|\mathcal{S}'| = s$, such that $\bigcup_{S_i \in \mathcal{S}'} = X$? As is common, we abbreviate this problem by X3C.

Let $(X,\mathcal{S})$ be such an instance of X3C. Without loss of generality, we may assume that every $x_i \in X$ appears in exactly three sets in $\mathcal{S}$; that this restriction of X3C is still NP-complete was shown by Gonzalez [33]. We also assume that $s \ge 3$ is even, which can be achieved by duplicating the instance. From this instance of X3C, we construct our STV-POSSIBLE-CLONING-ZC instance. The set of candidates is

$$C = \{p,c,d,e,f\} \cup X \cup \mathcal{S} \cup B,$$

where $B = \{b_1,\dots,b_{3s}\}$ and $p$ is the distinguished candidate. Set the committee size to $k = 2$. Since we are in the zero-cost model, the budget is set to infinity. For each $x_i \in X$, let $\mathcal{S}_{x_i} = \{S_j \in \mathcal{S} \mid x_i \in S_j\}$. We define $V$ to consist of the votes shown in Table 2.

We will break ties according to the linear order $\overrightarrow{B}\,\overrightarrow{C \setminus B}$.

To complete the proof of Theorem 4.1, we have to show the following claim.

**Claim 4.1.** $(X,\mathcal{S})$ *is a yes-instance of* X3C *if and only if $p$ can be made an STV winner of at least one winning committee obtained from* $(C,V,2)$ *by cloning, i.e., we have a yes-instance of STV-*POSSIBLE-CLONING-ZC.

*Proof.* From left to right, suppose there is an exact cover $\mathcal{S}'$ of $X$. Clone $d$ twice, let us call them $d^{(1)}$ and $d^{(2)}$, and order them such that $d^{(1)}$ takes the first position in all votes of the form $d\ S_i\ p\ c$ for every $S_i \in \mathcal{S}'$ (i.e., $d^{(1)}\ d^{(2)}\ S_i\ p\ c$) and $d^{(2)}$ is in front of $d^{(1)}$ in all other votes. We will show that $p$ is now part of a winning committee:

- In the first round, $d^{(2)}$ reaches exactly the quota and is added to the winning committee. Therefore, all votes where $d^{(2)}$ is in the first position are removed from the multiwinner election. Since $d^{(1)}$ was on top of the votes of the form $d^{(1)}\ d^{(2)}\ S_i\ p\ c$ for every $S_i \in \mathcal{S}'$, those votes are still present in the second round.
- In the second round, the scores of all candidates but $d^{(1)}$ are the same (except $e$ who now scores $s$ points fewer but still scores more points than $d^{(1)}$) as in the second round of the original multiwinner election while $d^{(1)}$ has $s$ points now. Therefore, $d^{(1)}$ is eliminated from the multiwinner election in the second round and transfers her points to all $S_i \in \mathcal{S}'$ equally.
- In the third round, we have the following scores:

| Candidate | $p$ | $c$ | $e$ | $f$ |
|---|---|---|---|---|
| Score | $4s+8$ | $4s+7$ | $4s+4$ | $4s+4$ |
| Candidate | $b_i \in B$ | $x_i \in X$ | $S_i \in \mathcal{S} \setminus \mathcal{S}'$ | $S_i \in \mathcal{S}'$ |
| Score | $s+4$ | $s+5$ | $s+3$ | $s+4$ |

No candidate reaches the quota and all $S_i \in \mathcal{S} \setminus \mathcal{S}'$ have the lowest score.

- Therefore, all candidates from $\mathcal{S} \setminus \mathcal{S}'$ are eliminated in the following $2s$ rounds (the order does not matter since the eliminated candidates will transfer their points only to $e$ and $f$). Then we have the following scores in the next round:

| Candidate | $p$ | $c$ | $e$ | $f$ | $b_i \in B$ | $x_i \in X$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|---|---|
| Score | $4s+8$ | $4s+7$ | $s^2+6s+4$ | $s^2+8s+4$ | $s+4$ | $s+5$ | $s+4$ |

Note that all candidates from $B$ and $\mathcal{S}'$ have the lowest score and due to the tie-breaking rule, the candidates from $B$ are eliminated first in the next $3s$ rounds. Those eliminated candidates will transfer their points to the still standing candidates from $\mathcal{S}$ or $e$ and $f$.

- In the next round, we have the following scores:

| Candidate | $p$ | $c$ | $e$ | $f$ | $x_i \in X$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|---|
| Score | $4s+8$ | $4s+7$ | $2s^2+10s+4$ | $2s^2+12s+4$ | $s+5$ | $2s+8$ |

- In the next $3s$ rounds, all candidates from $X$ are eliminated while each $x_i \in X$ transfers her $s+5$ points to a candidate from $\mathcal{S}_{x_i}$. Note that there is always exactly one candidate from $\mathcal{S}_{x_i}$ still standing in the multiwinner election since $\mathcal{S}'$ is an exact cover. This means that every $S_j \in \mathcal{S}'$ gains $s+5$ points from each $x_i \in S_j$. Furthermore, $c$ does not gain any more points in those rounds, which gives the following scores:

| Candidate | $p$ | $c$ | $e$ | $f$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|
| Score | $4s+8$ | $4s+7$ | $2s^2+10s+4$ | $2s^2+12s+4$ | $5s+23$ |

- We can see that $c$ has the least number of points and is eliminated in this round, transferring her points to $p$. Since $8s+15 > 5s+23$ with $s \geq 3$, all candidates from $\mathcal{S}'$ are eliminated in the next $s$ rounds, each transferring $3s+15$ points to $p$, $s+3$ points to $e$, and $s+4$ points to $f$. Then we have the following scores:

| Candidate | $p$ | $e$ | $f$ |
|---|---|---|---|
| Score | $3s^2+19s+8$ | $3s^2+13s+4$ | $3s^2+16s+4$ |

- We can see that $e$ is eliminated in the next round, transferring her points to $p$ who reaches the quota in the round thereafter.

For the converse direction, assume there is no exact cover of $X$. From Lemmas 4.1 and 4.2 we know that cloning candidates other than $d$ has no effect on the outcome of the multiwinner election. Note that candidate $d$ has $s$ points more than needed to reach the quota and $d$ will not gain any additional points before $p$ is eliminated. If the clones of $d$ are ordered in a way such that no clone reaches the quota and every clone has at least $4s+9$ points then the multiwinner election proceeds as if $d$ were not cloned and added to the winning committee up to the point in time when $p$ is eliminated from the multiwinner election.

If there are clones with fewer than $4s+9$ points, they will be eliminated before the elimination of $p$ and transfer their points to other clones of $d$. If all clones with fewer than $4s+9$ points are eliminated and there is still no clone who reaches the quota, we have the same situation as before where $p$ will be eliminated. If at some point a clone of $d$ reaches the quota (and $p$ is still present in the multiwinner election), she will be added to the winning committee and all but up to $s$ of her first-place votes will be removed, leaving $s$ votes where $d$ was in the first position in the original multiwinner election. Since $q$ arbitrary first-place votes are removed if a clone of $d$ has more first-place votes than the quota, we can definitely "save" some of those votes only by cloning $d$ and assigning clones to the top of those votes that are not added to the winning committee. Note that if $d$ is not cloned at all, $d$ reaches the quota with $s$ extra votes. Due to arbitrary tie-breaking of votes we might still be lucky and (at most) $s$ votes of the form $d\ S_i\ c\ p$ are not removed from the election. Then we arrive at the same situation as below.

We will now show that it does not matter which s votes are prevented from being removed from the multiwinner election when a clone of $d$ reaches the quota, since $p$ will always be eliminated when there is no exact cover. Firstly, whenever a clone of $d$ reaches the quota and is added to the winning committee, all remaining clones will be eliminated next, since they have at most $s$ points and all other candidates have more than $s$ points at any time. Secondly, saving votes of the form $d\ e\ f\ p\ \cdots$ from being removed is not advantageous for $p$, since she can beat $e$ and $f$ only much later in the multiwinner election (as can be seen in the original election). Also, the other votes that can be saved will give $p$ additional points only if $c$ is deleted earlier than $p$. Note that in the original multiwinner election the candidates from $\mathcal{S}$ were eliminated immediately after $d$ was added to the winning committee. By saving some votes of the form $d\ S_i\ c\ p\ \cdots$ we can save up to $s$ candidates in $\mathcal{S}$ from being eliminated in the first $5s+1$ rounds; let $\mathcal{S}'$ be the set of those candidates. Instead of the candidates from $\mathcal{S}$ without those up to $s$ candidates, the members of $B$ and $X$ can be eliminated earlier. Note that when candidates from $B$ are eliminated, they are tieing the candidates from $\mathcal{S}'$ in points but we will see soon that we want the candidates from $\mathcal{S}'$ to be eliminated as late as possible for $p$ to have a chance to survive longer.

Without candidates from $B$, the remaining candidates from $\mathcal{S}$ now have more points than $p$. Since we cannot prevent the candidates from $X$ from being eliminated before $c$, those candidates will transfer their points to either $c$ or a candidate from $\mathcal{S}'$ that is still standing. To be precise, a candidate $x_i$ will transfer her $s+5$ points to a still-standing candidate from $S_{x_i} \cap \mathcal{S}'$ or to $c$ if all candidates corresponding to members of $S_{x_i}$ have already been eliminated.

If $c$ gains points during the elimination of the candidates from $X$, $c$ will have more points than $p$. Therefore, $p$ only survives the round after the elimination of all candidates from $X$ if for every $x_i$ there is an $S_j \in \mathcal{S}'$ with $S_j \in \mathcal{S}_{x_i}$ that is still present in the multiwinner election. Since $|\mathcal{S}'| \leq s$ and every $S_j \in \mathcal{S}$ is in exactly three subsets $\mathcal{S}_{x_i}$, this is only possible if $\mathcal{S}'$ is an exact cover, which contradicts the assumption that there is none. ❑ Claim 4.1

This completes the proof of our theorem. ❑ Theorem 4.1

**Table 3** Votes $V$ for the proof of Theorem 4.2

| number | vote | for |
|---|---|---|
| $25s+2$ | $d_j\ r_1\ r_2\ S_1\ p\ \cdots$ | $1 \leq j \leq s$ |
| 1 | $d_j\ r_1\ r_2\ S_i\ p\ \cdots$ | $1 \leq j \leq s$ and $1 \leq i \leq 3s$ |
| 2 | $S_i\ e_i\ p\ \cdots$ | $1 \leq i \leq 3s$ |
| 3 | $b_i\ S_i\ f_i\ p\ \cdots$ | $1 \leq i \leq 3s$ |
| 4 | $x_i\ \mathcal{S}_{x_i}\ r_1\ p\ \cdots$ | $1 \leq i \leq 3s$ |
| 2 | $p\ \cdots$ | |
| 1 | $r_1\ p\ \cdots$ | |
| 1 | $r_2\ r_1\ p\ \cdots$ | |
| 5 | $e_i\ p\ \cdots$ | $1 \leq i \leq 3s$ |
| 4 | $f_i\ p\ \cdots$ | $1 \leq i \leq 3s$ |

Note that, by Lemmas 4.1 and 4.2, influencing the result of the multiwinner election by cloning is impossible if $k = 1$. This is, in fact, not very surprising, since single-winner STV is independent of clones [13].

The reduction above can be modified to show that constructive control by adding candidates—see the book chapters by Baumeister and Rothe [5] and Faliszewski and Rothe [6] for its definition and an overview of results for it—is NP-hard for STV.

Regarding STV-NECESSARY-CLONING-ZC, we can show that it is coNP-hard. Note that, in contrast to the STV-POSSIBLE-CLONING-ZC variant, we cannot fix $k$ here.

**Theorem 4.2.** *STV-*NECESSARY-CLONING-ZC *is* coNP*-hard.*

*Proof.* To show coNP-hardness of STV-NECESSARY-CLONING-ZC, we now reduce from the complement of X3C to STV-NECESSARY-CLONING-ZC. Let $(X, \mathcal{S})$ with $X = \{x_1, \ldots, x_{3s}\}$ and $\mathcal{S} = \{S_1, \ldots, S_{3s}\}$ be a given X3C instance and, again, assume that every $x_i \in X$ appears in exactly three elements of $\mathcal{S}$ (recall the result by Gonzalez [33]). We also assume that $s \geq 3$, which can be achieved by duplicating the instance. The set of candidates is

$$C = \{p, r_1, r_2\} \cup X \cup \mathcal{S} \cup B \cup D \cup E \cup F,$$

where $B = \{b_1, \ldots, b_{3s}\}$, $D = \{d_1, \ldots, d_s\}$, $E = \{e_1, \ldots, e_{3s}\}$, $F = \{f_1, \ldots, f_{3s}\}$, and $p$ is the distinguished candidate. Set the committee size to $k = s+1$. Since we are in the zero-cost model, the budget is set to infinity. For each $x_i \in X$, let $\mathcal{S}_{x_i} = \{S_j \in \mathcal{S} \mid x_i \in S_j\}$. We define $V$ to consist of the votes shown in Table 3.

We will break ties according to the linear order $\overrightarrow{X}\ p\ r_1\ r_2\ \overrightarrow{B}\ \overrightarrow{\mathcal{S}}\ \overrightarrow{D}\ \overrightarrow{F}\ \overrightarrow{E}$. It does not matter how ties are broken if more than one candidate reaches the quota, or which votes are removed from the multiwinner election if a candidate scores more points than the quota.

Let us analyze the multiwinner election $(C, V, s+1)$ we have just constructed. Since $|V| = 54s + s(28s+2) + 4$, the quota is

$$\left\lfloor \frac{54s + s(28s+2) + 4}{s+2} \right\rfloor + 1 = 28s + \left\lfloor \frac{4}{s+2} \right\rfloor + 1 = 28s + 1.$$

Each candidate $d_j \in D$ reaches the quota with $28s+2$ points and is added to the winning committee, and all but one vote $d_j\ \cdots$ for each $d_j \in D$ is removed from the multiwinner election. Since $d_j$ is removed from each remaining vote, $r_1$ gains $s$ points. In the following round, no one reaches the quota and $r_2$ is removed from the multiwinner election. In the next round, $p$ and every candidate from $\mathcal{S}$ are tied for the lowest score, so $p$ is eliminated due to the tie-breaking rule and is not part of the winning committee.

To complete the proof of Theorem 4.2, we will now show that $(X, \mathcal{S})$ is a no-instance of X3C if and only if $p$ can be made part of at least one winning committee obtained from $(C, V, s+1)$ by cloning, i.e., we have a yes-instance of STV-NECESSARY-CLONING-ZC.

From left to right, suppose there is no exact cover of $X$. We now show that there is a cloning vector such that $p$ is part of a winning committee for every possible ordering of clones. Specifically, consider the cloning vector in which every candidate from $D$ is cloned twice and consider the following three cases of how clones of a $d_j \in D$ can be ordered:

**(1)** one clone reaches the quota and the other has a score of one,
**(2)** one clone reaches the quota and the other has a score of zero (i.e., the ordering of clones is always the same for the votes where $d_j$ was in the top position), and
**(3)** both clones do not reach the quota.

In the first two cases, the candidate who reaches the quota, say $d_j^{(1)}$, will be added to the winning committee and, after all but one of her top position votes were removed from the multiwinner election, there is now a vote $d_j^{(2)}\ r_1\ r_2\ S_i\ p$ in which the other clone, $d_j^{(2)}$, is in the top position and scores one point.

In the third case, both clones score at least two points and the multiwinner election continues without adding any one of them to the winning committee.

Note that, in all three cases, $r_1$ and $r_2$ do not gain points and, after all clones of candidates from $D$ who reach the quota were added to the winning committee, the remaining clones have score at most one. So, $r_1$ and $r_2$ are eliminated from the multiwinner election in the next two rounds and after that all second clones of candidates from the cases (1) and (2) as well. At some point during the following rounds, for each $d_j \in D$ whose clones are ordered according to case (3), one clone might be eliminated, which would lead to the other clone reaching the quota in the next round. Either some $S_i \in \mathcal{S}$ or $p$ gains a point from the then not removed vote of the form $d_j\ \cdots$. The latter would help $p$ reaching the quota (but it is not needed), so we assume the worst case that some $S_i \in \mathcal{S}$ gains a point and that the clones from case (3) are eliminated or added to the winning committee now.

Therefore, as soon as $r_1$ and $r_2$ and all clones of candidates from $D$ are not part of the multiwinner election anymore, there is a subset $\mathcal{S}' \subseteq \mathcal{S}$, $|\mathcal{S}'| \leq s$, of candidates from $\mathcal{S}$ who gained at least one and up to $s$ points from the removed clones of candidates from $D$. Then we have the following scores:

| Candidate | $p$ | $b_i \in B$ | $e_i \in E$ | $f_i \in F$ | $x_i \in X$ | $S_i \in \mathcal{S} \setminus \mathcal{S}'$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|---|---|
| Score | 4 | 3 | 5 | 4 | 4 | 2 | $\geq 3$ |

Therefore, no one reaches the quota in the following round, so all candidates from $\mathcal{S} \setminus \mathcal{S}'$ (transferring their points to candidates from $E' = \{e_i \in E \mid S_i \in \mathcal{S} \setminus \mathcal{S}'\}$) and $B$ (transferring their points to candidates from $F' = \{f_i \in F \mid S_i \in \mathcal{S} \setminus \mathcal{S}'\}$ and $\mathcal{S}'$) are eliminated. Then the scores for the remaining candidates are as follows:

| Candidate | $p$ | $e_i \in E'$ | $e_i \in E \setminus E'$ | $f_i \in F'$ | $f_i \in F \setminus F'$ | $x_i \in X$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|---|---|
| Score | 4 | 7 | 5 | 7 | 4 | 4 | $\geq 6$ |

Due to the tie-breaking rule, each candidate $x_j \in X$ is now eliminated transferring each of her four points to either a candidate from $\mathcal{S}_{x_j}$ if $\mathcal{S}_{x_j} \cap \mathcal{S}' \neq \emptyset$, or else to $p$. It follows that $p$ does not gain points during this round only if $\mathcal{S}'$ is a cover of $X$, as then, for every candidate $x_j \in X$, there would be one candidate from $\mathcal{S}'$ sitting between $x_j$ and $p$ in those four votes of the form $x_j\ \mathcal{S}_{x_j}\ r_1\ p$. Since $|\mathcal{S}'| \leq s$, the cover $\mathcal{S}'$ must be an exact cover, which is not possible. Therefore, $p$ gains at least four points from the elimination of candidates from $X$. Since $p$ now has at least eight points and the scores of candidates from $F$ and $E$ did not change, all those candidates are eliminated, transferring their points to $p$. Note that $|E'| = |F'| = |\mathcal{S} \setminus \mathcal{S}'| \geq 2s$. Then the score of $p$ is at least

$$8 + 3s(5+4) + (3+2)|\mathcal{S} \setminus \mathcal{S}'| = 27s + 5|\mathcal{S} \setminus \mathcal{S}'| + 8 \geq 37s + 8.$$

Therefore, $p$ is added to the winning committee.

Conversely, assume there is an exact cover $\mathcal{S}'$ of $X$. We must show that for every cloning vector, there is an order of clones such that $p$ is not part of a winning committee.

From Lemmas 4.1 and 4.2 we know that cloning candidates other than from $D$ has no effect on the outcome of the multiwinner election. Therefore, we assume that those candidates were not cloned, and we focus on the clones of candidates from $D$ only.[5]

If a candidate $d_j \in D$ were not cloned or were only cloned once, then $r_1$ would gain a point after this candidate or her clone is added to the winning committee and, similarly to the original multiwinner election, $p$ would later be eliminated since $p$'s score would remain two.

We now assume that every $d_j \in D$ was cloned twice.[6] First, rename the candidates from $\mathcal{S}'$ such that $\mathcal{S}' = \{S'_1, \ldots, S'_s\}$ . Then, for every $d_j \in D$, order both clones such that the second clone is in front of the first clone in one vote of the form $d_j\ r_1\ r_2\ S'_j\ p$ and in the reverse order in every other vote.

It follows that, for every $d_j \in D$, the first clone reaches exactly the quota and all her first-place votes are removed from the multiwinner election. Then we have one vote of the form $d_j^{(2)}\ r_1\ r_2\ S'_j\ p$ for each $d_j \in D$ with $d_j^{(2)}$ being the second clone of $d_j$. Meanwhile, $r_1$ and $r_2$ did not gain points, so they are eliminated in the next two rounds due to tie-breaking after which the remaining clones are eliminated, transferring one point each to the candidates of $\mathcal{S}'$. Then we have the following scores:

| Candidate | $p$ | $b_i \in B$ | $e_i \in E$ | $f_i \in F$ | $x_i \in X$ | $S_i \in \mathcal{S} \setminus \mathcal{S}'$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|---|---|
| Score | 4 | 3 | 5 | 4 | 4 | 2 | 3 |

First every candidate from $\mathcal{S} \setminus \mathcal{S}'$ and then every candidate from $B$ is eliminated, and we have the scores:

[5] Alternatively, fix some arbitrary ordering of clones of candidates not from $D$ for one vote and repeat it for every other vote. Then all but one clone of each candidate from $C \setminus D$ have score zero and would be eliminated after the first round where the quota is not reached, leaving one clone for every candidate with the same score as if this candidate were not cloned.

[6] If a candidate $d_j \in D$ is cloned more than twice, order all but the first two clones behind those first two clones in every vote. Then all but the first two clones score zero points and are eliminated after the first round where the quota was not reached and before either of the first two clones are removed from the multiwinner election.

| Candidate | $p$ | $e_i \in E$ | $f_i \in F$ | $x_i \in X$ | $S_i \in \mathcal{S}'$ |
|---|---|---|---|---|---|
| Score | 4 | $\geq 5$ | $\geq 4$ | 4 | 6 |

Now, every $x_i \in X$ is eliminated due to tie-breaking, but since $\mathcal{S}'$ is an exact cover of $X$ for every $x_i \in X$, there is one candidate from $\mathcal{S}_{x_i}$ still present in the multiwinner election. Hence, all points from candidates of $X$ are transferred to candidates from $\mathcal{S}'$ and $p$ still has only four points. This leads to $p$ being eliminated due to tie-breaking in the next round, so $p$ is not part of a winning committee. ❑

## 4.2 SNTV

For SNTV where eliminated candidates do not transfer their points, the possible-cloning problem is easy to solve, even in the general-cost model (and thus also in the other two models).

**Theorem 4.3.** *SNTV*-POSSIBLE-CLONING-GC *is in* P.

*Proof.* For the distinguished candidate $p$ to be in a winning committee, $p$ needs to be among the $k$ best candidates with respect to their plurality score. By cloning, we can only decrease a candidate's score by splitting this candidate's points and distributing them among her clones. Therefore, we need to decrease the points of sufficiently many candidates that outscore the distinguished candidate $p$ by creating $k_a = \left\lfloor \frac{\text{score}(a)}{\text{score}(p)} \right\rfloor$ clones of a candidate $a \in C \setminus \{p\}$.[7]

Let $C' \subseteq C$ be the set of candidates with score greater than the score of $p$. For every candidate $a$ in $C'$, compute the cost of creating $k_a$ clones of $a$ and check whether the sum of the $|C'| - (k-1)$ smallest costs does not exceed the budget $B$, and accept accordingly. ❑

**Corollary 4.1.** *SNTV*-POSSIBLE-CLONING-UC *and SNTV*-POSSIBLE-CLONING-ZC *are in* P.

## 4.3 Bloc Voting

Since bloc voting is equivalent to plurality for committee size $k = 1$, Bloc-POSSIBLE-CLONING-GC is in P and necessary cloning (even in the general-cost model) is impossible due to the results of Elkind et al. [1].

For bloc voting with committee size $k \geq 2$, we will later show how to obtain NP-hardness for necessary cloning, even in the zero-cost model (and thus also in the other two models). For possible cloning in the zero-cost model, however, the problem is in P for each committee size. We start by presenting the proof of this result.

**Theorem 4.4.** *Bloc*-POSSIBLE-CLONING-ZC *is in* P.

*Proof.* To show this we provide a simple algortihm that solves this problem in polynomial time. Let $(C,V,k)$ be our given multiwinner election, $p$ the distinguished candidate, and since we are in the zero-cost model, the budget is set to infinity.

First, scan the first position of all votes and check if $p$ is ranked first in any of them.

*If this is the case*: The distinguished candidate $p$ is cloned $k-1$ times and every other candidate is cloned $k \cdot |V|$ times. Now, all clones of $p$ have at least one point (more if more

[7] Elkind et al. [1] showed that this is the least number of clones that needs to be added so that $p$ can overtake candidate $a$.

votes have ranked $p$ first) and every other candidate has at most one point, in some ordering of the clones, because even if every other vote had the same candidate ranked first, enough clones of this candidate were created to fill the first $k$ positions of every vote with two different clones of this candidate such that each of these clones now has at most one point. Thus a clone of $p$ now belongs to a winning committee of the cloned election.

*If this is not the case*: Check if $p$ is ranked among the first $k$ positions in any vote.

*If this is not the case*: Reject the input because $p$ cannot be made part of a winning committee through cloning this election because $p$ cannot gain any points, no matter which candidates are cloned.

*If this is the case*: Without loss of generality, we may assume that in this vote the candidates in $W = \{c_1, \ldots, c_{k-1}\}$ are ranked among the first $k$ positions along with $p$. Now, every candidate $c \in C \setminus (W \cup \{p\})$ is cloned $k \cdot |V|$ times. Again, we can order these clones in a way that every candidate other than those in $W \cup \{p\}$ have at most one point, while $p$ and the candidates in $W$ have at least one point. Thus $p$ can be made part of a winning committee through cloning this election, so we accept our input. ❑

For committee size $k \geq 2$ in bloc voting, we now show that possible cloning is NP-hard in the unit-cost (and thus also in the general-cost) model.

**Theorem 4.5.** *Bloc-*POSSIBLE-CLONING-UC *is* NP*-hard for instances with committee size* $k \geq 2$*, even if* $k = 2$.

*Proof.* To show NP-hardness, we now reduce from the well-known NP-complete problem HITTING-SET [31] (see also, e.g., [27, 28, 32]): Given a triple $(X, \mathcal{S}, r)$ with $X = \{x_1, \ldots, x_s\}$ and $\mathcal{S} = \{S_1, \ldots, S_t\}$ such that $S_i \subseteq X$ for each $i$, $1 \leq i \leq t$, does there exist a *hitting set* of size at most $r$, i.e., a subset $X' \subseteq X$ with $|X'| \leq r$ such that $X' \cap S_i \neq \emptyset$ for each $i$, $1 \leq i \leq t$?

Let $(X, \mathcal{S}, r)$ be such an instance of HITTING-SET. Without loss of generality, we may assume that $|S_i| \leq r$ for each $i$, $1 \leq i \leq t$, that $t \geq r+2$, and that $\mathcal{S}$ does not contain a singleton, say $\{x_j\}$, such that $x_j \in X$ occurs only in this singleton and in none of the other subsets of $X$ contained in $\mathcal{S}$. From this instance of HITTING-SET, we construct our Bloc-POSSIBLE-CLONING-UC instance as follows. Set the committee size to $k = r+1$, so $r = k-1$. Note that $t \geq r+2 = k+1$ and thus $t > k$. Define the set $C = \{p, l, w\} \cup X \cup D \cup E$ of candidates with $D = \{d_1, \ldots, d_{k-1}\}$ and $E = \{e_1, \ldots, e_{t(k-2)}\}$, where $p$ is the distinguished candidate we want to make part of a winning committee. There are four kinds of voters in $V$:

1. $t$ voters who rank the candidates from $D$ first (in any order) and rank $p$ in the $k$-th position.
2. $t$ voters who rank $w$ first, followed by the candidates from $D$ (in any order).
3. One voter who ranks the candidates from $D$ first (in any order) and ranks $l$ in the $k$-th position.
4. For each $S_i \in \mathcal{S}$, there is one voter ranking the candidates from $S_i$ first (in any order), followed by any $k-1-|S_i|$ candidates from $E$ such that no candidate from $E$ appears more than once, and ranking $w$ in the $k$-th position.

We can now see that the candates in $C$ score the following points:

- Candidate $p$ has exactly $t$ points.
- Candidate $l$ has exactly one point.
- Each $e_i \in E$ has at most one point.
- Each $d_i \in D$ has exactly $2t+1$ points.

- Each $x_i \in X$ has at most $t$ points.
- Candidate $w$ has exactly $2t$ points.

It is easy to see that $p$ is not part of a winning size-$k$ committee since the $k$ candidates in $\{w\} \cup D$ each score more points than $p$. We will now show that $p$ can be made part of a winning committee of this election via possible cloning of at most $r$ candidates with unit-cost if and only if $(X, \mathcal{S}, r)$ is a yes-instance of HITTING-SET.

From right to left, suppose there is a hitting set $X'$ of size at most $r$. Clone each candidate from $X'$ once. Because $X'$ is a hitting set, we clone at least one candidate from the first $k-1$ positions of each vote in group 4. Thus, in each of those votes, $w$ is pushed out of the first $k$ positions and does not score a point anymore, lowering its score to $t$ points. Also, every new candidate (i.e., every clone) has at most $t$ points, the same as its original candidate. Therefore, $p$ now caught up to $w$ in points and is now part of a winning committee of this election.

From left to right, assume that $((C,V,k),r)$ is a yes-instance for Bloc-POSSIBLE-CLONING-UC. This means that it is possible to make $p$ part of a winning committee by cloning at most $k-1$ times. To make $p$ part of a winning committee of this election, $p$ needs to catch up in points to at least one candidate from $\{w\} \cup D$. However, $p$ cannot catch up to the candidates from $D$, as they cannot be cloned (note that this would result in $p$ losing all of her points in voter group 1). Therefore, they cannot be removed from the first $k$ positions of the votes from groups 1 and 3, and therefore, they always have at least $t+1$ points.

This means that $p$ must have caught up to $w$ in points. This could not have been achieved by cloning $w$, because even if $w$ were cloned $k-1$ times, each clone would still get $t$ points from voter group 2, while some clones would still get points from voter group 4, beating $p$. Also, this cannot be achieved by cloning candidates from $E$, because there are more than $k-1$ votes in group 4. Thus it must have been achieved by cloning candidates from $X$.[8] But if $p$ caught up to $w$ by cloning candidates from $X$, then $w$ lost all its points from voter group 4. This means that the set of candidates that have been cloned in this way forms a hitting set of size at most $r$. ❑

**Corollary 4.2.** *Bloc-*POSSIBLE-CLONING-GC *is* NP-*hard for instances with committee size* $k \geq 2$*, even if* $k = 2$.

Next, we turn to necessary cloning with zero cost for bloc voting, again showing NP-hardness.

**Theorem 4.6.** *Bloc-*NECESSARY-CLONING-ZC *is* NP-*hard for instances with committee size* $k \geq 2$*, even if* $k = 2$.

*Proof.* For fixed $t \geq 2$, $t$-approval-NECESSARY-CLONING-ZC was shown to be NP-hard by Elkind et al. [1]. We will reduce 2-approval-NECESSARY-CLONING-ZC to Bloc-NECESSARY-CLONING-ZC.[9] Let $((C,V),p)$ be an instance of 2-approval-NECESSARY-CLONING-ZC. From this instance, we construct an instance of Bloc-NECESSARY-CLONING-ZC, where bloc voting is a multiwinner voting rule. Set the committee size to $k = 2$, so bloc voting uses 2-approval scores. We create an additional candidate $w \notin C$ and a set $D$ of $|V|+1$ additional dummy candidates. Next, we create a list $V'$ of $|V|+1$ votes which have $w$ in the first position and a dummy candidate from $D$ in the second position such that every

[8] Here, we use our without-loss-of-generality assumption that $\mathcal{S}$ does not contain a singleton $\{x_j\}$ with $x_j \in X$ occurring only in this singleton and in none of the other members of $\mathcal{S}$. Indeed, without that assumption, it might be possible to clone a candidate from $E$ instead of $x_j$ in the corresponding vote.

[9] Note that $t$-approval is here understood as a single-winner voting rule; therefore, we do not specify a committee size.

dummy candidate only scores one point from those new votes. The other candidates can be ordered arbitrarily. Furthermore, the new candidates are ordered last in all votes of $V$. We show that $((C,V),p)$ is a yes-instance of 2-approval-NECESSARY-CLONING-ZC if and only if $((C\cup D\cup\{w\},V\cup V',2),p)$ is a yes-instance of Bloc-NECESSARY-CLONING-ZC.

From left to right, assume that $((C,V),p)$ is a yes-instance of 2-approval-NECESSARY-CLONING-ZC. Then we can clone candidates from $C$ such that $p$ has the highest score in $(C,V)$. Note that the score of $p$ is larger than 1 and at most $|V|$. Thus we can clone candidates from $C$ such that $p$ has the second-highest score in the multiwinner election $(C\cup D\cup\{w\},V\cup V',2)$, since the candidates from $C$ do not gain additional points from $V'$, all additional dummy candidates score only one point, and $w$ scores with $|V|+1$ points more points than $p$. Therefore, $p$ is in a winning committee of $(C\cup D\cup\{w\},V\cup V',2)$.

For the converse direction, assume that $((C,V),p)$ is a no-instance of 2-approval-NECESSARY-CLONING-ZC. Then, whichever candidates of $C$ we clone, $p$ is never a winner of $(C,V)$, which means that there always is a candidate with a higher score than $p$. Therefore, $p$ is always behind one candidate of $C$ in the multiwinner election $(C\cup D\cup\{w\},V\cup V',2)$ as well, since cloning $w$ or any dummy candidates does not change the allocation of points in $V$ and no candidate of $C$ gains additional points from the votes in $V'$. If $p$ has the second-highest score of all candidates in $C$, it could still reach a winning committee if we could reduce the score of $w$ by cloning $w$, but this is not possible since the voters of $V'$ could order the clones of $w$ such that one clone scores $|V'|=|V|+1$ points, which is a higher score than any candidate in $C$ can have. It follows that $p$ cannot be in any winning committee of $(C\cup D\cup\{w\},V\cup V',2)$ if the order of clones cannot be controlled. ❑

## 4.4 *k*-Borda

Elkind et al. [1] proved that $k$-Borda-POSSIBLE-CLONING-GC is NP-hard for the single-winner version. This lower bound immediately transfers to the multiwinner variant of the problem, provided we are cloning in the general-cost model. When restricted to unit costs, however, we can show that it is easy to solve.

**Theorem 4.7.** *$k$-Borda-*POSSIBLE-CLONING-UC *is in* P.

*Proof.* For every candidate $a\in C$, let $n_a$ be the number of votes in which $p$ is preferred to $a$ and compute the value $k_a=\lceil dist(a,p)/n_a\rceil$.[10] Let $r$ be the $k$-th highest value among the values $k_a$ just computed. Then create $r+1$ clones of $p$. This leads to $p$'s score being the $k$-th highest score, so $p$ is in a winning committee of size $k$. ❑

On the other hand, the problem of necessary cloning in the zero-cost model becomes NP-hard for $k$-Borda, even for size-1 committees.

**Theorem 4.8.** *$k$-Borda-*NECESSARY-CLONING-ZC *is* NP*-hard for instances with committee size $k\geq 1$, even if $k=1$.*

*Proof.* We prove NP-hardness by reducing X3C to 1-Borda-NECESSARY-CLONING-ZC.

Given an X3C instance $(X,\mathcal{S})$ with $X=\{x_1,\dots,x_{3s}\}$ and $S=\{S_1,\dots,S_{3s}\}$ (again, we assume that every $x_i\in X$ appears in exactly three elements of $\mathcal{S}$), the candidate set is $C=\{p,a,d\}\cup X\cup\mathcal{S}$ and $V$ is defined to consist of the following eight groups of votes:

[10]Elkind et al. [1] showed that creating $k_a+1$ clones of $p$ is just enough and also the optimal way for one of $p$'s clones to at least tie $a$ in points.

(1) $7s+1$ votes of the form $a\ p\ \overrightarrow{X}\ \mathcal{S}\ d$ and $7s+1$ votes of the form $\overleftarrow{X}\ a\ p\ \mathcal{S}\ d$.
(2) One vote $\overrightarrow{X}\ p\ \mathcal{S}\ a\ d$ and one vote $\overleftarrow{X}\ p\ \mathcal{S}\ a\ d$.
(3) For every $S_i \in \mathcal{S}$ and for every $x_j \in S_i$, there is one vote $x_j\ S_i\ a\ p\ \overrightarrow{X \setminus \{x_j\}}\ \mathcal{S} \setminus \{S_i\}\ d$ and one vote $x_j\ S_i\ a\ p\ \overleftarrow{X \setminus \{x_j\}}\ \mathcal{S} \setminus \{S_i\}\ d$.

We also need some voters to control the point balances between $p$ and every $x_i \in X$ and between $p$ and $a$:

(4) 13 votes of the form $a\ p\ \overrightarrow{X}\ \mathcal{S}\ d$ and 13 votes of the form $p\ a\ \overleftarrow{X}\ \mathcal{S}\ d$.
(5) $9s$ votes of the form $\overrightarrow{X}\ a\ p\ \mathcal{S}\ d$ and $9s$ votes of the form $\overleftarrow{X}\ p\ a\ \mathcal{S}\ d$.
(6) For every $x_j \in X$, there are $2s+4$ votes of the form $\overleftarrow{X \setminus \{x_j\}}\ a\ p\ x_j\ d\ \mathcal{S}$ and $2s+4$ votes of the form $x_j\ d\ p\ a\ \overrightarrow{X \setminus \{x_j\}}\ \mathcal{S}$.
(7) 8 votes of the form $\overrightarrow{X}\ p\ a\ \mathcal{S}\ d$ and 8 votes of the form $p\ \overleftarrow{X}\ a\ \mathcal{S}\ d$.
(8) 16 votes of the form $\overrightarrow{X}\ p\ a\ \mathcal{S}\ d$ and 16 votes of the form $a\ d\ p\ \overleftarrow{X}\ \mathcal{S}$.

We have the following point balances between $p$ and the other candidates:

$$
\begin{aligned}
&dist_{(C,V,1)}(p,a) = -(14s+2)+(6s+2)-18s+24s = -26s+24s = -2s,\\
&dist_{(C,V,1)}(p,x_i) = -(7s+1)-(3s+1)-18+3s(9s-3)-(9s-13)(3s+2)-(2s+4) = 2,\\
&dist_{(C,V,1)}(p,S_i) > 6, \text{ and}\\
&dist_{(C,V,1)}(p,d) > 0.
\end{aligned}
$$

**Lemma 4.3.** *In the election constructed in the proof of Theorem 4.8, if a candidate from $C \setminus \mathcal{S}$ is cloned more than once, $p$ and all clones of $p$ lose the election and thus are not in any winning committee.*

*Proof.* Note that whenever we clone a candidate other than $p$, say $c$, the worst-case ordering of the clones (from $p$'s perspective) is that there is one clone of $c$ who is in front of all other clones in every vote. In the following, if we speak of point balances between $c$ and $p$ after cloning candidates, we mean the point balance between the best clone of $p$ and the best clone of $c$.

1. Cloning $x_i \in X$: In the worst case, one clone would gain one point for every vote and $p$ gains one point for only about half of all votes, which means $p$ would lose $6s^2+37s+33$ points on $x_i$. Furthermore, for every $x_j \in X \setminus \{x_i\}$, there are six more votes with $p\ x_j\ x_i$ than votes with $x_i\ x_j\ p$, so $p$ gains 6 points on $x_j$ and $p$ gains 8 points on $a$. Still, the deficit of $p$ to $x_i$ from cloning $x_i$ cannot be caught up on by cloning other candidates without losing to some other candidate.
2. Cloning $a$: It is easy to verify that there are $16s-6$ more votes in which we have $x_i\ a\ p$ than votes in which we have $p\ a\ x_i$ such that $x_i$ gains at least 3 points on $p$ if $a$ is cloned. Also, $p$ loses $6s^2+53s+31$ points on $a$. Therefore, cloning $a$ is never an option.
3. Cloning $d$: There are no votes with $p\ d\ x_i$ but $2s+4$ votes $x_i\ d\ p$ and no votes with $p\ a\ d$ but 16 votes $a\ d\ p$. If $d$ is cloned, $p$ loses $2s+20$ points on $d$. Therefore, it is never beneficial for $p$ to clone $d$.

4. Cloning $p$: Each $x_i$ is in front of $p$ in $6s^2+37s+33 = \frac{|V|}{2} - 6$ votes and $a$ is in front of $p$ in $6s^2+53s+31 = \frac{|V|}{2} + 16s - 8$ votes. If $p$ is cloned $r > 1$ times, we can order the clones of $p$ in such a way that they all have the same number of points. Let $s$ be the score of $p$ in the original election. Then every clone has $s + \frac{|V|}{2}(r-1)$ points.

   Meanwhile, $x_i$ gains $(r-1)\left(\frac{|V|}{2} - 6\right)$ points, and $a$ gains $(r-1)\left(\frac{|V|}{2} + 16s - 8\right)$ points. It follows that $p$ now loses on $a$ with $2s + (16s-8)(r-1)$ points and leads on each $x_i \in X$ with $2+6(r-1)$ points. To decrease $p$'s point deficit on $a$ without $p$ losing to some other candidate, only candidates from $\mathcal{S}$ can be cloned. Still, there need to be at least $s+(8s-4)(r-1)$ additional clones of candidates from $\mathcal{S}$ for $p$ to overtake $a$, but we can only afford $s+3s(r-1)$ clones, for otherwise $p$ would certainly lose to some $x_i$.

This completes the proof. ❑

**Lemma 4.4.** *In the election constructed in the proof of Theorem 4.8, cloning a candidate $S_i \in \mathcal{S}$ twice changes the point balances between $p$ and the other candidates in the following way:*

1. *$p$ loses at most 6 points on both clones of $S_i$,*
2. *$p$ gains 2 points on $a$,*
3. *$p$ loses 2 points on each $x_j \in S_i$,*
4. *$p$ does not gain or lose points on any $x_j \in X \setminus S_i$,*
5. *$p$ gains points on $d$, and*
6. *$p$ never loses points on candidates in $\mathcal{S} \setminus \{S_i\}$.*

*Proof.* Let $S_i \in \mathcal{S}$. The lemma follows from the following observations that each can be easily verified.

1. $p$ is in front of $S_i$ in all but six votes (those in group 3). Therefore, $p$ gains $|V|-6$ votes from the additional clone of $S_i$. Let $s$ be the score of $S_i$ in the original election. Then, in the worst case, a clone of $S_i$ can reach $s+|V|$ points if in every vote one clone is preferred to the other. The score of $p$ in the original election was greater than $s+6$. Therefore, after cloning $S_i$ twice it is now greater than $s+|V|$ and the point difference of $p$ and $S_i$ is reduced by at most 6 points, as it is now only greater than 0 instead of greater than 6 in the election.
2. There are exactly two votes with $p\ S_i\ a$ (in group 2) and no votes with $a\ S_i\ p$.
3. For every $x_j \in S_i$, there are exactly two votes with $x_j\ S_i\ p$ and no votes with $p\ S_i\ x_j$.
4. For every $x_j \in X \setminus S_i$, $S_i$ is never between $p$ and $x_j$ in any vote.
5. There is no vote with $d\ S_i\ p$ but there are several votes with $p\ S_i\ d$.
6. For every $S_j \in \mathcal{S} \setminus \{S_i\}$, there is no vote with $S_i\ S_j\ p$.

This completes the proof. ❑

Equipped with these two lemmas, we can now complete the proof of Theorem 4.8 by showing that $(X, \mathcal{S})$ is a yes-instance of X3C if and only if $(C,V)$ is a yes-instance of 1-Borda-NECESSARY-CLONING-ZC.

From left to right, suppose there is an exact cover $\mathcal{S}'$. Clone every $S_i \in \mathcal{S}'$ twice (i.e., the original candidate $S_i$ is substituted by a clone and there is an additional clone of $S_i$). From Lemma 4.4 and the point balances in the original election it follows that $p$ is now tieing $a$ and every $x_i \in X$ in points and beats every other candidate. Therefore, $p$ is a winner of the election.

From right to left, suppose we can make $p$ a winner of the election by cloning candidates. From Lemma 4.3 it follows that we must clone candidates from $\mathcal{S}$ to make $p$ not lose the election immediately. Adding an additional clone of any $S_i \in \mathcal{S}$ to the election improves $p$'s point balance with $a$ by 2 points and worsens $p$'s point balance with all $x_j \in S_i$ by 2 points. Considering the point balances before cloning any candidates, it follows that we may only clone each $S_i \in \mathcal{S}$ at most twice (which means adding an additional clone of $S_i \in \mathcal{S}$ to the election), as otherwise $p$ would be beaten by all $x_j \in S_i$. Furthermore, we need to add at least $k$ additional clones of candidates from $\mathcal{S}$ for $p$ to at least tie $a$. Therefore, there exists an exact cover of $X$ in $\mathcal{S}$. ❑

Since 1-Borda is equivalent to the single-winner variant of $k$-Borda we also showed that NECESSARY-CLONING-ZC is NP-hard for single-winner Borda. The complexity of this problem was left open by Elkind et al. [1].

## 4.5 *t*-Approval-CC

As winner determination for CC multiwinner voting rules is NP-hard, all considered problems are trivially NP-hard for those rules as well. We will now show, however, that $t$-approval-CC-NECESSARY-CLONING-GC is fixed-parameter tractable when parameterized by the number of either candidates or voters. The following lemma will be helpful in the proofs of Theorems 4.9 and 4.10.

**Lemma 4.5.** *Given a multiwinner election $(C,V,k)$ and a candidate $p \in C$, if we can make $p$ be a member of a winning committee under $t$-approval-CC and for all possible orderings of clones, we can do so by cloning candidates up to $t$ times.*

*Proof.* Assume a candidate was cloned $t$ times.[11] Now, if this candidate is again cloned any number of times, set those clones behind all other clones of this candidate in every vote. Note that because there were $t$ clones of the candidate before we cloned her for the second time, the additional clones would receive zero points from all voters if we order them this way and no other candidate is pushed in or out of the first $t$ positions in any vote (i.e., there is no other candidate or clone that received a point from a voter before the second cloning happened and now receives zero points and vice versa). That means that all committees including only candidates or clones that were present before we added clones a second time have the same score as before the second cloning. Additionally, for any committee, if a candidate or clone of this committee is replaced by one of the new clones, the score of this committee cannot increase. It follows that if there is an order of clones before the second cloning in which $p$ was not part of a winning committee, $p$ is still not part of a winning committee after the second cloning, since the committees' scores do not change and the score of committees including $p$ cannot be raised by replacing committee members with new clones. ❑

**Theorem 4.9.** *For fixed $t \geq 2$, $t$-approval-CC-*NECESSARY-CLONING-GC *is in* FPT *when parameterized by the number of candidates.*

*Proof.* Adapting the FPT algorithm by Bredereck et al. [34] for $t$-approval-CC-SHIFT BRIBERY and using Lemma 4.5 we obtain an FPT algorithm that solves the problem. Given an instance of $t$-approval-CC-NECESSARY-CLONING-GC with $m$ candidates and $n$ voters,

[11] If the candidate was cloned fewer than $t$ times, then we can add more clones such that there are exactly $t$ clones and there would always be an order of the clones such that $p$ is not part of a winning committee.

iterate over all possible cloning vectors $(K_1,\dots,K_m)$ with $K_i \leq t$ for all $1 \leq i \leq m$ that are feasible within the budget $B$. For each such cloning vector, iterate over all committees $W$ in a cloned multiwinner election via $K$ that preclude $p$ or any clone of $p$. For each combination of cloning vector $K$ and committee $W$, solve the following integer linear program (ILP). Let $m' \leq mt$ be the number of candidates in a cloned multiwinner election via $K$. There are $m!$ different types of votes in the original multiwinner election and $m'!$ different types of votes in any cloned multiwinner election via $K$. We order them arbitrarily and associate with each $i \in [m!]$ and each $j \in [m'!]$ the $i$-th and $j$-th vote type of the original and cloned multiwinner election, where $[a]$ is the set of integers less than or equal to an integer $a$. We then create an integer variable $S_{i,j}$ for each pair of vote types. $S_{i,j}$ represents the number of votes that had type $i$ in the original multiwinner election and then have type $j$ in the cloned multiwinner election after all partial votes were extended to complete votes. With $n_i$ being the number of votes of type $i$ in the original multiwinner election, we create the constraint

$$\sum_{j\in[m'!]} S_{i,j} = n_i \quad \text{for every } i \in [m!] \tag{1}$$

to ensure that the number of votes stays the same in the cloned election. Next, we introduce a constraint

$$\sum_{i\in[m!],j\in[m'!]} S_{i,j} \cdot \mathrm{feas}(i,j) = 0 \tag{2}$$

that ensures that it is possible to transform a vote of type $i$ in the original multiwinner election to a vote of type $j$ in the cloned multiwinner election. Here, we use a boolean variable $\mathrm{feas}(i,j)$, which is zero if a vote of type $i \in [m!]$ can be transformed to a vote of type $j \in [m']$, and is one otherwise. We now create integer variables $N_j$ for each $j \in [m'!]$ that describe the number of votes of type $j$ in the cloned multiwinner election:

$$\sum_{i\in[m!]} S_{i,j} = N_j. \tag{3}$$

Then we have to make sure that the committee $W$ beats all committees that contain $p$ or clones of $p$. For a committee $C'$ and vote type $i$ in the cloned multiwinner election, denote by $\omega(i,C')$ the score that a vote of type $i$ assigns to the committee $C'$. Then, for each committee $W'$ containing $p$ or clones of $p$, we create the constraint:

$$\sum_{i\in[m'!]} \omega(i,W) \cdot N_i > \sum_{i\in[m'!]} \omega(i,W') \cdot N_i. \tag{4}$$

The ILP tells us if there is any ordering of clones such that $W$ beats every committee containing $p$ or clones of $p$. If the ILP is not solvable for every committee $W$, there is a cloning vector such that in every cloned multiwinner election via this cloning vector, there always is a committee containing $p$ or a clone of $p$ among the winning committees for all orderings of clones, so output accept.

If we have iterated over all cloning vectors and there always is some ordering of clones such that a committee not containing $p$ or clones of $p$ beats all committees containing $p$ or clones of $p$ in a cloned multiwinner election, output reject.

Due to Lemma 4.5, we only need to check cloning vectors in which every component is at most $t$. Additionally, $\text{feas}(i,j)$ and $\omega(i,C')$ can be precomputed in FPT before the ILP is solved. Regarding the runtime, the ILP will be called at most $t^m \cdot 2^{mt}$ times and can be solved in FPT due to the famous result by Lenstra [35] (which was improved by Kannan [36] and by Fredman and Tarjan [37]) that ILPs can be solved in FPT with respect to the number of integer variables as the parameter. ❑

In the two upcoming proofs, which consider the number of voters as a parameter, we will use the following result due to Betzler et al. [26].

**Lemma 4.6** (Betzler, Slinko, and Uhlmann [26])**.** *Given a multiwinner election with $m$ candidates and $n$ voters, the winning committees under $t$-approval-CC can be computed in time $O(2^m \cdot nm)$ or $O(n^n \cdot poly(m,n))$ with $poly(m,n)$ being some polynomial only dependent on $m$ and $n$.*

Now, we show that $t$-approval-CC-NECESSARY-CLONING-GC is fixed-paramater tractable when parameterized by the number of voters.

**Theorem 4.10.** *For fixed $t \geq 2$, $t$-approval-CC-*NECESSARY-CLONING-GC *is in* FPT *when parameterized by the number of voters.*

*Proof.* We provide an FPT algorithm that solves the problem: Given an instance of $t$-approval-CC-NECESSARY-CLONING-GC with $m$ candidates and $n$ voters, iterate over all possible cloning vectors $(K_1,\dots,K_m)$ with $K_i \leq t$ if $c_i$ is in the first $t$ positions in at least one vote, and else $K_i = 0$, $1 \leq i \leq m$, that are feasible within the budget $B$. For each such cloning vector, iterate over all possible orders of clones and use Lemma 4.6 to check if $p$ is part of a winning committee. If for any order $p$ is not part of a winning committee, continue with the next scoring vector. If all possible orders of clones have been checked and $p$ is part of a winning committee in all of them, accept the input. If all cloning vectors have been checked and none of them led to acceptance of the input, reject the input.

Correctness follows from Lemma 4.5 and the fact that candidates that are never in the first $t$ positions in any vote are irrelevant for the multiwinner election, as they (or their clones) will never contribute points to the score of a committee. Additionally, by cloning those irrelevant candidates, the score of committees involving other candidates will not be changed.

Regarding the runtime, there are at most $nt$ relevant candidates leaving $t^{nt}$ scoring vectors that will be checked by the algorithm. Then there are at most $nt^2$ candidates in the multiwinner election after the cloning and at most $(((nt)^2)!)^n$ different orders of clones. The runtime inferred from Lemma 4.6 then takes time at most $O(n^n \cdot \text{poly}(n,nt^2))$. Overall, we have a runtime of $O(t^{nt} \cdot (((nt)^2)!)^n \cdot n^n \cdot \text{poly}(n,nt^2))$. ❑

Next, we turn to $t$-approval-CC-POSSIBLE-CLONING-GC. We cannot use Lemma 4.5 for this problem, as it may be necessary to clone a candidate more than $t$ times, since the order of clones may be chosen freely.

**Example 4.1.** *Let $t = 1$ (i.e., we consider 1-approval-CC), $C = \{p,c_1,c_2\}$ and $V$ consist of the following voters:*

- *one vote $p\ \cdots$,*
- *$n_1$ votes $c_1\ \cdots$ for some $n_1 > 1$, and*
- *$n_2$ votes $c_2\ \cdots$ for some $n_2 > 1$.*

*If $k = 2$, we can make $p$ be part of a winning committee only by cloning $c_1$ at least $n_1 > t$ times or $c_2$ at least $n_2 > t$ times and by assigning a different clone of $c_1$ (respectively, of $c_2$) to the top position of each of her first-ranked votes.*

However, with the notion of *relevant candidates* we can show that the problem is in FPT when it is parameterized by the number of voters.

**Theorem 4.11.** *For fixed $t \geq 2$, $t$-approval-CC-*POSSIBLE-CLONING-GC *is in* FPT *when parameterized by the number of voters.*

*Proof.* We provide an FPT algorithm that solves the problem.

Given an instance of $t$-approval-CC-POSSIBLE-CLONING-GC with $m$ candidates and $n$ voters, iterate over all possible cloning vectors $(K_1, \dots, K_m)$ that are feasible within the budget $B$ with $K_i \leq 1$ if the $i$-th candidate is irrelevant for the multiwinner election, and $K_i \leq nt$ otherwise. For each such cloning vector, iterate over all possible ordering of clones and use Lemma 4.6 to check if $p$ is part of a winning committee. If the answer is yes for any cloning vector and order of clones, accept the input, and if this is never the case, reject the input.

Correctness follows from the fact that cloning a candidate beyond $nt$ times produces irrelevant clones (i.e., if there are more than $nt$ clones of a candidate for every ordering of the clones, there is at least one clone that is never in the top $t$ positions of any vote) and irrelevant candidates or clones of irrelevant candidates do not contribute to the score of a committee and do not influence the score of committees they are not part of. It follows that the algorithm checks all cloning vectors and orders of clones that may lead to a successful cloning.

Regarding the runtime, there are at most $nt$ relevant candidates and thus at most $(nt)^{nt}$ cloning vectors that the algorithm iterates over. Furthermore, each cloning vector produces at most $(nt)^2$ clones of candidates, so there are at most $(((nt)^2)!)^n$ possible orderings of clones (again, this is a very loose bound). With at most $(nt)^2$ clones of candidates, using Lemma 4.6 then takes time at most $O(n^n \cdot \text{poly}(n, (nt)^2))$. Overall, the algorithm has a runtime of $O((nt)^{nt} \cdot (((nt)^2)!)^n \cdot n^n \cdot \text{poly}(n, (nt)^2))$. ❑

## 4.6 Borda-CC

We will show that Borda-CC-POSSIBLE-CLONING-GC is $W[1]$-hard even for committees of size $k = 1$ (in which case Borda-CC is just single-winner Borda) when parameterized by the number of voters.

**Theorem 4.12.** *Borda-CC-*POSSIBLE-CLONING-GC *is $W[1]$-hard when parameterized by the number of voters, even if the committee size is one and there are only two different values of costs.*

*Proof.* We prove $W[1]$-hardness by providing a parameterized reduction from the problem MULTICOLORED-INDEPENDENT-SET: Given an undirected graph $G = (V(G), E(G))$, an integer $f$ , and a partition of $V(G)$ into $f$ sets $W_1, \dots, W_f$, does there exist an independent set $X \subseteq V(G)$ (i.e., the induced subgraph of $G$ restricted to $X$ has no edges) that contains exactly one vertex of every set $W_i$, $1 \leq i \leq f$? Multicolored-Independent-Set is $W[1]$-hard when parameterized by the number of colors [29].

Let $(G, f, (W_1, \dots, W_f))$ be a MULTICOLORED-INDEPENDENT-SET instance. We may assume that the number of vertices for each color is the same (so $|V(G)| = \ell \cdot f$ for some $\ell \geq 1$) and that there are no edges between vertices with the same color. For $v \in V(G)$, denote by $E(v)$ the set of edges incident to $v$ and by $d(v)$ the degree of $v$. For each color $i$, $1 \leq i \leq f$, denote by $\delta(i)$ the sum of degrees of vertices with color $i$, and let $\Delta = \sum_{1 \leq i \leq f} \delta(i)$.

From $(G, f, (W_1, \ldots, W_f))$ we will now construct a Borda-CC-POSSIBLE-CLONING-GC instance. Let $C = \{p\} \cup V(G) \cup E(G) \cup H \cup D_1 \cup D_2$ with $H = \{h_1, \ldots, h_f\}$ and $D_1$ and $D_2$ being sets of dummy candidates whose sizes we will define later, where $p$ is the distinguished candidate we want to make part of a winning committee.

For a color $i$, $1 \leq i \leq f$, let $W_i = \left\{v_1^{(i)}, \ldots, v_\ell^{(i)}\right\}$, and for a subset $X \subseteq V(G)$, let $G \setminus X$ be the graph $G$ without vertices from $X$ (and without edges incident to vertices from $X$). Define $V$ to consist of the following votes:

(1) For every color $i$, with $1 \leq i \leq f$ , there are two votes:

$$p\ h_i\ \overrightarrow{E(v_1^{(i)})}\ v_1^{(i)}\ \cdots\ \overrightarrow{E(v_\ell^{(i)})}\ v_\ell^{(i)}\ \overrightarrow{E(G \setminus W_i)}\ \overrightarrow{V(G) \setminus W_i}\ \overrightarrow{H \setminus \{h_i\}}\ D_2\ D_1,$$

$$p\ h_i\ \overleftarrow{E(v_\ell^{(i)})}\ v_\ell^{(i)}\ \cdots\ \overleftarrow{E(v_1^{(i)})}\ v_1^{(i)}\ \overleftarrow{E(G \setminus W_i)}\ \overleftarrow{V(G) \setminus W_i}\ \overleftarrow{H \setminus \{h_i\}}\ D_2\ D_1.$$

(2) Further, there are two votes: $p\ \overrightarrow{H}\ \overrightarrow{D_2}\ \overrightarrow{E(G)}\ \overrightarrow{D_1}\ \overrightarrow{V(G)}$ and $\overleftarrow{E(G)}\ \overleftarrow{D_1}\ \overleftarrow{H}\ \overleftarrow{D_2}\ p\ \overleftarrow{V(G)}$.

To determine the number of dummy candidates needed, let us consider the point balances between $p$ and candidates $h_i \in H$ and $e_j \in E(G)$ from the votes in the first group:

$$dist(p, h_i) = 2 + (f-1)(2(E(G) + V(G)) + f + 2),$$
$$dist(p, e_j) = 4 + 2(\ell - 1) + (f-2)(2\ell + E(G) + 3) + \Delta.$$

Then we set $D_2$ to contain $dist(p, h_i) + 2(f-1)$ candidates and $D_1$ to contain $dist(p, e_j) + 2(f-2) + 1$ candidates. Let $B = f$ be the budget. Regarding the cost functions, for every $v \in V(G)$, let the cost of cloning $v$ twice be one, let the cost of cloning $v$ more than twice be $B+1$, and let the cost of cloning any other candidate more than once be $B+1$. Finally, let $k = 1$ be the committee size. It is easy to see that we will only need to worry about the scores of $p$, of candidates from $H$, and of candidates from $E(G)$, since $p$ beats all other candidates even if candidates from $V(G)$ are cloned. For $h_i \in H$ and $e_j \in E(G)$, $p$ is trailing behind $h_i$ with $2(f-1)$ points and behind $e_j$ with $2(f-2)+1$ points. We will now show that $(G, f, (W_1, \ldots, W_f))$ is a yes-instance of MULTICOLORED-INDEPENDENT-SET if and only if the above constructed instance of Borda-CC-POSSIBLE-CLONING-GC is a yes-instance.

From left to right, suppose there is multicolored independent set $X \subseteq V(G)$. Clone every $v \in X$ twice (i.e., the original candidate $v$ is substituted by a clone and there is an additional clone of $v$). Let $i$ be the color of a $v \in X$ (i.e., $v \in W_i$). From the additional clone of $v$, $p$ gains two points on every candidate $H \setminus \{h_i\}$. Since $|V'| = h$ and each candidate in $X$ has a different color, $p$ is now tied with every candidate in $H$. Since the vertex candidates cloned are an independent set for each $e = \{v, v'\}$, at least one of $v$ and $v'$ (say, $v$) was not cloned. If $v$ is of color $i$ then there is another vertex candidate of color $i$ that was cloned (since $X$ contains a vertex of every color), so $p$ gained one point on $e$, and from the cloned vertex candidates that were not of the colors of $v$ and $v'$ candidate $p$ gained $2(f-2)$ points, so $p$ at least ties $e$. Therefore, $p$ now ties or beats all candidates from $H$ and $E(G)$ and wins the multiwinner election (i.e., is in the winning committee of size one).

From right to left, suppose there is no multicolored independent set. We can clone at most $f$ vertex candidates twice. They must be of different colors each and we need to clone $f$ vertex candidates, or else $p$ cannot beat all candidates from $H$. Let us analyze how a cloned vertex candidate $v \in V(G)$ with color $i$ affects the points balance between $p$ and the edge candidates in $E(G)$:

1. $p$ gains zero points on edge candidates in $E(v)$,
2. $p$ gains one point on edge candidates who were incident to vertices of $W_i \setminus \{v\}$ in $G$, and
3. $p$ gains two points on all other edge candidates.

Since there is no multicolored independent set of size $f$, in each $X \subseteq V(G)$ with $|X| = f$ and each $v \in X$ having a different color, there must be $v, v' \in X$ such that $e = \{v, v'\} \in E(G)$. Assume the candidates in $X$ were cloned twice. Since $v$ and $v'$ were cloned and no other candidate with the colors of $v$ and $v'$ were cloned, $p$ could not gain any points on $e$ from the cloning of $v$ and $v'$. Although $p$ gains $2(f-2)$ points on $e$ from the cloning of candidates $X \setminus \{v, v'\}$, $e$ still beats $p$ by one point. So, $p$ cannot win the multiwinner election (and thus is not in any size-one winning committee). . ❑

Since in the reduction above the ordering of clones did not matter, the following holds as well.

**Corollary 4.3.** *Borda-CC-*NECESSARY-CLONING-GC *is $W[1]$-hard when parameterized by the number of voters, even if $k = 1$.*

# 5 Conclusions and Open Problems

We have initiated the study of cloning in various well-known multiwinner elections. Our complexity results are summarized in Table 1. They imply that cloning is intractable in general and is tractable only for simple multiwinner voting rules (such as SNTV) or a few restricted cases (e.g., $k$-Borda-POSSIBLE-CLONING-ZC/UC). Studying the parameterized complexity of the related problems might be fruitful since cloning for more involved voting rules (such as $t$-approval-CC) can be fixed-parameter tractable, even though that is not necessarily so (e.g., not for Borda-CC).

There are a number of interesting open problems (specified by question marks in Table 1). Specifically, the parameterized complexity of possible cloning in $t$-approval-CC for the number of candidates (rather than the number of voters) remains open in all cost models, and so does that of possible and necessary cloning in Borda-CC in the zero-cost and unit-cost models for the number of voters and in all cost models for the number of candidates. Of course, there are many more multiwinner voting rules than those studied here (see, e.g., the book chapters by Baumeister et al. [7] and Faliszewski et al. [8] for overviews and more background), and we propose to extend to them the study initiated here.

Further possible research directions are to study additional cost models such as all-or-nothing cost-functions, as was done by Bredereck et al. [38] for shift bribery, and to further explore the parameterized complexity for problems that are NP-hard.

**Acknowledgments.** We thank the anonymous AAMAS'20 reviewers for helpful comments. We gratefully acknowledge that this work was supported in part by DFG grants RO 1202/14-2 and RO 1202/21-2 (project number 438204498).

## Declarations

**Funding:** This work was supported in part by Deutsche Forschungsgemeinschaft under grants RO 1202/14-2 and RO 1202/21-2 (project number 438204498).

**Non-financial interests:** Author Jörg Rothe currently is or has been on the following editorial boards of scientific journals:

- *Annals of Mathematics and Artificial Intelligence* (AMAI), Associate Editor, since 01/2020,
- *Journal of Artificial Intelligence Research* (JAIR), Associate Editor, 09/2017–08/2023,
- *Mathematical Logic Quarterly* (MLQ – Wiley), Editorial Board, 01/2008–12/2019, and
- *Journal of Universal Computer Science* (J.UCS), Editorial Board, since 01/2005.

**Availability of data and materials:** Data sharing is not applicable to this article as no datasets were generated or analyzed during the current study.

**Conflict of Interest:** The authors declare that they have no conflict of interest.

## References

[1] Elkind, E., Faliszewski, P., Slinko, A.: Cloning in elections: Finding the possible winners. Journal of Artificial Intelligence Research **42**, 529–573 (2011)

[2] Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.): Handbook of Computational Social Choice. Cambridge University Press, Cambridge, UK (2016)

[3] Rothe, J. (ed.): Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, 2nd edn. Classroom Companion: Economics. Springer, Cham, Switzerland (2024)

[4] Conitzer, V., Walsh, T.: Barriers to manipulation in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.) Handbook of Computational Social Choice, pp. 127–145. Cambridge University Press, Cambridge, UK (2016). Chap. 6

[5] Baumeister, D., Rothe, J.: Preference aggregation by voting. In: Rothe, J. (ed.) Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, 2nd edn. Classroom Companion: Economics, pp. 233–367. Springer, Cham, Switzerland (2024). Chap. 4

[6] Faliszewski, P., Rothe, J.: Control and bribery in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.) Handbook of Computational Social Choice, pp. 146–168. Cambridge University Press, Cambridge, UK (2016). Chap. 7

[7] Baumeister, D., Faliszewski, P., Rothe, J., Skowron, P.: Multiwinner voting. In: Rothe, J. (ed.) Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, 2nd edn. Classroom Companion: Economics, pp. 403–465. Springer, Cham, Switzerland (2024). Chap. 6

[8] Faliszewski, P., Skowron, P., Slinko, A., Talmon, N.: Multiwinner voting: A new challenge for social choice theory. In: Endriss, U. (ed.) Trends in Computational Social Choice, pp. 27–47. AI Access Foundation, aiaccess.org (2017). Chap. 2

[9] Lu, T., Boutilier, C.: Budgeted social choice: From consensus to personalized decision making. In: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, pp. 280–286. AAAI Press/IJCAI, www.ijcai.org (2011)

[10] Elkind, E., Faliszewski, P., Skowron, P., Slinko, A.: Properties of multiwinner voting rules. Social Choice and Welfare **48**(3), 599–632 (2017)

[11] Skowron, P., Faliszewski, P., Lang, J.: Finding a collective set of items: From proportional multirepresentation to group recommendation. Artificial Intelligence **241**, 191–216 (2016)

[12] Ghosh, S., Mundhe, M., Hernandez, K., Sen, S.: Voting for movies: The anatomy of recommender systems. In: Proceedings of the 3rd Annual Conference on Autonomous Agents, pp. 434–435. ACM Press, dl.acm.org (1999)

[13] Tideman, N.: Independence of clones as a criterion for voting rules. Social Choice and Welfare **4**(3), 185–206 (1987)

[14] Zavist, T., Tideman, N.: Complete independence of clones in the ranked pairs rule. Social Choice and Welfare **6**(2), 167–173 (1989)

[15] Schulze, M.: A new monotonic, clone-independent, reversal symmetric, and Condorcet-consistent single-winner election method. Social Choice and Welfare **36**(2), 267–303 (2011)

[16] Conitzer, V.: Anonymity-proof voting rules. In: Proceedings of the 4th International Workshop on Internet & Network Economics. Lecture Notes in Computer Science, vol. 5385, pp. 295–306. Springer, Heidelberg and Berlin, Germany (2008)

[17] Aziz, H., Bachrach, Y., Elkind, E., Paterson, M.: False-name manipulations in weighted voting games. Journal of Artificial Intelligence Research **40**, 57–93 (2011)

[18] Rey, A., Rothe, J.: False-name manipulation in weighted voting games is hard for probabilistic polynomial time. Journal of Artificial Intelligence Research **50**, 573–601 (2014)

[19] Baumeister, D., Bouveret, S., Lang, J., Nguyen, N., Nguyen, T., Rothe, J., Saffidine, A.: Positional scoring-based allocation of indivisible goods. Journal of Autonomous Agents and Multi-Agent Systems **31**(3), 628–655 (2017)

[20] Kuckuck, B., Rothe, J.: Duplication monotonicity in the allocation of indivisible goods. AI Communications **32**(4), 253–270 (2019)

[21] Ayadi, M., Ben Amor, N., Lang, J., Peters, D.: Single transferable vote: Incomplete

knowledge and communication issues. In: Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems, pp. 1288–1296. IFAAMAS, www.ifaamas.org (2019)

[22] Elkind, E., Faliszewski, P., Slinko, A.: Clone structures in voters' preferences. In: Proceedings of the 13th ACM Conference on Electronic Commerce, pp. 496–513. ACM Press, dl.acm.org (2012)

[23] Conitzer, V., Rognlie, M., Xia, L.: Preference functions that score rankings and maximum likelihood estimation. In: Proceedings of the 21st International Joint Conference on Artificial Intelligence, pp. 109–115. AAAI Press/IJCAI, www.ijcai.org (2009)

[24] Chamberlin, J., Courant, P.: Representative deliberations and representative decisions: Proportional representation and the Borda rule. The American Political Science Review **77**(3), 718–733 (1983)

[25] Procaccia, A., Rosenschein, J., Zohar, A.: On the complexity of achieving proportional representation. Social Choice and Welfare **30**(3), 353–362 (2008)

[26] Betzler, N., Slinko, A., Uhlmann, J.: On the computation of fully proportional representation. Journal of Artificial Intelligence Research **47**, 475–519 (2013)

[27] Papadimitriou, C.: Computational Complexity, 2nd edn. Addison-Wesley, Reading, MA, USA (1995)

[28] Rothe, J.: Complexity Theory and Cryptology. An Introduction to Cryptocomplexity. EATCS Texts in Theoretical Computer Science. Springer, Heidelberg and Berlin, Germany (2005)

[29] Downey, R., Fellows, M.: Parameterized Complexity, 2nd edn. Springer, Heidelberg and Berlin, Germany (2013)

[30] Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford, UK (2006)

[31] Karp, R.: Reducibility among combinatorial problems. In: Miller, R., Thatcher, J., Bohlinger, J. (eds.) Complexity of Computer Computations, pp. 85–103. Plenum Press, New York, NY, USA (1972)

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

[33] Gonzalez, T.: Clustering to minimize the maximum intercluster distance. Theoretical Computer Science **38**, 293–306 (1985)

[34] Bredereck, R., Faliszewski, P., Niedermeier, R., Talmon, N.: Complexity of shift bribery in committee elections. In: Proceedings of the 30th AAAI Conference on Artificial Intelligence, pp. 2452–2458. AAAI Press, Palo Alto, CA, USA (2016)

[35] Lenstra Jr., H.: Integer programming with a fixed number of variables. Mathematics of Operations Research **8**(4), 538–548 (1983)

[36] Kannan, R.: Minkowski's convex body theorem and integer programming. Mathematics of Operations Research **12**(3), 415–440 (1987)

[37] Fredman, M., Tarjan, R.: Fibonacci heaps and their uses in improved network optimization algorithms. Journal of the ACM **34**(3), 596–615 (1987)

[38] Bredereck, R., Chen, J., Faliszewski, P., Nichterlein, A., Niedermeier, R.: Prices matter for the parameterized complexity of shift bribery. Information and Computation **251**, 140–164 (2016)

# Chapter 6

# Conclusion

In this last chapter we provide an overview of the results presented in this thesis as well as some possible directions for future work. Overall, we studied the computational complexity of many voting rules, placing special emphasis on control by replacing and on multiwinner elections.

Firstly, in Chapter 3, we provided proofs for multiple open cases of control for a range of single-winner voting rules. This even included new results for voting rules as well-studied as Condorcet, and the first set of results for Plurality with Runoff and Veto with Runoff. We also placed special focus on control by replacing candidates or voters – a model that had not been studied as much so far. However, it might be an important model in practice, since replacing can be a subtle way for the chair to influence the election without changing its size. For all of this, we showed that many problems either belong to P or are NP-complete. These results may help differentiate the many available voting rules, as higher computational complexity for these problems might indicate greater safety against malicious actions from the chair. This line of research can be continued in multiple directions. There are still some open cases regarding the complexity of control in the literature – for example, partitioning of either candidates or voters in both Plurality with Runoff and Veto with Runoff. Since our results cover the worst-case complexity of these problems, it might also be interesting to conduct a more practical analysis to see whether these results hold in practice.

Next, in Chapter 4, we extended this line of research to multiwinner voting. We built upon the results by Meir et al. [24], but applied some changes to their model. One reason for this was to better align our work with the existing literature on control in single-winner elections. To do this, we modeled our decision problems similarly to those in the single-winner setting. More importantly we decided to restrict the target committee size to be larger than one. This ensures that the multiwinner voting rules always select multiple winners. As a consequence, the computational complexity of these multiwinner rules can no longer be directly inferred from their single-winner counterparts. Therefore, we provided new proofs showing the complexity of these problems for two of the most important multiwinner voting rules: SNTV and Bloc voting. For these two rules, we covered all possible cases of control by adding, deleting, and – importantly – replacing either candidates or voters. For future work, it would be interesting to investigate the complexity of these control problems for additional multiwinner voting rules. Of special interest, for example, is the family of Chamberlin-Courant rules, for which an analysis of their parameterized complexity in this context might be fruitful. Furthermore, there is still little literature on control by partitioning voters or candidates in multiwinner voting.

Lastly, in Chapter 5, we transferred another kind of control problem to multiwinner voting: cloning candidates. We adapted the model of cloning candidates for single-winner voting by Elkind, Faliszewski, and Slinko [9] and included multiple cost models in our version. We proofed the complexity of these new problems for several well-known multiwinner voting rules. Additionally, we studied the parameterized complexity for some of these rules where the problem is NP-hard. There are still some open cases in this area, which could be an interesting topic for future research. There is also the possibility to apply new cost models to our problem.

In summary, this thesis expanded the known results on the computational complexity of electoral control and also opened up new directions for this kind of research. It placed special emphasis on control by replacing and on multiwinner voting – areas that may prove to be just as important in practice as single-winner voting, yet are still less studied. Therefore, this thesis leaves many interesting problems to be investigated in the future.

# Bibliography


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

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

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

[4] D. Baumeister, P. Faliszewski, J. Rothe, and P. Skowron. Multiwinner voting. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Classroom Companion: Economics, chapter 6, pages 403–465. Springer-Verlag, 2nd edition, 2024.

[5] D. Baumeister and J. Rothe. Preference aggregation by voting. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Classroom Companion: Economics, chapter 4, pages 233–367. Springer-Verlag, 2nd edition, 2024.

[6] J. Carlson, A. Jaffe, and A. Wiles. *The millennium prize problems*. American Mathematical Soc., 2006.

[7] J. Chamberlin and P. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *The American Political Science Review*, 77(3):718–733, 1983.

[8] S. Cook. The complexity of theorem-proving procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*, pages 151–158. ACM Press, 1971.

[9] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections: Finding the possible winners. *Journal of Artificial Intelligence Research*, 42:529–573, 2011.

[10] E. Elkind, P. Faliszewski, and A. Slinko. Clone structures in voters' preferences. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 496–513. ACM Press, June 2012.

[11] G. Erdélyi, M. Neveling, C. Reger, J. Rothe, Y. Yang, and R. Zorn. Towards completing the puzzle: Complexity of control by replacing, adding, and deleting candidates or voters. *Journal of Autonomous Agents and Multi-Agent Systems*, 35(2):41:1–41:48, 2021.

[12] G. Erdélyi, C. Reger, and Y. Yang. Towards completing the puzzle: Solving open problems for control in elections. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 846–854. IFAAMAS, May 2019.

[13] P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7, pages 146–168. Cambridge University Press, 2016.

[14] R. Gandy. Church's thesis and principles for mechanisms. In *Studies in Logic and the Foundations of Mathematics*, volume 101, pages 123–148. Elsevier, 1980.

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

[16] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

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

[18] G. Karh Bet, J. Rothe, and R. Zorn. Complexity of candidate control for single nontransferable vote and bloc voting. In *International Conference on Algorithmic Decision Theory*, pages 3–17. Springer, 2024.

[19] G. Karh Bet, J. Rothe, and R. Zorn. Complexity of control for single nontransferable vote and bloc voting. *Annals of Mathematics and Artificial Intelligence*, (Submitted).

[20] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[21] L. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973. In Russian. English translation in *Problems of Information Transmission*, 9:265–266, 1973.

[22] A. Loreggia. *Iterative Voting, Control and Sentiment Analysis*. PhD thesis, University of Padova, 2016.

[23] A. Loreggia, N. Narodytska, F. Rossi, B. Venable, and T. Walsh. Controlling elections by replacing candidates or votes (extended abstract). In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 1737–1738. IFAAMAS, May 2015.

[24] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.

[25] M. Neveling and J. Rothe. The complexity of cloning candidates in multiwinner elections. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 922–930. IFAAMAS, May 2020.

[26] M. Neveling, J. Rothe, and R. Zorn. The complexity of controlling Condorcet, fallback, and *k*-veto elections by replacing candidates or voters. In *Proceedings of the 15th International Computer Science Symposium in Russia*, pages 314–327. Springer-Verlag *Lecture Notes in Computer Science #12159*, June/July 2020.

[27] M. Neveling, J. Rothe, and R. Zorn. The complexity of cloning candidates in multiwinner elections. *Journal of Autonomous Agents and Multi-Agent Systems*, (Submitted).

[28] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.

[29] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[30] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[31] N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[32] A. Turing. On computable numbers, with an application to the entscheidungsproblem. *Proceedings of the London Mathematical Society, Ser. 2*, 42:230–265, 1936.

[33] A. Turing. On computable numbers, with an application to the entscheidungsproblem. a correction. *Proceedings of the London Mathematical Society, Ser. 2*, 43:544–546, 1937.

# Eidesstattliche Erklärung

*entsprechend §5 der Promotionsordnung vom 15.06.2018*

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“erstellt worden ist.

Des Weiteren erkläre ich, dass ich eine Dissertation in der vorliegenden oder in ähnlicher Form noch bei keiner anderen Institution eingereicht habe.

Teile dieser Dissertation wurden bereits in Form von Zeitschriftenartikeln und Konferenzberichten veröffentlicht oder zur Begutachtung eingereicht und sind entsprechend referenziert: [11], [12], [26], [19], [18], [27] und [25].

Düsseldorf, 8. August 2025

Roman Zorn