A Computational Complexity Study of Various Types of Electoral Control, Cloning, and Bribery

Inaugural-Dissertation

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

vorgelegt von
Marc Neveling
aus Wuppertal

Düsseldorf, September 2021

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. Herr Prof. Dr. Jörg Rothe

2. Frau Jun.-Prof. Dr. Dorothea Baumeister

Tag der mündlichen Prüfung: 20. September 2021

Abstract

This thesis deals with computational social choice which combines computational complexity theory, one of the most important areas of theoretical computer science, with social choice theory which is highly relevant to economists, politicians, and other figures involved in decision-making processes. Although being a fairly young research field, springing to life in the early 1990s, computational social choice has established itself as one of the central pillars of artificial intelligence and multi-agent systems research.

The central objects of interest for computational social choice are elections, which model decision-making processes where preferences of different agents or voters over candidates have to be aggregated into a final decision, and voting rules, which specify methods of how to aggregate those preferences. Naturally, all parties involved in an election, e.g., the agents, the organizing chair or even an outside agent, might have an interest to influence the outcome of an election. So-called election tampering attempts take many different forms: The agents may submit insincere preferences (i.e., strategic voting or election manipulation), the organizing chair might alter the structure of the election (i.e., electoral control) or an outside agent might want to bribe agents to change their votes to her liking (i.e., bribery). In this thesis we investigate, from a computational complexity perspective, to what degree elections evaluated by certain voting rules can be influenced by those types of election tampering attempts.

Firstly, we study electoral control for the Borda Count which is one of the oldest and most important voting rules. We consider different types of electoral control such as adding or deleting candidates or voters and in particular electoral control by partitioning the candidates or voters into two groups. Furthermore, we consider so-called online electoral control in which candidates or voters appear one after another in the election and the chair may decide only in the moment of appearance to exert some control action. We find that Borda is rather resistant against electoral control by proving NP-hardness of several control problems.

Secondly, we study replacement control which is a special kind of electoral control in which candidates or voters that are removed from the election need to be replaced by, as of yet, unregistered candidates or voters. We find that the complexity of replacement control problems usually follows the complexity of the corresponding classical control problems. Furthermore, we fill gaps in the literature regarding the complexity of the classical electoral control problems regarding adding or deleting candidates or voters.

Thirdly, we consider multiwinner elections in which we seek to elect a fixed-sized set of candidates, a committee, instead of single candidates. We devise a model and define several decision problems to model electoral control by cloning of candidates. A candidate is cloned by adding a new candidate to the election that is very similar to the original candidate. We study the introduced model for cloning candidates in multiwinner elections for several popular multiwinner voting rules and find a wide range of complexity results.

The last contribution of this thesis deals with shift bribery, which is a special kind of bribery in which only one special candidate may be moved forwards or backwards in the voters' preferences. We study shift bribery for iterative voting rules that decide the outcome of an election in several rounds. We find that iterative voting rules are generally very resistant to shift bribery. In contrast to non-iterative voting rules, for which several examples of vulnerability against shift bribery can be found in the literature, shift bribery is NP-hard for all of our considered iterative voting rules.

Contents

CHAPTER 1

INTRODUCTION

Collective decision-making, the act of aggregating individual preferences of a group of individuals into a final decision, is important in almost every social aspect of life ranging from politics over economics to everyday activities like choosing where to go on vacation. In the wake of digitalization a multitude of additional settings became important including multi-agent systems, meta-search engines or recommendation systems for online multimedia platforms like Youtube or Netflix. In each of those settings there is a set of candidates or alternatives from which we would like to choose one and a set of agents, which we will call voters, with preferences over the candidates. Depending on the specific settings candidates might be politicians to be elected, bills to be ratified, applicants to company positions or objects to be chosen from. The agents might be registered voters, jury or committee members, or even processes running on servers.

The most common way to aggregate individual preferences and come to a collective decision is to run an election: collect the preferences of the voters and use an aggregation procedure, which we will call voting rules, to determine the winning candidate. But, choosing a good voting rule for the job at hand is more intricate than at first glance. Firstly, it is important to consider in what form the voters' preferences are given. There may be ordinal preferences meaning voters order the candidates linearly according to their liking or there may be cardinal preferences in which the voters assign each candidate points or even a mixture of both. Still, even if we focus on one type of preferences, for example ordinal preferences, there is a multitude of possible voting rules we could use to aggregate those preferences. As early as the age of ancient Greece elections were used to elect representatives or settle disputes but the scientific study of elections did not start until the late 18th century when the Marquis de Condorcet [33] started a research field called social choice theory by applying mathematics to voting theory and rigorously formalizing elections and voting rules. With this, general theorems or statements can be deduced and voting rules can be characterized which hopefully helps us choose the right voting rule for the right task.

One of the many famous results of Condorcet's work is the Condorcet paradox. Consider the following linear preferences of three voters over the candidates a, b and c: Voter 1 prefers a to b to c, voter 2 prefers b to c to a, and voter 3 prefers c to a to b. Whichever candidate we choose as the winner there is always another candidate who is preferred by two of the three voters (e.g., if a is chosen as the winner, voter 2 and 3 both prefer candidate c to a) implying that it might not be possible to find a satisfying outcome for an election. Condorcet's preferred method of voting, thus called the "Condorcet method", chooses the candidate as the winner of an election that defeats all other candidates in pairwise comparison. Other famous theorems resulting from this "golden age" of social choice theory are the Condorcet jury theorem [33], the median voter theorem [18], and May's theorem [110]. But the most famous result is certainly due to Arrow [2], who later received a Nobel Memorial Prize in Economic Sciences together with John Hicks. In his so-called impossibility theorem it was shown that no voting rule, which accepts votes that rank all candidates, can satisfy three reasonable criteria simultaneously:[1] (1) If every voter prefers some candidate a over some candidate b, then the voting rule cannot choose b over a, (2) if the voting rule prefers a to b then this is still the case even if candidates other than a and b are removed from the election, and (3) no single voter should be able to decide the outcome. The theorem implies that there does not exist a "perfect" voting rule.

Another method of voting was proposed by Jean-Charles de Borda to elect the members of the French Academy of Science in 1770 [20]. In the so-called "Borda method" candidates score points from every voter depending on where they are placed in the voter's preference. Condorcet and Borda famously argued whose voting method is better with Borda once defending his voting method by proclaiming "My scheme is intended only for honest men" [141]. He is thereby implying that when using his voting rule a voter can benefit by casting a dishonest vote, which is also called strategic voting. Strategic voting should ideally be discouraged since it would give the dishonest voters more influence over the election outcome as the other, honest voters. Alas, Allan Gibbard [78] and Mark Satterthwaite [142] showed independently from each other that a voting rule that is non-dictatorial, meaning that there is no single voter who determines the outcome, and non-imposing, meaning every candidate can possibly win in some election, is necessarily manipulable by strategic voting.[2] Since a fair voting rule should always be non-dictatorial and non-imposing, strategic voting is always possible. This notion of reasonable voting rules being manipulable is further reinforced by the Duggan-Schwartz Theorem [48]. Their theorem deals with non-resolute voting rules which means that they are choosing not a single candidate as the winner but a subset of candidates (i.e., candidates may tie for the win). Duggan and Schwartz show that a non-resolute voting rule that is anonymous (i.e., all voters are treated the same), non-imposing, and where there may be voters whose top ranked candidate is not in the set of winners is necessarily manipulable by strategic voting. The third property is reasonable to assume for a voting rule in order to have a meaningful set of winners since the winning set of a voting rule that includes all top ranked candidates in the set of winners would always be very large.

The celebrated Gibbard-Satterthwaite impossibility theorem inspired John Bartholdi III, Craig Tovey, and Michael Trick in the late 1980s to a series of papers applying computational complexity theory to social choice theory launching a research field called computational social choice that sits at the intersection of economics and computer science. Their approach to combat strategic voters was to choose a voting rule for which it is intractable to decide if an election can be manipulated by strategic voting. Then, under the condition that the election is large enough which is true in most multi-agent settings, it might take a strategic voter too long to decide if the election can be manipulated or to even compute a successful strategic vote [37]. Remarkably, intractability is seen as a positive property for this use case whereas it mostly seen negatively in computational complexity theory (i.e., the problem at hand cannot always be solved quickly). This notion is in many ways similar to how computational complexity was successfully used in cryptography to investigate the vulnerability of cryptosystems to attacks. Ideally, a cryptosystem should be hard to break, which can be achieved if it takes a long time for an attacker to break it [135].

The approach used to combat strategic voting can be extended to other forms of election tampering. Besides voters, other entities involved in elections could try to influence the outcome of an election. An election's chair organizing the election could be interested in steering the election into a certain direction without directly submitting a vote but instead by altering the structure of the election. We call this form of election tampering "electoral control" and it appears in the real world as, e.g., voter

[1] To be precise, we also need to require that there are at least three candidates and the votes are unrestricted in their structure.

[2] Again, we tacitly assume that there are at least three candidates and the votes are not restricted in some way.

disenfranchisement [154], cloning of candidates [150], or Gerrymandering [61, 91]. For voter disenfranchisement regulations are introduced to prevent groups of voters from voting that would vote contrary to the chair's liking. Felony disenfranchisement is the most common form of voter disenfranchisement in which people with criminal convictions are excluded from voting. In the US it is believed that felony disenfranchisement heavily influenced the presidential election in 2000 [105]. By cloning a candidate, i.e., introducing additional candidates to the election that are similar to an already participating candidate, the chair can split up the support of the cloned candidate. This type of electoral control is also known as the "strategic candidacy problem" [49]. Gerrymandering is used in district-based elections in which voters are partitioned into districts usually constrained by their geographical location. The voters then elect a representative for their district and the representatives of all districts elect the overall winner in a separate election. By manipulating the district borders the chair can reduce the impact votes of a certain group of voters have on the overall election result. This technique was first used by Massachusetts governor Elbridge Gerry in 1812 by creating a salamander shaped district coining the term "Gerrymandering" [61, 91].

Another way to tamper with an election is bribery. Now, an external agent, who is not part of the election, is bribing voters to change their vote in some way that favors the external agent's own preference. Bribery problems come in many different flavors depending on which voters can be bribed and how their votes can be changed [70]. Apart from the obvious example of a malicious briber trying to influence an election, a whole range of real world scenarios can be modeled by bribery problems. The most common scenarios are political campaign management and as a robustness measure for election results.[3] The former is related to bribery in that the campaign manager "bribes" the voters to change their votes by running, e.g., targeted ad campaigns. A special kind of bribery in which only the favorite candidate of the external agent can be moved forward in the votes models a type of ethical campaign management in which the campaign manager is not allowed to apply smear tactics to damage the standing of other candidates. The robustness of an election result is important in the following way. If the winner of the election can be dethroned by only a few changes to the votes, then the supposed winner might be incorrect due to vote counting errors or manipulation attempts.

Beyond election tampering, computational social choice is concerned with winner determination in elections—in particular, with possible and necessary winners [99]—winner prediction [35], and iterative elections [112]. Another sub-field deals with multiwinner elections that seek to elect a fixed-sized set of candidates, a committee, and can be used for, e.g., shortlisting or parliamentary elections [71]. Over time other research fields were integrated and adapted to computational social choice to address settings such as meta-search engines [50, 98], information extraction [145], planning [55], automated scheduling [81], collaborative filtering [128], computational linguistics [124], kidney exchanges [134], and assignments of students to schools [79]. Very recently, special attention was given to participatory budgeting [6, 13, 30] which is concerned with the question of how a budget can be allocated to different projects depending on the preferences of voters.

In this thesis we will extend the pioneering work of Bartholdi, Tovey, and Trick focusing on the line of research concerning electoral control and bribery.

[3] Another way to measure the robustness of election results is the margin of victory [155].

Outline

In Chapter 2 we will provide the background for the following chapters including preliminary definitions and theorems of voting theory and computational complexity theory, and a brief survey of previous related works in computational social choice. In Chapter 3 we investigate how resistant the Borda voting rule is against the classical electoral control defined by Bartholdi, Tovey, and Trick [10] and Hemaspaandra, Hemaspaandra, and Rothe [83]. In Chapter 4 we study electoral control once again solving open problems to try to complete the puzzle of the computational complexity of electoral control for the most popular voting rules. In particular, we study replacement control for which we may only remove a candidate or voter from the election if, in turn, we add an unregistered candidate or voter, respectively. In Chapter 5 we devise a new model, in the context of multiwinner elections, for electoral control by cloning, which is the act of adding candidates to an election that are similar (i.e., clones) to an already participating candidate, and study the computational complexity of this model for various multiwinner voting rules. In Chapter 6 we study shift bribery, which is a special kind of bribery in which only the position of a special candidate can be improved or worsened. For this type of bribery we study how resistant iterative voting rules, which elect the winner of an election in multiple rounds, are against it. Lastly, in Chapter 7 we summarize the previous chapters and provide starting points for future research.

CHAPTER 2

BACKGROUND

In this chapter we will provide background information necessary for the following chapters including definitions, notation, and some important prior work.

For a comprehensive overview of computational social choice see the recent book by Brandt et al. [23], especially the book chapters therein by Conitzer and Walsh [37] and Faliszewski and Rothe [70] concerning manipulation, electoral control, and bribery, and the book by Rothe [136]. A more concise introduction to computational social choice is provided by the surveys of Chevaleyre [31], Faliszewski and Procaccia [69], and Faliszewski, Hemaspaandra, and Hemaspaandra [64]. Additionally, see the very recent survey by Lang [99] of elections with incomplete knowledge. For an introduction to multiwinner elections see the book chapter by Kilgour [95] and the book chapter by Faliszewski et al. [71] surveys recent research from the computational complexity perspective. The books by Arora and Barak [1], Papadimitriou [126], and Rothe [135] deal with complexity theory (see Tovey's tutorial [151] for an introduction). For parameterized complexity theory see the books by Cygan et al. [39], Downey and Fellows [47], Flum and Grohe [76], and Niedermeier [122].

2.1 Computational Complexity

Our main tools for analyzing elections and classifying voting rules originate in a research field called computational complexity theory started by the seminal work of Hartmanis and Stearns [80] in 1965 who use the abstract computational model of Turing machines defined by Alan Turing [152]. Computational complexity theory deals with the question of whether different computational tasks can be solved "efficiently". By efficiently we mean the amount of resources (usually the time it takes to finish or the space that is used) required to complete the computation. Computational tasks are formalized as decision problems[1] that consist of the problem's name, given information (i.e., the input), and a question about the input. For example, the important decision problem SAT [77] is defined as follows.

SATISFIABILITY (SAT)	
Input:	A boolean formula ϕ with a set of (boolean) variables X and a set of clauses K over X.
Question:	Is there a satisfying truth assignment to the variables of ϕ?

[1]Note that there are many different kinds of problems besides decision problems such as optimization problems, search problems, sort problems or counting problems but in this thesis we will only deal with decision problems. Therefore, when we speak of "problems" we always mean "decision problems".

A specific case of a decision problem is called an *instance* (i.e., for SAT an instance would be a specific boolean formula). We call an instance I of a decision problem A a *yes-instance* if the answer to the question of the problem for this instance is "yes" and a *no-instance* otherwise. Sometimes it is useful to think of the decision problem as the set of all yes-instances of the problem so $I \in A$ if and only if I is a yes-instance of A (e.g., $\mathrm{SAT} = \{\phi \mid \phi$ is a satisfiable boolean formula$\}$).

Example 2.1 (Instances of decision problems)**.** Consider the instance (X,K) of SAT with $X = \{x_1, x_2\}$ and $K = \{K_1, K_2\}$ with $K_1 = (x_1 \vee \overline{x_2})$ and $K_2 = (\overline{x_1} \vee \overline{x_2})$. Can we find a truth assignment to x_1 (i.e., x_1 is set to *true* or *false*) and x_2 (i.e., x_2 is set to *true* or *false*) such that both K_1 and K_2 evaluate to *true*? Here, the answer is *yes* since we can set x_1 to *true* and x_2 to *false* to satisfy both clauses. Therefore, (X,K) is a yes-instance of SAT. If K would contain a third clause $K_3 = (x_2)$, we would have a no-instance since we cannot satisfy K_1, K_2, and K_3 at the same time.

Note that the input to a decision problem can be of many forms. In addition to boolean formulas they may have strings, integers, graphs, elections, etc. as inputs. To be able to do computations on different kinds of input we assume—on a lower level—that inputs are encoded in some way, usually as strings over $\{0,1\}$ called the *binary encoding*.

A (deterministic) *algorithm* that solves a decision problem takes any input or instance to the problem (in binary encoding), does some basic computational steps, and outputs the answer to the question of the problem, therefore *deciding* if the instance belongs to the problem or not. Assuming the Church-Turing Thesis [32, 152], that any real-world algorithm can be simulated by Turing machines, holds true we use Turing machines as the computational model to represent algorithms (although, for the sake of readability, we will use a more descriptive language to define algorithms in this thesis).

The type of algorithms defined above are deterministic meaning the steps the algorithm takes for an input are singular and predetermined leading to a *computation path* in which every node is a state of the algorithm. In contrast, a nondeterministic algorithm may choose from several possible steps on how to proceed with the computation on the current state. In particular, the algorithm makes every possible decision (i.e., which computational step to do next) simultaneously building a *computation tree* instead of a path. Then, the nondeterministic algorithm accepts the input if there is at least one path in its computation tree that accepts. Note that some paths on the computations tree of a nondeterministic algorithm may be infinitely long in contrast to deterministic algorithms in which every computation stops eventually. Therefore, nondeterministic algorithms can only accept decision problems and never decide them like deterministic algorithms. Due to the fact that there are no modern computers who can run nondeterministic algorithms (yet) they would have to be simulated by deterministically running through every path in the computation tree if used in practice which is very inefficient. For a deterministic algorithm we define its (worst-case) runtime as the maximum number of steps the algorithm takes over all possible inputs. For a nondeterministic algorithm its (worst-case) runtime is the maximum number of steps of the shortest accepting path over all inputs that the algorithm accepts.[2] Runtimes are given as functions depending on the size of the input that describe the growth of the computational effort of the algorithm with increasing size of the input. For those function we are only interested in the fastest growing factor, called the *runtime bound*, as slower growing factors and constants become irrelevant for larger input sizes (e.g., the values of the functions $f(x) = 2x+3$ and $g(x) = 3x$ for very large x are very similar but much smaller in comparison to values of $h(x) = 2^x$). That means we are interested in the asymptotic behavior,

[2] There are other complexity measures such as space that we will not discuss here.

the asymptotic bounds, of algorithms when the size of the input increases. In this thesis, the most important runtime functions are polynomial and exponential functions. A polynomial function p is defined as $p(x) = c_k x^k + c_{k-1}x^{k-1} + \cdots + c_1 x + c_0$ for some constant k and $k+1$ constants $c_0, \ldots, c_k$, and an exponential function f is defined as $f(x) = 2^{p(x)}$ for some polynomial function p. We say that an algorithm has a polynomial-time runtime if its runtime functions is in $O(p(x))$ for some polynomial function p and an exponential-time runtime if its runtime function is in $O(f(x))$ for some exponential function f.[3]

Complexity Classes

Computational complexity theory aims to group similarly complex problems together into so-called *complexity classes*. The most important complexity classes are P, containing problems that can be solved in deterministic polynomial time, and NP, containing problems that can be solved in nondeterministic polynomial time. In order to show that a decision problem belongs to P we need to find a deterministic polynomial-time algorithm that solves the problem and for NP-membership the algorithm only needs to be nondeterministic.

Example 2.2 (SAT is in NP)**.** To show that SAT is in NP, a nondeterministic algorithm could (nondeterministically) guess a truth assignment for the variables of a given instance of SAT and then check (in polynomial time) if it satisfies the given formula of the instance.

It is widely believed that P represents the class of problems that can be efficiently solved (problems belonging to P are also sometimes called *tractable*) although a polynomial runtime with a large exponent is still very slow for large inputs. But it turns out that many natural problems in P actually have algorithms with polynomial runtime with only small exponents so this assumption seems reasonable. Obviously, it holds that P $\subseteq$ NP since every deterministic algorithm is also nondeterministic with a computation tree that only consists of one path. We already mentioned that nondeterministic algorithms can be simulated by deterministic algorithms with an exponential blow-up in the runtime. The question of whether there exists a deterministic polynomial-time algorithm for every nondeterministic polynomial-time algorithm is formalized as the famous open P $=$ NP problem [38]. It is widely believed that P $\neq$ NP, i.e., there exist problems in NP for which no deterministic polynomial-time algorithm exists and we also require this assumption in this thesis. Another complexity class is coNP which contains the complements of problems that are in NP (i.e., let A be a decision problem and I be an instance of A, then I is in the complement of A if and only if $I \notin A$). We mention in passing that there are other complexity classes that contain problems who are (allegedly) even harder than problems in P, NP or coNP. Together they form the so-called *polynomial hierarchy* introduced by Meyer and Stockmeyer [115] and Stockmeyer [149] but we will not further discuss those complexity classes here.

Showing that a decision problem belongs to a complexity class is in some sense showing an upper bound of the complexity of that problem since then we can follow that the problem can be solved at least as fast as the most difficult problems in that complexity class. In this thesis we aim to find upper bounds of P or NP. In contrast, showing a lower bound of the complexity for a problem is much more

[3]We assume the reader to be familiar with the Big O notation. Roughly, for two functions f and g, f is in $O(g)$ if there is some input to f and g after which f does not grow faster than g (i.e., f is upper bounded by g barring constant factors and finitely many exceptions).

complicated as we would need to show that there does not exist an algorithm with a specific runtime that solves the problem. The notion of reducibility which we will discuss in the following section enables us to proof such lower bounds for decision problems.

Reducibility and Complexity Lower Bounds

We say that a decision problem A (polynomial-time many-one)[4] reduces to another decision problem B (formally, $A \leq_m^p B$) if we can construct in polynomial time from each instance I of A an instance I' of B such that I is a yes-instance of A if and only if I' is a yes-instance of B. Intuitively, if we can reduce A to B, then A is at least as hard to solve as B. Then, a decision problem A is $\leq_m^p$-hard for a complexity class $\mathscr{C}$ (or simply $\mathscr{C}$-hard) if $B \leq_m^p A$ for every $B \in \mathscr{C}$. Furthermore, a decision problem A is $\leq_m^p$-complete for a complexity class $\mathscr{C}$ (or simply $\mathscr{C}$-complete) if A is $\mathscr{C}$-hard and $A \in \mathscr{C}$. The notion of hardness for a complexity class intuitively means that a decision problem is at least as hard to solve as the hardest problems of that complexity class and completeness for a complexity class even implies that the problem is one of the hardest problems in this class. Even still, in order to show hardness with this basic definition we would need to reduce every one of (possibly) infinitely many problems in the complexity class to the problem we want to show hardness for. Luckily, due to the transitivity of $\leq_m^p$ (i.e., for three problems A, B, and C, it holds that if $A \leq_m^p B$ and $B \leq_m^p C$, then $A \leq_m^p C$), if a decision problem B is $\mathscr{C}$-hard for a complexity class $\mathscr{C}$ and $B \leq_m^p A$ for another decision problem A, then A is $\mathscr{C}$-hard as well. Thus, given a $\mathscr{C}$-hard problem we can show $\mathscr{C}$-hardness of another problem by reducing from the $\mathscr{C}$-hard problem to it. Cook [38] showed that SAT is NP-hard opening up the possibility to show NP-hardness (and even NP-completeness) for many other problems by reducing from it. Karp [93] then showed for several natural problems that they are NP-complete. Since P contains the decision problem that can be solved in polynomial time, showing NP-hardness and assuming P $\neq$ NP implies that the problem is not solvable efficiently or that it is intractable. This makes the P $\neq$ NP question central to computational complexity theory as important natural problems (e.g., the problem of creating mathematical proofs) are NP-hard and a collapsing of both complexity classes would mean that they are efficiently solvable. Interestingly, showing for only one NP-complete problem that it is in P would immediately show P-membership of all other NP-complete problems as well and, thus, showing P $=$ NP. Note that NP-hardness of a problem does not mean that specific instances of the problem are hard to solve as the size of a specific instance is predetermined and, therefore, can be solved in constant time. Rather, the NP-hardness of a problem means that, unless P $=$ NP, as instances of the problem increase in size the time to solve those instances grows exponentially.

We can also use reducibility to show upper bounds. Since P, NP, and coNP are closed under $\leq_m^p$-reducibility, for a complexity class $\mathscr{C} \in \{\mathrm{P}, \mathrm{NP}, \mathrm{coNP}\}$ and two decision problems, A and B, with $A \leq_m^p B$ and $B \in \mathscr{C}$ it follows that $A \in \mathscr{C}$.

Parameterized Complexity Theory

Showing NP-hardness is not the end of research but merely the beginning. There are several approaches on how to deal with the NP-hardness of a decision problem and gain more insight into what

[4]There are many other notions of reducibility but in this thesis we will only use polynomial-time many-one reducibility so we will simply call it “reducibility”.

makes the problem hard to solve. Apart from average-case analysis, finding approximate solutions, or using heuristics, studying the parameterized complexity of a problem has seen much attention recently.

The general idea of parameterized complexity theory is to choose some part of the input to an NP-hard decision problem as the parameter and then design an algorithm that runs in polynomial time if the parameter is small or even a constant (it can only run in exponential time in general as the problem is NP-hard, unless P = NP) or proof that such an algorithm probably does not exist. To this end, we turn a decision problem into a *parameterized decision problem* by taking some part of the input as the parameter. An instance of such a parameterized decision problem is a pair (I, p) with I being an instance of the "not-parameterized" problem and p being some part of I.

Example 2.3 (Parameterization of VERTEX COVER)**.** Consider the following decision problem.

VERTEX COVER	
Input:	An undirected graph $G = (V, E)$ with n vertices and m edges and an integer k.
Question:	Does there exist a set $V' \subseteq V$ of at most k vertices of G such that V' covers the edges of G (i.e., for each $\{v_1, v_2\} \in E$, $v_1 \in V'$ or $v_2 \in V'$)?

The obvious parameters of VERTEX COVER are the integer k, the number of vertices n, or the number of edges m. We can also combine parameters by adding them, e.g., we could choose the parameter $k+n$. We may also study parameters that are given by the structure of the instance. For problems that deal with graphs, like VERTEX COVER, it is common to choose the maximum degree of the graph or the treewidth of the graph as parameters.

A parameterized decision problem is in FPT (i.e., it is fixed-parameter tractable) if there exists an algorithm that solves it in $O(f(p) \cdot |I|^{O(1)})$ time for some computable function f. If p is small or a constant, then $|I|^{O(1)}$ is the fastest growing factor of the runtime of an FPT-algorithm which turns the runtime polynomial. The complexity class XP consists of problems that can be solved in $O(f(p) \cdot |I|^{g(p)})$ time for some computable functions f and g. Notice that if the parameter is a constant, then the problem can be solved in polynomial time. Therefore, problems in this complexity class are often called *slice-wise* polynomial-time solvable as instances (I, p) of a parameterized problem for some fixed p can be seen as a *slice* of the problem. In practice XP-algorithms often times do not perform very well as the constant parameter might be relatively large which results in a high degree polynomial as the runtime. In contrast, the degree of the polynomial in the runtime of an FPT-algorithm must be independent of the parameter.

Example 2.4 (VERTEX COVER parameterized by k is in FPT)**.** Since VERTEX COVER is NP-complete (see, e.g., Garey and Johnson [77]) we expect that there is no polynomial-time algorithm that solves the problem. The trivial brute-force algorithm that iterates through all $\binom{n}{k}$ possible sets of vertices of size k and checks whether it is a vertex cover obviously needs exponential time in the worst case ($O(n^k)$ time to be precise) to solve the problem. Notice that if k is a constant, the algorithm runs in polynomial time. Therefore, VERTEX COVER parameterized by k is in XP. On the other hand, the algorithm above does not satisfy the requirements to be a FPT-algorithm. But, we can improve the trivial algorithm in the following way. Firstly, we can remove any vertex from the graph that has no neighbor since we cannot cover any edges with such a vertex. Afterwards, notice that, given an instance (G,k) of VERTEX COVER, if a vertex v of G has at least $k+1$ neighbors, this vertex needs to be in a vertex cover for otherwise the cover would have to include all at least $k+1$ neighbors of v to cover all of v's incident edges. Therefore, we can immediately remove such a vertex (including its incident edges) from the graph and decrease k by one. So, after those two reduction steps all remaining vertices have at most k neighbors but at least one and we can reject the instance if there are more than k^2 edges since every vertex can cover at most k edges. Otherwise, it follows that there are at most $2k^2$ vertices that were not removed in the reduction step and we can brute-force over all (at most) $\binom{2k^2}{k}$ sets of vertices of size k to check whether all edges are covered. This algorithm has a worst case runtime of $O(2^k k^{2k} \cdot m + nm)$ and, therefore, runs in FPT-time if k is the parameter.

The technique shown in Example 2.4 to first reduce the size of an instance which can then be solved in FPT-time is also called *kernelization*. Another technique to show FPT-membership uses integer linear programs. An *integer linear program* (ILP) consists of p variables and a set of m linear inequalities over the variables. Then, the goal of the ILP is to choose values for the variables such that the inequalities are satisfied. Formally, we define the INTEGER LINEAR PROGRAMMING FEASIBILITY problem as follows. We are given a constraint matrix $A \in \mathbb{Z}^{m\times p}$ and a bias vector $b \in \mathbb{Z}^m$. The question is whether there exists a variable vector $x \in \mathbb{Z}^p$ such that $Ax \leq b$. INTEGER LINEAR PROGRAMMING FEASIBILITY is known to be NP-complete but the following theorem was proven by Lenstra [101].

Theorem 2.1 (Lenstra's theorem [101])**.** *An* INTEGER LINEAR PROGRAMMING FEASIBILITY *instance of size L with p variables can be solved in time* $O(p^{2.5p+o(p)} \cdot L)$.

It follows that INTEGER LINEAR PROGRAMMING FEASIBILITY is in FPT if parameterized by p, i.e., the number of variables. Therefore, if we manage to express a parameterized problem as an ILP in which the number of variables is only bounded by the parameter, we can solve this ILP (and subsequently the problem) in FPT-time by using Lenstra's theorem. We illustrate the technique by the following example.

Example 2.5 (An ILP for VERTEX COVER)**.** Given an instance (G,k) of VERTEX COVER, the ILP has a boolean variable x_i for every $v_i \in V$ (i.e., $x_i \in \{0,1\}$ for $1 \leq i \leq n$). If a variable x_i is set to 1, this corresponds to vertex v_i being in the vertex cover. Furthermore, the ILP consists of the following constraints.

$$x_i \geq 0 \qquad \text{for every } v_i \in V \tag{2.1}$$

$$x_i + x_j \geq 1 \qquad \text{for every } \{v_i, v_j\} \in E \tag{2.2}$$

$$\sum_{i=1}^{n} x_i \leq k \tag{2.3}$$

Constraint (2.2) ensures that every edge of the graph is covered and constraint (2.3) ensures that the vertex cover contains at most k vertices. Due to Theorem 2.1 and the fact that we have n variables, we can solve the ILP in FPT-time if n is the parameter and, therefore, have shown that VERTEX COVER is in FPT for this parameter. Note that parameterizing VERTEX COVER by n is not all that useful since the size of the instance is directly related to n (i.e., $k \leq n$ and $m \leq \frac{n(n-1)}{2}$). Intuitively, that means if we assume n to be small, then instances are small as well and the result that those instances are fast to solve is unsurprising.

The ILP technique described above was successfully used in computational social choice to show FPT-membership of problems dealing with winner determination [9], bribery [24, 29, 45], control [66], possible winner [15, 26], and lobbying [25]. We will also use this technique in Chapter 5 to show FPT-membership.

Similarly to NP-hardness we can define a notion of hardness for parameterized problems as well showing (under a separation assumption similar to P $\neq$ NP) that there is no FPT-algorithm for a parameterized problem. First, we need another set of parameterized complexity classes, the so-called W hierarchy. We omit the details of defining it formally as it is out of scope of this thesis (see the book by Downey and Fellows [47] for the formal definitions). The most important fact is that for every $t \geq 1$ there is a parameterized complexity class W[t] and it holds that FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP. Showing that a parameterized problem is W[t]-hard for some $t \geq 1$ and assuming that FPT $\neq$ W[1] then prevents the problem from being in FPT. For showing W[t]-hardness we need to extend the notion of reducibility to parameterized problems. We say a parameterized decision problem A, with parameter p, reduces to another parameterized decision problem B, with parameter r, if we can construct for each instance (I, p) of A in $O(f(p) \cdot |I|^{O(1)})$ time, for some computable function f, an instance (I', p') with $p' \leq g(p)$ for some computable function g such that (I, p) is a yes-instance of A if and only if (I', p') is a yes-instance of B. The main difference to "not-parameterized" reductions is that we need the constructed parameter for the instance of the target problem to be *exclusively* bounded by some function of the parameter of the original instance. With this notion of reducibility we can reduce some W[t]-hard problem to some other parameterized problem to show the W[t]-hardness of the latter problem. Although the formal definition of the W hierarchy is quite technical involving combinatorial circuits, for W[1] and W[2], there are natural problems with natural parameterizations that are hard for one of those classes and from which we can reduce to show hardness for other parameterized problems. For W[1]-hardness the following problem can be used.

MULTICOLORED CLIQUE	
Input:	Given an undirected graph $G = (V, E)$, an integer f, and a partition of V into f sets $W_1, \ldots, W_f$.
Question:	Does there exist a clique $X \subseteq V$ (i.e., the induced subgraph of G restricted to X is complete) that contains exactly one vertex of every set W_i with $1 \leq i \leq f$?

MULTICOLORED CLIQUE is W[1]-hard if parameterized by f [46]. For W[2]-hardness a central problem is SET COVER which is defined as follows.

SET COVER	
Input:	Given a set $X = \{x_1, ..., x_m\}$, a family $\mathscr{S} = \{S_1, \ldots, S_n\}$ of subsets of X, and an integer k.
Question:	Does there exist a cover of X of size at most k, i.e., a subfamily $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq k$ such that $\bigcup_{S_j \in \mathscr{S}'} S_j = X$?

If SET COVER is parameterized by k, it is W[2]-hard [46]. We note in passing that hardness for some class of the W hierarchy does not, in general, imply NP-hardness since a parameterized reduction allows the construction to be done in FPT-time with respect to the parameter which might be exponential-time with respect to the size of the input.

Lastly, we will discuss the parameterized complexity class para-NP which sits above XP (unless P = NP). A parameterized decision problem is para-NP-hard if it is NP-hard for some constant value of the parameter. Intuitively, if some slice of a parameterized problem is intractable, then it cannot belong to XP (unless P = NP) since this would imply that all slices are tractable. Interestingly, para-NP bridges the gap between parameterized complexity and classical complexity as it can be shown that FPT = para-NP if and only if P = NP [76].

2.2 Voting

An *election* is defined as a pair (C,V) with C being a finite set of *candidates* and V being a multiset of the voters' preferences over C, sometimes referred to as the *preference profile*. Typically, voters express their preference (i.e., the vote or the ballot they cast) as a linear order $\succ$ over the candidates in C with the following three properties.

1. Completeness: For every pair of candidates $c,d \in C$, we have $c \succ d$ or $d \succ c$;

2. Transitivity: For every triplet of candidates $c,d,e \in C$, if $c \succ d$ and $d \succ e$, it follows that $c \succ e$;

3. Antisymmetry: For every pair of candidates $c,d \in C$, if $c \succ d$, then $d \succ c$ does not hold.

For example, given a set of candidates $C = \{a,b,c,d\}$ a voter that prefers b to a, a to d, and d to c would have the preference $b \succ a \succ d \succ c$ (sometimes we omit the $\succ$ symbols and write the preference as a string $b\ a\ d\ c$). Note that the first and third property imply that the voter is sure for every pair of candidate which one the voter prefers over the other, i.e., the preferences are *strict*. In the (computational) social choice literature it is sometimes allowed that the voter may be indifferent of candidates (i.e., we drop the completeness property) but in this thesis we will always assume strict preferences. Apart from those *ordinal preferences* voters' preferences may be *cardinal* which means that each voter assigns each candidate a number of points. A special type of cardinal preferences are *approval-based preferences* in which a voter can only assign the values 0 or 1 to a candidate corresponding to the voter, respectively, disapproving or approving the candidate. Then, the vote is simply given as the subset of approved candidates. In some cases voters may even have preferences that are a mixture of both cardinal and ordinal preferences. In the following, we will assume that voters' preferences are ordinal and explicitly mention it in the few cases where it is not the case.

The outcome of an election is determined by a *voting rule*[5] that is a mapping which assigns every possible election a subset of the set of candidates which form the winners of the election. For some voting rule $\mathscr{E}$ and an election (C,V) we call a candidate that is part of the set of winners of the election under $\mathscr{E}$ an $\mathscr{E}$-winner of (C,V). A resolute voting rule, in which always only a single candidate can be a winner, is also known as a *social choice function*. If we are looking for a complete linear order over the candidates as an outcome of an election, we speak of *social welfare functions*.

We will now define the studied voting rules.

[5]In the literature, voting rules are often referred to as social choice correspondences, voting protocols, or voting systems.

Positional Scoring Rules

An important class of voting rules are (positional) scoring rules. Let m be the number of candidates. Then, scoring rules use a so-called *scoring vector* $(\alpha_1, \alpha_2, \dots, \alpha_m)$ with each α_i, $1 \leq i \leq m$, being a positive integer called a *score value* and $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$ (i.e., the monotonicity of the score values) to determine the score of each candidate (i.e., the candidate in position i in a vote gains α_i points and the points are summed up over all votes) and the candidate(s) with the highest score win(s). Therefore, each scoring rule is defined by a series of scoring vectors; one scoring vector for each possible number of candidates. To represent this infinite series of vectors succinctly Betzler and Dorn [14] defined the class of *pure scoring rules* in which we can obtain the scoring vector of size m by inserting an additional score value somewhere into the scoring vector of size $m-1$ maintaining the monotonicity of the scoring vector as described above. As Hemaspaandra, Hemaspaandra, and Schnoor [89] observe we can also assume that $\alpha_m = 0$ and that there is no integer which divides all score values which restricts the class of pure scoring rules only slightly.

We focus on the following (pure) scoring rules.

k-approval: The first k score values are 1 and all others are 0. 1-approval is also known as **plurality**.

k-veto: The last k score values are 0 and all others are 1. 1-veto is simply called **veto**.

Borda (Count): Let m be the number of candidates. Then, we have for each i, $1 \leq i \leq m$, that $\alpha_i = m - i$.

We can also define *iterative* variants of scoring rules in which the winner(s) are determined in several rounds.

Hare: Uses plurality scores to eliminate, in each round, the candidates with the lowest score until all remaining candidates have the same score which are proclaimed winners of the election. This voting rule is often known as single transferable voting (STV) but we will use the name STV for the multiwinner variant below.

Coombs: Works the same as Hare but uses veto scores.

Baldwin: Works the same as Hare but uses Borda scores.

Nanson: Uses Borda scores but eliminates all candidates that have less then the average Borda score which is defined as $(m-1)\frac{n}{2}$ with m being the number of candidates and n the number of voters.

Plurality/Veto with runoff: We always have only two rounds. In the first round only the candidates with the highest plurality/veto score proceed to the second round, except when there is a unique winner in which case the candidates with the highest and second highest plurality/veto score proceed to the next round. In the second round plurality/veto scores are used to determine the winner(s).

Iterated plurality/veto: All candidates are eliminated that do not have the highest plurality/veto score.

Notice that in all iterative voting rules we eliminate all candidates if there is a tie in some round. Sometimes a tie-breaking rule (e.g., a linear order over the candidates deciding which candidate is eliminated first if a tie occurs) is used to break ties instead. Then, only one candidate is eliminated in each round.

Lastly, we define the fallback voting rule which is a hybrid between positional scoring rules and approval-based voting rules.

Fallback: Instead of linear orders we assume that voters' preferences are given as a set of approved candidates and disapproved candidates while the former set is ordered linearly as well. For example, given a set of candidates $\{a,b,c,d,e\}$ a voter might have the set $\{a,b,c\}$ as approved candidates, the set $\{d,e\}$ as disapproved candidates, and orders the former set as $b \succ c \succ a$. Then this voter's vote would be written as $b \succ c \succ a \mid \{d,e\}$. We call a vote a level-i approval for some candidate c if c is in the first i positions of the voter's approved set of candidates. Furthermore, we call a candidate a level-i winner if the candidate is in the first i positions of approved sets of candidates in at least half of the voters' preferences. Then, the fallback winners are those candidates that are level-i winners for the smallest i and have the highest number of level-i approvals. If there are no such candidates, then fallback chooses the candidates with the most (overall) approvals as the winners.

Condorcet Extensions

The following voting rules rely on pairwise comparisons of the candidates. For an election (C,V) and two candidates $c,d \in C$, let $N_{(C,V)}(c,d)$ be the number of voters whose preferences rank c in front of d. Condorcet is one of the oldest and most prominent voting rules but has the downside that there may not be a winner at all. Therefore, Copeland and maximin try to imitate Condorcet by always choosing a Condorcet winner if there is one and provide a nonempty set of winners otherwise.

Condorcet: The Condorcet-winner is a candidate c who beats all other candidates in direct comparison (i.e., $N_{(C,V)}(c,d) > N_{(C,V)}(d,c)$ for all candidates $d \in C \setminus \{c\}$).

Copeland: The Copeland$^\alpha$ score with $0 \leq \alpha \leq 1$ and for a candidate $c \in C$ is defined as

$$|\{d \in C \setminus \{c\} \mid N_{(C,V)}(c,d) > N_{(C,V)}(d,c)\}| + \alpha|\{d \in C \setminus \{c\} \mid N_{(C,V)}(c,d) = N_{(C,V)}(d,c)\}|.$$

Intuitively, c gains a point for each candidate that c beats in direct comparison and α points for each tie. Then, the Copeland$^\alpha$ winners are the candidates with the highest Copeland$^\alpha$ score. Copeland$^\alpha$ with $\alpha = \frac{1}{2}$ is referred to as **Copeland**.

Maximin: The maximin score of a candidate c is defined as $\min_{d \in C \setminus \{c\}} N_{(C,V)}(c,d)$. Then, the candidates with the highest maximin score are the winners.

Range Voting

For the last two voting rules we assume cardinal preferences. That means a voter's preference is a point vector $v \in \{0,1,\ldots,k\}^m$ of size m and describes the amount of points a voter assigns to every candidate. (We assume here that the candidates are ordered lexicographically such that the i-th component of the vector corresponds to the i-th candidate according to this ordering.) The number k is the maximum number of points a voter can give to a single candidate. For an election (C,V), if k is fixed and every voters gives at most k points to a candidate we call (C,V) a *k-range election*. Note that in a k-range election it is not required that all voters give 0 or k points to some candidate which in reality is very unlikely as voters tend to maximize the points given to their favorite candidate and minimize the points given to their most despised candidate. We will later see how votes are normalized to display this behavior.

Range Voting: Given a k-range election, we simply sum up the points each candidate is given by the voters and the candidates with the highest score are k-range voting winners. 1-range voting is commonly known as **approval voting**.

Normalized Range Voting: Given a k-range election, we first normalize each voter's point vector as follows. For a candidate $c \in C$ and a voter $v \in V$, let s be the points the candidate is given by this voter and let $s_{\min}$ and $s_{\max}$ be the minimal and maximal points given to any candidate by this voter. Then, the normalized score that c is given by voter v is $\frac{k(s-s_{\min})}{s_{\max}-s_{\min}}$. We can assume that $s_{\min}$ and $s_{\max}$ are not equal for otherwise the voter would be indifferent of every candidate. Similarly to range voting, we will then sum up the normalized points for each candidate and the candidates with the highest normalized score win.

For all of the voting rules above the outcome of an election can be computed in polynomial time. Voting rules for which winner determination[6] is NP-hard (e.g., Kemeny [94] and Dodgson [44]) will not be discussed here but some of them will be defined later in the context of multiwinner elections.

Manipulating Elections

We will now explore how different actors in an election may be able to influence its outcome. We start with insincere voters which is often known as strategic voting. Consider the following example.

Example 2.6 (Manipulation)**.** Let $C = \{a,b,c,d\}$ and consider the following four voters in V.

$$
\begin{aligned}
v_1 &: c \succ d \succ b \succ a \\
v_2 &: a \succ d \succ c \succ b \\
v_3 &: d \succ c \succ b \succ a \\
v_4 &: a \succ d \succ c \succ b
\end{aligned}
$$

The Borda scores of the candidates in (C,V) are as follows. Candidate a has 6 points, b has 2 points, c has 7 points, and d has 9 points. So, d is the unique Borda-winner of the election. Now consider the first voter v_1 and assume that before she casts her vote she finds out how the other voters vote and that together with her (honest) vote her favorite candidate c does not win. Then, she might be tempted to change her vote to $c \succ b \succ a \succ d$ which would lead to c being tied for the win together with a and d.

[6]The problem $\mathscr{E}$-WINNER DETERMINATION for a singlewinner voting rule $\mathscr{E}$ is defined by the input that consists of an election (C,V) and candidate $c \in C$ and the question of whether c is an $\mathscr{E}$-winner of (C,V). On page 28 we will define the problem for multiwinner voting rules as well, which is slightly different than this variant for singlewinner voting rules.

Strategic voting with only one manipulating voter is formalized as the $\mathscr{E}$-MANIPULATION problem for some voting rule $\mathscr{E}$ which was first defined by Bartholdi, Tovey, and Trick [8] who studied the complexity of this problem for various voting rules.

$\mathscr{E}$-MANIPULATION	
Input:	An election (C,V), an additional voter v (the strategic voter), and a distinguished candidate c.
Question:	Is there a vote that v can cast such that c is an $\mathscr{E}$-winner of the election $(C,V\cup\{v\})$?

Note that in contrast to Example 2.6 the honest vote of the strategic voter is not given in the input and we simply ask if there is any vote v can cast so that c wins which might as well be her honest vote. Example 2.6 illustrates that a voter can actually benefit from casting a strategic vote instead of an honest one.

For most voting rules it seems that MANIPULATION with only a single manipulator is easy since Bartholdi, Tovey, and Trick [8] provided a simple greedy algorithm that solves MANIPULATION for many common voting rules. Surprisingly, they also found a natural voting rule for which MANIPULATION is NP-hard, namely second-level Copeland which is the Copeland voting rule with a tie-breaking mechanism. Later on, the NP-hardness of MANIPULATION was also shown for Hare with single-candidate elimination by Bartholdi and Orlin [7] and for a voting rule called Ranked Pairs by Xia et al. [158].

If there is more than one strategic voter and they work together, this is called coalitional manipulation and was formalized by Conitzer, Sandholm, and Lang [36] for so-called weighted elections. In a weighted election (C,V,w) in addition to the set of candidates C and the multiset of the voters' preferences V we are given a weight function $w: V \to \mathbb{N}$ that assigns every voter a positive integer. Although elections with weighted votes are violating the democratic principle that all votes should be weighted equal in many settings this is not the case. For example, the countries in the European Union are weighted and in decision processes within a company the shareholders' votes are weighted depending on how many shares each shareholder holds. Up until now we have always assumed that the manipulating actor has a favorite candidate that she wants to make a winner of the election but she may also have a despised candidate which she wants to prevent from winning. Conitzer, Sandholm, and Lang [36] described those two notions as constructive manipulation and destructive manipulation letting them define the following decision problems as their central problems to study.

$\mathscr{E}$-CONSTRUCTIVE-COALITIONAL-WEIGHTED-MANIPULATION ($\mathscr{E}$-CCWM)	
Input:	A weighted election (C,V,w), a coalition of manipulators V' with weights w', and a distinguished candidate c.
Question:	Are there votes the manipulators in V' can cast such that c is an $\mathscr{E}$-winner of the weighted election $(C,V\cup V',w\cup w')$?

The destructive variant ($\mathscr{E}$-DCWM) has the same input and asks whether there are votes the manipulators of V' can cast such that c is *not* an $\mathscr{E}$-winner of the weighted election $(C,V\cup V',w\cup w')$. Those two problems were studied for a variety of voting rules by Conitzer, Sandholm, and Lang [36] especially under the aspect of how many manipulators are needed until coalitional manipulation becomes intractable for a voting rule. Continuing this line of research, Hemaspaandra and Hemaspaandra [82] settled the complexity of weighted manipulation for the class of all scoring rules by showing that weighted manipulation is intractable for a scoring rule if and only if it satisfies the "diversity of dislike" property (i.e., the score value of the second best and worst candidate in a vote are different).

In contrast, much less is known of the unweighted variant (i.e., each weight is 1) of these problems: Faliszewski, Hemaspaandra, and Schnoor [68] showed that coalitional manipulation is intractable for Copeland voting while it is tractable with only one manipulator; Davies et al. [41] and Betzler, Niedermeier, and Woeginger [16] independently showed that coalitional manipulation is intractable for Borda; and other results were shown by Xia et al. [157, 158] and Narodytska and Walsh [116].

Electoral Control

Besides manipulation, Bartholdi, Tovey, and Trick initiated the study of electoral control in 1992 [10]. Instead of manipulation attempts by voters electoral control deals with election tampering attempts by the election chair that organizes the election. The chair can influence the structural parts of the election which is the set of candidates or the voters that participate in the election and might even be able to influence the election process by holding subelections.

Example 2.7 (Electoral control)**.** Consider the election of Example 2.6 again evaluated with Borda. An election chair who would like c to win might choose to remove the candidate d from the election which would change the original election as follows.

Original election $(\{a,b,c,d\},V)$	Controlled election $(\{a,b,c\},V)$
$v_1 : c \succ d \succ b \succ a$	$v_1 : c \succ b \succ a$
$v_2 : a \succ d \succ c \succ b$	$v_2 : a \succ c \succ b$
$v_3 : d \succ c \succ b \succ a$	$v_3 : c \succ b \succ a$
$v_4 : a \succ d \succ c \succ b$	$v_4 : a \succ c \succ b$

In the controlled election, a has 4 points, b has 2 points, and c has 6 points turning c into the unique winner of the election while d won the original election uniquely.

Bartholdi, Tovey, and Trick [10] defined eleven different decision problems dealing with various kinds of election tampering by the chair which were doubled to 22 problems by Hemaspaandra, Hemaspaandra, and Rothe [83] who defined the destructive variants of the original electoral control problems. The problems concerned with altering the set of candidates and the multiset of voters' preferences can be conveniently combined to the following problem called multimode control problem which was first defined by Faliszewski, Hemaspaandra, and Hemaspaandra [65].

$\mathscr{E}$-CONSTRUCTIVE-MULTIMODE-CONTROL	
Input:	An election $(C \cup D, V \cup W)$ with C and D being disjoint sets of, respectively, registered and unregistered candidates and V and W being disjoint multisets of, respectively, preferences of registered and unregistered voters, four nonnegative integers $\ell_{DC}, \ell_{AC}, \ell_{DV}$, and ℓ_{AV}, and a distinguished candidate $c \in C$.
Question:	Are there subsets $C' \subseteq C \setminus \{c\}, D' \subseteq D, V' \subseteq V$, and $W' \subseteq W$ with $\lvert C'\rvert \leq \ell_{DC}, \lvert D'\rvert \leq \ell_{AC}, \lvert V'\rvert \leq \ell_{DV}$, and $\lvert W'\rvert \leq \ell_{AV}$ such that c is an $\mathscr{E}$-winner of the election $((C \setminus C') \cup D', (V \setminus V') \cup W')$?

Then, we obtain the special cases

$\mathscr{E}$-Constructive-Control-by-Adding-Candidates ($\mathscr{E}$-CCAC) by setting $\ell_{DC} = \ell_{AV} = \ell_{DV} = 0$ and $W = \emptyset$;

$\mathscr{E}$-Constructive-Control-by-Adding-an-Unlimited-Number-of-Candidates ($\mathscr{E}$-CCAUC) by setting $\ell_{DC} = \ell_{AV} = \ell_{DV} = 0$, $\ell_{AC} = \lvert D\rvert$, and $W = \emptyset$;

$\mathscr{E}$-Constructive-Control-by-Deleting-Candidates ($\mathscr{E}$-CCDC) by setting $\ell_{AC} = \ell_{AV} = \ell_{DV} = 0$ and $D = W = \emptyset$;

$\mathscr{E}$-Constructive-Control-by-Adding-Voters ($\mathscr{E}$-CCAV) by setting $\ell_{AC} = \ell_{DC} = \ell_{DV} = 0$ and $D = \emptyset$; and

$\mathscr{E}$-Constructive-Control-by-Deleting-Voters ($\mathscr{E}$-CCDV) by setting $\ell_{AC} = \ell_{DC} = \ell_{AV} = 0$ and $D = W = \emptyset$.

We will study those problems for various voting rules in Chapter 4.

The second set of problems deals with partitioning the set of candidates or multiset of voters' preferences. Then, one or two subelections are run before the overall winners are decided by a final election with a reduced set of candidates. Note that in an election with a reduced set of candidates the votes are always masked down to the participating candidates (see Example 2.8 below). Bartholdi, Tovey, and Trick [10] also considered two notions of tie-breaking in the subelections. The first one is called *ties-promote* (TP) in which all tied candidates proceed to the final and the second one is called *ties-eliminate* (TE) in which only a unique winner of a subelection proceeds to final and all candidates are eliminated if there is a tie for the win. We can now define the decision problems. Given an election (C,V) and a distinguished candidate $c \in C$, $\mathscr{E}$-CONSTRUCTIVE-CONTROL-BY-RUNOFF-PARTITION-OF-CANDIDATES ($\mathscr{E}$-CCRPC) asks whether we can partition C into two disjoint subsets C_1 and C_2 such that c is an $\mathscr{E}$-winner of the two stage election in which the winners of the first stage (sub)elections (C_1,V) and (C_2,V) (with either ties-eliminate or ties-promote tie-breaking) proceed to a second and final runoff election in which the overall winners are determined. Then, we can define the following variants that have the same input as $\mathscr{E}$-CCRPC but ask slightly different questions.

- For $\mathscr{E}$-CONSTRUCTIVE-CONTROL-BY-PARTITION-OF-CANDIDATES ($\mathscr{E}$-CCPC) we ask the same question as for $\mathscr{E}$-CCRPC except we only run one subelection in the first stage, (C_1,V), and the candidates from C_2 get a bye to the final election, and

- $\mathscr{E}$-CONSTRUCTIVE-CONTROL-BY-PARTITION-OF-VOTERS ($\mathscr{E}$-CCPV) asks the question of whether there is a partition of V into two disjoint subsets, V_1 and V_2, such that c is an $\mathscr{E}$-winner of the two stage election in which the winners of the first stage (sub)elections (C, V_1) and (C, V_2) (with either ties-eliminate or ties-promote tie-breaking) proceed to a second and final runoff election with all voters in V in which the overall winners are determined.

Again, we obtain the destructive versions of the above problems by asking if we can make sure that c is not a winner of the election and replacing "Constructive" with "Destructive" in the problem names. To indicate which tie-breaking mechanism is used, we append "TE" for ties-eliminate tie-breaking or "TP" for ties-promote ties-breaking to the problem names. We illustrate how electoral control by partitioning the set of candidates works with the following example.

Example 2.8 (Control by runoff partition of candidates)**.** We will use ties-eliminate tie-breaking in this example. Consider the election of Example 2.6 which d wins uniquely if the election is evaluated by Borda. If we want do prevent d from winning (i.e., destructive control), we might choose to partition the candidate set C into $C_1 = \{a, d\}$ and $C_2 = \{b, c\}$. Then the first stage contains the following subelections with the votes being masked down to the respective reduced sets of candidates.

First subelection $(\{a, d\}, V)$	Second subelection $(\{b, c\}, V)$
$v_1 : d \succ a$	$v_1 : c \succ b$
$v_2 : a \succ d$	$v_2 : c \succ b$
$v_3 : d \succ a$	$v_3 : c \succ b$
$v_4 : a \succ d$	$v_4 : c \succ b$

If we again use Borda to evaluate the subelections, a and d tie for the win in the first subelection and are eliminated due to ties-eliminate tie-breaking. Therefore, we have achieved our goal to prevent d from winning. For completeness, c beats b in the second subelection and proceeds to the final election which is won by c as well since she is the only candidate still standing.

In contrast to manipulation we might not be able to influence the election outcome by some type of electoral control using some voting rule. For example, assume we try to make some candidate a winner of an election evaluated with Condorcet by adding additional candidates. Then, either the candidate is already a winner of the election or she is beaten by some candidate in pairwise comparison which we cannot change by adding additional candidates to the election. In this case we would call the voting rule *immune* against this type of electoral control. Otherwise we call the voting rule *susceptible* to this type of electoral control and further investigate the computational complexity of the associated decision problem. If we can show that the decision problem is solvable in polynomial time, we call the voting rule *vulnerable* to this type of electoral control or if we can show that the decision problem is NP-hard, we call the voting rule *resistant* against this type of electoral control.

Electoral control has since been studied extensively for a variety of voting rules [56, 59, 66, 108, 109, 114, 127]. Currently, the voting rules with the most resistances against electoral control types are fallback (see the work of Erdélyi et al. [56]) and normalized range voting (see the work of Menton [114]) who are only vulnerable to two of the 22 types. Although Hemaspaandra, Hemaspaandra, and Rothe [84] constructed a hybrid voting rule that is resistant against all types while still being computationally easy to compute there is still no "natural" voting rule that is resistant to all 22 types. In Section 3 we will continue the study of classical types of electoral control for the Borda Count.

Another type of electoral control that was not part of the set of classical control types but related to control by adding candidates is control by cloning of candidates. The notion of cloning candidates in elections was first studied by Tideman [150] as the so-called *independence of clones* property of voting rules and was later formalized as a decision problem by Elkind, Faliszewski, and Slinko [53]. They defined the action of cloning candidates as a size-m vector of nonnegative integers $(k_1,\dots,k_m)$ with m being the number of candidates and some arbitrary (e.g., lexicographic) ordering of the candidates. Each entry k_i, $1 \leq i \leq m$, in the vector with $k_i > 0$ means that the i-th candidate of the election is replaced by k_i clones $c_i^{(1)},\dots,c_i^{(k_i)}$. If $k_i = 0$ for some i with $1 \leq i \leq m$, then c_i stays in the election and no clone for this candidate is added. Notice that this definition of cloning candidates is slightly different than in the work of Elkind, Faliszewski, and Slinko [53] in that we allow candidates to not be cloned (Elkind, Faliszewski, and Slinko [53] replace every candidate by at least one clone) which seems more natural and does not restrict the model. Given an election $E = (C,V)$ with $C = \{c_1,\dots,c_m\}$ and a vector $K = (k_1,\dots,k_m)$ of nonnegative integers, a *cloned election* $E^* = (C^*,V^*)$ via K is derived from E by the set of candidates

$$C^* = \left(C \setminus \bigcup_{k_i \in K, k_i > 0} \{c_i\}\right) \cup \left(\bigcup_{k_i \in K, k_i > 0} \{c_i^{(1)},\dots,c_i^{(k_i)}\}\right)$$

and for every vote $v_j \in V$ there is a vote $v_j^* \in V^*$ which is a (complete) linear order over C^* such that for every pair of candidates $c_i, c_j \in C$ it holds for v_j that $c_i \succ c_j$ if and only if $c'_i \succ c'_j$ in v_j^* with $c'_i = c_i$ if $k_i = 0$ or for every $c'_i \in \{c_i^{(1)},\dots,c_i^{(k_i)}\}$ otherwise, and $c'_j = c_j$ if $k_j = 0$ or for every $c'_j \in \{c_j^{(1)},\dots,c_j^{(k_j)}\}$ otherwise. We illustrate the notion of cloned elections with the following example.

Example 2.9 (Cloned elections)**.** Let $E = (C,V)$ be an election with $C = \{c_1,c_2,c_3\}$ and V consisting of two voters with preferences $v_1 : c_1 \succ c_2 \succ c_3$ and $v_2 : c_2 \succ c_1 \succ c_3$. Consider the vector $K = (0,2,0)$ which means that c_1 and c_3 remain in the election but c_2 is replaced by two clones $c_2^{(1)}$ and $c_2^{(2)}$ yielding $C^* = \{c_1,c_2^{(1)},c_2^{(2)},c_3\}$. Regarding the voters, v_1 might be extended to $v_1^{(1)} : c_1 \succ c_2^{(1)} \succ c_2^{(2)} \succ c_3$ or $v_1^{(2)} : c_1 \succ c_2^{(2)} \succ c_2^{(1)} \succ c_3$ and v_2 to $v_2^{(1)} : c_2^{(1)} \succ c_2^{(2)} \succ c_1 \succ c_3$ or $v_2^{(2)} : c_2^{(2)} \succ c_2^{(1)} \succ c_1 \succ c_3$. Thus, there are four possible cloned elections of E via K (i.e., $(C^*,\{v_1^{(1)},v_2^{(1)}\})$, $(C^*,\{v_1^{(1)},v_2^{(2)}\})$, $(C^*,\{v_1^{(2)},v_2^{(1)}\})$, and $(C^*,\{v_1^{(2)},v_2^{(2)}\})$).

Notice that there are several possible cloned elections depending on how the clones of a single candidate are ordered against each other. Therefore, the following decision problem is defined for some $q \in \{0^+\} \cup (0,1]$ that describes the *probability of success* that we need to reach. That means that q is the fraction of all possible cloned elections in which the distinguished candidate needs to be a winner. The special case $q = 0^+$ means that we need only one cloned election in which the distinguished candidate is a winner in order to be successful. To decide to what degree we can clone candidates we are given a cost function $\rho : [m] \times [t] \to \mathbb{N} \cup \{+\infty\}$ for some integer $t > 1$. Then, $\rho(i,j)$ defines the cost of adding the jth clone of the ith candidate to the election. Since adding the first clone of a candidate only replaces the original candidate we require $\rho(i,1) = 0$ for every $i, 1 \leq i \leq m$.

$\mathscr{E}$-q-CLONING	
Input:	An election (C,V), a distinguished candidate $c \in C$, a positive integer $t > 1$, a cost function $\rho : [m] \times [t] \to \mathbb{N} \cup \{+\infty\}$, and a budget B.
Question:	Is there a vector of nonnegative integers $K = (k_1, \ldots, k_m)$ with $\sum_{k_i \in K} \sum_{j=2}^{k_i} \rho(i,j) \leq B$ such that c (or some clone of c) is an $\mathscr{E}$-winner of a cloned election of (C,V) via K with probability q?

Elkind, Faliszewski, and Slinko [53] also considered two special cases with the cost functions that have $\rho(i,j) = 0$ for all i, $1 \leq i \leq m$, and $j \in \mathbb{N}$ which is called ZERO COST (ZC) and $\rho(i,j) = 1$ for all i, $1 \leq i \leq m$ and $j \geq 2$ which is called UNIT COST (UC).

Bribery

In contrast to manipulation and electoral control, the notion of bribery was introduced to computational social choice only much later by Faliszewski, Hemaspaandra, and Hemaspaandra [63]. Bribery assumes that there is an outside agent that tries to influence an election by bribing the voters to change their vote to the outside agent's preference.

Example 2.10 (Bribery)**.** Again, consider the election of Example 2.6 evaluated with Borda. If we want the candidate c to be the winner, we can bribe the second voter to change her vote to $c \succ a \succ d \succ b$ which would lead to c scoring 9 points while d scores 8 points, a scores 5 points, and b scores 3 points.

Similarly to the previous section we will define a very general bribery problem (see the book chapter by Faliszewski and Rothe [70]) that captures the different flavors of bribery by Faliszewski, Hemaspaandra, and Hemaspaandra [63] and by Elkind, Faliszewski, and Slinko [52].

$\mathscr{E}$-CONSTRUCTIVE-PRICED-BRIBERY	
Input:	An election (C,V) with m candidates and n voters, a list of price functions $(\rho_1, \ldots, \rho_n)$ such that for each i, $1 \leq i \leq n$, and each possible linear order υ over C, $\rho_i(\upsilon)$ is the price to pay so that voter i changes her vote to υ, a distinguished candidate c, and a positive integer B.
Question:	Can we bribe the voters in V with a budget of B such that c becomes an $\mathscr{E}$-winner of the resulting election?

Then, we can define the other bribery problems by restricting the range of price functions the voters may have. The problems $\mathscr{E}$-CONSTRUCTIVE-BRIBERY and $\mathscr{E}$-CONSTRUCTIVE-\$BRIBERY defined by Faliszewski, Hemaspaandra, and Hemaspaandra [63] have so-called *discrete* and *\$discrete* price functions, respectively. We call a price function ρ_i discrete if $\rho_i(\upsilon_i) = 0$ with υ_i being the preference order of voter i and $\rho_i(\upsilon) = 1$ for every preference order $\upsilon \neq \upsilon_i$ (intuitively, the briber pays nothing for not bribing and can freely change the preference of a voter for unit cost). A \$discrete price function ρ_i is defined similarly except that we have $\rho_i(\upsilon) = c_i$ for every preference order $\upsilon \neq \upsilon_i$ with c_i being some constant (i.e., the price of voter i to be bribed which may vary for different voters). The third type of price functions are *swap-bribery* price functions which were first introduced by

Faliszewski et al. [66] for irrational voters[7] and later studied by Elkind, Faliszewski, and Slinko [52] for voters with linear preference orders. Formally, a swap-bribery price function ρ_i is defined by a constant $c_i^{\{x,y\}}$ for each pair of candidates $x, y \in C$ such that for each preference order υ, $\rho_i(\upsilon)$ is the sum of all constants $c_i^{\{x,y\}}$ for which the candidates $x, y \in C$ are in opposite order in υ and υ_i, the preference order of voter i (intuitively, the briber pays a voter to swap two candidates in her preference order). SWAP-BRIBERY turned out to be NP-hard for most of the voting rules considered by Elkind, Faliszewski, and Slinko [52] so they also studied a natural special case called *shift bribery*. For $\mathscr{E}$-SHIFT-BRIBERY swap-bribery price functions are used with the restriction that for each pair of candidates $x, y \in C \setminus \{c\}$, with c being the distinguished candidate, we have $c_i^{\{x,y\}} = B + 1$ (intuitively, the briber can only shift the distinguished candidate forwards or backwards in the voters' preference orders). Swap bribery and shift bribery have natural applications in practice as they model campaign management. A campaign manager for a specific candidate might try to improve her candidate's chances of winning by running ads that target specific groups of voters and make them change their opinion of the ordering of candidates. Shift bribery models a more ethical approach to campaign management as only the position of the distinguished candidate (i.e., the candidate for which the campaign is managed) may be altered by campaign management actions such as ads. Due to this very natural application, shift bribery has been thoroughly studied since its introduction. Schlotter, Faliszewski, and Elkind [143] studied shift bribery for approval-like voting rules; Bredereck et al. [24] studied shift bribery for several classes of price functions; Kaczmarczyk and Faliszewski [92] studied destructive shift bribery; Bredereck et al. [29] studied shift bribery in the context of multiwinner elections; and Bredereck et al. [28] studied a combinatorial variant of shift bribery in which one bribe action causes changes to the preferences of multiple voters. In Section 6 we will extend the study of shift bribery to the iterative voting rules defined above.

Each of the above bribery problems can also be defined for (a) weighted elections which will be denoted by adding "Weighted" to the problem names and (b) with a destructive goal which will be denoted by replacing "Constructive" with "Destructive" in the problem names. Destructive bribery is especially interesting as it can measure the *robustness* of an election result (see the work of Xia [104] for a more detailed discussion): If the winner of an election can be dethroned by only a few changes to the election, the current winner might be wrong due to vote counting errors or even raise the suspicion of election manipulation. The robustness of election results (in the context of multiwinner elections) was also studied by Bredereck et al. [27] although their method of investigating robustness is not directly related to bribery. Furthermore, Dey, Misra, and Narahari [125] studied frugal bribery in which a voter can only be bribed if the change to her vote improves the election result for this voter with respect to her preference; Faliszewski [62] studied so-called nonuniform bribery which is a model of bribery for (k, b)-elections which is a special type of elections in which voters submit their preferences by allocating k points to the candidates while never giving a candidate more than b points; and Erdélyi, Hemaspaandra, and Hemaspaandra [57] studied bribery under the assumption that the voting rule used to evaluate the election is not fixed, i.e., there is uncertainty about which voting rule is used.

Notice that in all decision problems that we have defined above the goal is to make the distinguished candidate *a* winner of the election which means that the distinguished candidate does not need to beat every candidate but at least tie them. This is called the *nonunique-winner model*. In contrast, we can ask for the distinguished candidate to be the unique winner which is then called the *unique-winner*

[7] In contrast to (rational) voters having linear preference orders, an irrational voters may have cycles in her preference order. For example, given a set of three candidates $\{a, b, c\}$ an irrational voter may prefer a to b, b to c, and c to a.

model. The former is more common in the computational social choice literature which is why we use this winner model as well. We will later see that the choice of the winner model is not only a matter of taste but there might even be a change in complexity for some decision problems when the winner model is changed.

Possible and Necessary Winners

Up until now we required the voters to have complete preferences over the candidates. In practice, this is very rarely the case: The ballots of voters are kept secret until the election is over and some voting rules, such as plurality, do not require complete preferences. Moreover, complete preferences might not even be desirable: Does a voter really know who she prefers of every pair of candidates or are most of them simply ordered randomly or, even worse, lexicographically? Regarding "unrealistic" complete preferences, one could argue that if some type of election tampering is hard with full information, it is at least as hard with only partial information. Still, it makes sense to study elections with partial information.

We can define *partial* preferences from complete preferences by dropping the completeness property (i.e., a partial preference is a linear order over the candidates that is transitive and antisymmetric). Usually, a partial preference is defined by a set of pairwise comparisons of the form $c_i \succ c_j$. Then, a partial preference profile is a multiset of the voters' partial preferences. A complete preference v' over a set of candidates C *extends* a partial preference v over C if for all $c_i, c_j \in C$ it holds that if $c_i \succ c_j$ in v, then $c_i \succ c_j$ in v'. We call a multiset of complete preferences $\{v'_1, \dots, v'_n\}$ an *extension* of a multiset of partial preferences $\{v_1, \dots, v_n\}$ if for every i, $1 \leq i \leq n$, v'_i extends v_i.

We can now define the $\mathscr{E}$-POSSIBLE-WINNER and $\mathscr{E}$-NECESSARY-WINNER problems introduced by Konczak and Lang [97].

$\mathscr{E}$-POSSIBLE-WINNER	
Input:	An election (C,V) with a set of candidates C and a partial preference profile V and a distinguished candidate c.
Question:	Is there an extension V' of V to complete preferences such that c is an $\mathscr{E}$-winner of the election (C,V')?

$\mathscr{E}$-NECESSARY-WINNER is defined similarly but we ask whether c is an $\mathscr{E}$-winner of the election (C,V') *for all* extensions V' of V.

Both problems were further studied by Xia and Conitzer [156], Walsh [153], Pini et al. [130], Betzler, Hemmann, and Niedermeier [15], Betzler and Dorn [14], and Baumeister and Rothe [12]. Interestingly, $\mathscr{E}$-Possible-Winner generalizes the $\mathscr{E}$-CONSTRUCTIVE-COALITIONAL-MANIPULATION problem [97] and is itself a special case of $\mathscr{E}$-SWAP-BRIBERY [52].

Electoral Control in Sequential Elections

Another partial information model was introduced and studied by Hemaspaandra, Hemaspaandra, and Rothe [85, 86, 87, 88] in a series of papers concerning different kinds of election tampering attempts in sequential elections. We will define the so-called *online* models for electoral control [86, 87] in detail and refer to the corresponding papers for the online models for manipulation [85] and bribery [88]. Later in Section 3 we will study the *online* models for electoral control for the Borda Count.

Online candidate control [86] models voting scenarios in which the candidates are added to the election (and evaluated against the already participating candidates by the voters) one after the other and the election chair may decide, only at the moment a candidate appears and never after that, to exert a control action (such as adding or deleting) on this candidate. The corresponding online control problems *online constructive control by deleting candidates* for a voting rule $\mathscr{E}$ (online-$\mathscr{E}$-CCDC), *online constructive control by adding candidates* for a voting rule $\mathscr{E}$ (online-$\mathscr{E}$-CCAC) and their destructive variants online-$\mathscr{E}$-DCDC and online-$\mathscr{E}$-DCAC capture such a *moment of decision* for the election chair. For online-$\mathscr{E}$-CCDC we are given the set of candidates C, the set of voters V (note that only in this section the voters' preferences are given separately later as they are not complete over the set of candidates), the election chair's ideal ranking σ over the candidates, the election chair's distinguished candidate $d \in C$, an order of the candidates describing in which order they appear in the election with a flag for each candidate saying who the current candidate is and which of the already revealed candidates were deleted, the voters' preferences over the still standing (i.e., already revealed but not deleted) candidates including the current candidate, and the number of deletions k that the election chair has left to use. Then we ask whether the election chair can make a decision about the current candidate (whether to delete her if possible or not) so that the chair has a *forced win* by which we mean that no matter what happens in the future (i.e., how not yet revealed candidates appear in the voters' preferences) the chair can make decisions on later revealed candidates with the information available at the time such that the distinguished candidate d or some candidate ranked higher than d according to the chair's ranking σ is an $\mathscr{E}$-winner of the election in which only the not-deleted candidates participate.

The following example illustrates how a moment of decision and a forced win work for online-$\mathscr{E}$-CCDC.

Example 2.11 (Online control by deleting candidates)**.** In this example we will use plurality as the voting rule. Consider the following instance of online-$\mathscr{E}$-CCDC.

- Let $C = \{a,c,d,e\}$ and $V = \{v_1,v_2\}$.
- The chair's ranking is $d \succ a \succ b \succ c$.
- The distinguished candidate is d (i.e., the chair succeeds only if d wins).
- The candidates' order of appearance is d a b c.
- No candidate has been deleted as of yet and the current candidate is a (i.e., d and a are already revealed).
- The voters preferences are $v_1 : \ d \succ a$ and $v_2 : \ a \succ d$ (note that b and c have not shown up yet and are therefore not included in the preferences).
- Finally, $k = 2$.

Now the chair has to decide whether the current candidate a must be removed or not in order to have a forced win (i.e., no matter how the not yet revealed candidates b and c appear in the voters' preferences there exist decisions about b and c such that d wins). Notice that if the chair decides to remove a, then there is only one removal left so either b or c must remain in the election. In the worst case the candidate that cannot be deleted will be ranked above d in both preferences and therefore beats d. So, by removing a the chair does not have a forced win. If the chair decides to leave a in the election, both b and c can be removed from the election later so no matter how they actually appear in the preferences at future moments of decision they will not appear in the preferences after all candidates have been revealed. Since d wins the election $(\{a,d\},V)$, the chair has a forced win by not deleting a.

For online-$\mathscr{E}$-CCAC the input changes slightly: We now have a set of *registered* candidates that are certainly part of the election and a disjoint set of unregistered candidates that may be added to the election only at the moment of decision when such a candidate is revealed. The order of appearance of candidates is over the union of both sets and the rest stays the same (the flag for a candidate now indicates whether an already revealed, unregistered candidate has been added to the election by the election chair and the deletion limit k is now an addition limit).

For the destructive variants, online-$\mathscr{E}$-DCDC and online-$\mathscr{E}$-DCAC, the chair's goal is to make sure none of the candidates d or worse in their ideal ranking win after all decisions have been made. Regarding online-$\mathscr{E}$-DCDC the election chair might try to delete all candidates d or worse to win trivially so Hemaspaandra, Hemaspaandra, and Rothe [86] proposed two approaches to prevent this behavior. The first one is called the *non-hand-tied chair model* and lets the chair delete some but never all candidates d or worse. In contrast the *hand-tied chair model* prevents the election chair from deleting any candidates d or worse.

The **online voter control** model [87] assumes that the set of candidates that are part of the election is fixed but now the voters are revealed sequentially (with preferences over the full set of candidates) and susceptible to control actions by the election chair, again, only at the moment they are revealed. Before we define the control problems that were introduced by Hemaspaandra, Hemaspaandra, and Rothe [87] we define the general information that all of them have in common namely an *online voter control setting* (OVCS) given by (C,u,V,σ,d) which contains the candidate set C, the current voter u, an *election snapshot* $V = (V_{<u},u,V_{u<})$ with $V_{<u}$ being the set of voters that were revealed before u and $V_{u<}$ being the set of not-yet-revealed voters, the election chair's ideal ranking σ of the candidates, and a distinguished candidate d. Note that $V_{<u}$ and u have already cast their votes so their preference orders are known but $V_{<u}$ only specifies the order in which the not-yet-revealed voters cast their votes. Then, the question is whether the election chair can make a decision about the current voter (whether to exert the control action at hand if possible or not) to have a *forced win* (i.e., the election chair can reach their—constructive or destructive—goal by making future decisions about not-yet-revealed candidates with the—up to each point of decision—revealed information). As before, the constructive goal of the chair is to make the candidate d or some candidate that is ranked higher than d in their ideal ranking a winner of the election after all voters have shown up and all decisions about the voters have been made, and the destructive goal aims to prevent all candidates d or worse in the chair's ideal ranking from winning.

For *online control by deleting voters* (i.e., the problems online-E-CCDV and online-E-DCDV) in addition to an OVCS we are given a nonnegative integer k (the number of deletions the election chair has left to use) and for each voter of $V_{<u}$ a flag that says whether the voter was deleted or not. The election after the voting process includes all voters that were not deleted by the election chair.

For *online control by adding voters* (i.e., the problems online-E-CCAV and online-E-DCAV) the OVCS is, again, augmented by a nonnegative integer k which is the limit of additions the election chair may use and for each voter there is a flag that indicates whether the voter is unregistered (i.e., the chair can choose to add her or not) or registered (i.e., the voter is definitely in the election) and for each voter in $V_{<u}$ there is another flag indicating whether this voter was added to the election. The election after all voters have shown up then includes the registered voters and all unregistered voters that have been added by the election chair.

Lastly, for *online control by partition of voters* (i.e., the decision problems online-E-CCPV and online-E-DCPV) the chair partitions the set of voters by assigning each voter to the left or the right part of the partition after they are revealed. Then, after all voters have been revealed the election proceeds in two stages in which the winners of two subelections with each part of the partition determine the overall winners in a final runoff election with all voters. So, in addition to an OVCS we are given a flag for every voter in $V_{<u}$ that indicates whether a voter was assigned to the left part or the right part of the partition. Similarly to the classical control by partition problems we adopt either the ties-promote model or the ties-eliminate model to decide whether, respectively, all candidates or none of the candidates that are tied for the win in a subelection proceed.

Multiwinner Voting

Another branch of the computational social choice landscape is concerned with elections that have a fixed-sized set of candidates—a committee—as the election outcome. This type of elections are known as *multiwinner elections* (in the literature they are also sometimes called *committee elections*). The notion of multiwinner elections and committees was implicitly introduced by Fishburn [75] in the context of so-called *choice functions* although the committee size was not fixed then. Debord [43] and Felsenthal and Maoz [74] later introduced k-choice functions which always output a size-k committee.

In comparison to singlewinner elections, in which the most popular candidate should be the winner, for multiwinner elections there are several approaches of which committee might be considered the “best” winning committee for a given election. Depending on the specific application of multiwinner elections the properties a winning committee should have change fundamentally. Elkind et al. [51] distinguish between three kinds of multiwinner elections:

Excellence-Based Elections: (Used for short-listing candidates for awards or job positions.) The winning committee should contain the most popular or highest-rated candidates.

Selecting a Diverse Committee: (Used for choosing items to display on a storefront or offer to a group of people.) The chosen candidates should be as diverse as possible.

Proportional Representation: (Used for parliamentary elections.) We seek to choose candidates such that the different views of the voters are represented proportionally in the committee.

Under those aspects we must choose a (multiwinner) voting rule that delivers an appropriate committee for a given application. Categorizing and analyzing multiwinner voting rules under those aspects to be able to choose the right voting rule for the right task has been given much attention (see, e.g., the work of Elkind et al. [51], Aziz et al. [3], Kilgour, Brams, and Sanver [96], Faliszewski et al. [72], Skowron, Faliszewski, and Lang [146], and Skowron, Faliszewski, and Slinko [147]). Interestingly, an impossibility theorem similar to that of Gibbard and Satterthwaite for singlewinner voting rules [78, 142] can be formulated for multiwinner voting rules as well: Peters [129] showed that no multiwinner voting rule can simultaneously satisfy a weak proportionality property[8] and a weak form of strategy-proofness. Multiwinner voting rules usually fall into one of three categories: *Committee Scoring Rules* [147], in which voters' preferences are given as linear orders, *Approval-Based Counting Rules* [95], in which voters submit a subset of approved candidates, and *Condorcet-Inspired Rules* [4, 144]. We focus on multiwinner voting rules of the first category and refer to the corresponding literature for definitions of the other two.

Formally, a multiwinner election (C,V,k) is defined by a (singlewinner) election (C,V) with the set of m candidates C and the preference profile V of n voters augmented with a nonnegative integer k which is the size of the committee that we seek to elect. Given the committee size k, a multiwinner voting rule $\mathscr{E}$ is a function mapping each multiwinner election (C,V,k) to a nonempty family of size-k subsets of C, the winning committees of (C,V,k) under $\mathscr{E}$. We will now define the multiwinner voting rules that we focus on.

Single transferable vote (STV): Given the quota $q = \lfloor \frac{n}{k+1} \rfloor + 1$, we choose candidates for a winning committee iteratively as follows. We compute plurality scores and if some candidate reaches the quota, we add her to the committee and remove q voters that vote for her. If no candidate reaches the quota, we remove the candidate with the lowest plurality score from the election (i.e., we remove her from all preference orders in the preference profile and the voters voting for this candidate now transfer their vote to the second highest candidate in their preference order).

An important issue is how ties are handled especially since we might need to break ties between voters in cases when a candidate has more than q voters voting for him but only q of them are removed. Conitzer, Rognlie, and Xia [34] devised a very fair tie-breaking scheme called *parallel-universes tiebreaking* (PUT) for which a committee is winning under STV if there exists a series of choices, breaking ties, such that the candidates of the committee are chosen by using STV. Sadly, using this tie-breaking method makes determining whether a committee is winning (see the corresponding decision problem below) intractable [34].

Single nontransferable vote (SNTV): Choose k candidates with highest 1-approval score.

Bloc: Choose k candidates with highest k-approval score.

k-Borda: Choose k candidates with highest Borda score.

$\mathscr{E}$-Chamberlin–Courant ($\mathscr{E}$-CC): Given a scoring rule $\mathscr{E}$, each committee is assigned a score by each voter that is the score under scoring rule $\mathscr{E}$ that the highest-ranked member of the committee in the voter's preference order would receive from the voter. The committee(s) with the highest overall score, summed up over all voters, are winning. We focus on **k-approval-CC** and **Borda-CC** which use k-approval and Borda scores, respectively.

For all $\mathscr{E}$-CC rules, the winner determination problem (defined below) is intractable [103, 132] but it is in FPT if parameterized by the number of candidates or voters [17].

Regarding computational considerations a central problem is the *winner determination* problem which was studied for various multiwinner voting rules by Aziz et al. [5], Procaccia, Rosenschein, and Zohar [131] and Baumeister, Dennisen, and Rey [40].

[8] Proportionality for multiwinner voting rules roughly means that a produced winning committee must represent the voters' preferences proportionally.

$\mathscr{E}$-WINNER DETERMINATION	
Input:	A multiwinner election (C,V,k) and a size-k committee $C' \subseteq C$.
Question:	Is C' a winning committee of (C,V,k) under $\mathscr{E}$?

The study of election tampering attempts in multiwinner elections was initiated by Meir et al. [113] focusing on strategic voting and proceeded by Aziz et al. [5], Obraztsova, Zick, and Elkind [123], and Baumeister, Dennisen, and Rey [40]. Bredereck et al. [27, 29] and Faliszewski et al. [73] studied bribery in multiwinner elections. In Section 5 we will extend the model of electoral control by cloning candidates for singlewinner elections that was introduced above to the multiwinner setting and study it for the multiwinner voting rules above.

CHAPTER 3

CONTROL COMPLEXITY IN BORDA ELECTIONS: SOLVING ALL OPEN CASES OF OFFLINE CONTROL AND SOME CASES OF ONLINE CONTROL

3.1 Summary

The Borda Count is one of the most important voting rules which finds applications not only in voting settings but also can be used for the allocation of indivisible goods [21] (for a thorough introduction to the field of fair division see, e.g., the book chapter by Lang and Rothe [100]) and hedonic games [138] (an introduction to hedonic games can be found in, e.g., the book chapter by Elkind and Rothe [54]). We first survey recent research in all three fields relating to Borda.

Then, we study electoral control for Borda. Especially the electoral control problems involving partitioning the set of candidates or voters were largely unexplored for Borda: Out of the twelve cases only one case (namely, Borda-DCPV-TE) was solved by Russel [140]. In particular, we solve all open cases of classical electoral control introduced by Bartholdi, Tovey, and Trick [10] and Hemaspaandra, Hemaspaandra, and Rothe [83] for Borda showing that Borda is resistant to all cases of constructive control and vulnerable to all but three cases of destructive control. We obtain our results for both winner models and also found two of the rare cases, namely destructive control by partition and by run-off partition of candidates with ties-promote tie-breaking, for which the complexity changes depending on which winner model is assumed.

Lastly, we study the model of online control, which was introduced by Hemaspaandra, Hemaspaandra, and Rothe [86, 87], showing that Borda is vulnerable against constructive and destructive online control by adding or deleting candidates and resistant against all types of online voter control (to be precise, we show coNP-hardness results for all cases).

3.2 Publication – Neveling and Rothe [120]

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

Preliminary versions of this paper were published in the proceedings of the *31st and the 33rd AAAI Conference on Artificial Intelligence* (AAAI'17 and AAAI'19, see [118, 137]) and of the *18th Italian Conference on Theoretical Computer Science* (ICTCS'17, see [117]).

3.3 Personal Contribution

The writing was done jointly with Jörg Rothe. All technical results are my contribution. Parts of this work already appeared in my Bachelor's and Master's Thesis. Specifically, Theorem 1 was part of my Bachelor's Thesis and Theorem 3, 10 and 13 were in my Master's Thesis.

Artificial Intelligence 298 (2021) 103508

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Control complexity in Borda elections: Solving all open cases of offline control and some cases of online control ☆

Marc Neveling, Jörg Rothe *

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

A R T I C L E I N F O

Article history:
Received 23 March 2019
Received in revised form 13 April 2021
Accepted 16 April 2021
Available online 21 April 2021

Keywords:
Borda election
Voting
Computational social choice
Computational complexity
Control
Online control

A B S T R A C T

Borda Count is one of the earliest and most important voting rules and has been central to many applications in artificial intelligence. We study the problem of control in Borda elections where an election chair seeks to either make a designated candidate win (constructive case), or prevent her from winning (destructive case), via actions such as adding, deleting, or partitioning either candidates or voters. These scenarios have been studied for many voting rules and the related control problems have been classified in terms of their computational complexity. However, for one of the most prominent natural voting rules, the Borda Count, complexity results have been known for only half of these cases until recently. We settle the complexity for all missing cases, focusing on the unique-winner model. We also exhibit two of the very rare cases where the complexity of control problems differs depending on the winner model chosen: For destructive control by partition and by run-off partition of candidates when ties promote, Borda is resistant in the unique-winner model (i.e., these two control problems are NP-hard), yet is vulnerable in the *non*unique-winner model (i.e., one can decide in polynomial time whether control is possible). Finally, we turn to the model of online control in sequential elections that was recently proposed by Hemaspaandra et al. [62,61]. We show that sequential Borda elections are vulnerable to constructive and destructive online control by adding or deleting candidates, whereas we obtain coNP-hardness results for all types of online voter control in sequential Borda elections.

1. Introduction

More than 230 years ago, Borda [15] proposed one of the most important and influential voting rules up to date. It is simple and strikingly elegant: When there are m candidates, the voters rank them by a linear order according to their preferences; a candidate in ith position of a voter's ranking scores $m - i$ points; and the candidates with the most points win.

Borda and its modifications have been widely used in political elections (e.g., in Slovenia or to elect the leader of the Irish Green Party) or by academic institutions. For instance, the French Academy of Sciences adopted this rule to elect its

☆ This paper combines and extends a series of previous papers that appeared in the proceedings of the *31st* and the *33rd AAAI Conference on Artificial Intelligence* (AAAI'17 and AAAI'19, see [80,91]) and of the *18th Italian Conference on Theoretical Computer Science* (ICTCS'17, see [79]) and were presented at the *7th International Workshop on Computational Social Choice* (COMSOC'18, with nonarchival proceedings).

* Corresponding author.
E-mail addresses: marc.neveling@hhu.de (M. Neveling), rothe@hhu.de (J. Rothe).

https://doi.org/10.1016/j.artint.2021.103508

members for about two decades in the 18th century. The debates between the Chevalier de Borda and the Marquis de Condorcet, both members of this Academy, about whose voting method is better are legendary. Social choice theorists have continued to fiercely dispute this question up to now; for example, Saari [95,96] champions Borda, a view that Risse [89] strongly disagrees with. Our goal, however, is not a social-choice-theoretic treatise of Borda compared with other voting rules[1]; rather, our goal is to present recent advances related to Borda within the field of *computational* social choice as a subarea of collective decision making. To provide some motivation of our work, let us start by giving an overview of some of the most exciting related work in computational social choice—having evolved to become an established subarea of distributed AI and multiagent systems—during the last decades. In particular, we will present the standard attacks that have been proposed within computational social choice to model strategic behavior in voting, such as manipulation, control, and bribery attacks of various types.

1.1. Motivation and related work

Some of the most exciting work in computational social choice of the previous decades is the study of strategic behavior in voting, and how computational complexity can be used as a barrier against such attacks that aim at influencing the outcome of an election (notwithstanding the well-known and intensively debated limitations of worst-case complexity in this regard, as surveyed by Rothe and Schend [93]). Three basic attack types are distinguished in the literature: manipulation, control, and bribery. While we will focus on Borda's voting rule to illustrate such attacks and to discuss how resistant Borda is to them in terms of computational complexity, we will also mention in passing some related results for other voting rules. For a comprehensive overview of related results, we refer to the book chapters and surveys by Conitzer and Walsh [25], Faliszewski et al. [50,42,45,48], Baumeister et al. [11,9], and Chevaleyre et al. [23].

1.1.1. Manipulation in Borda elections

Consider the following example.

Example 1. Suppose we are given an election (C, V) with five candidates, $C = \{a, b, c, d, e\}$, and the following list of five votes in V, each cast by an honest voter:

$$\begin{array}{llllll} v_1: & d & c & a & e & b \\ v_3: & d & b & e & a & c \\ v_5: & c & b & d & a & e \end{array} \qquad \begin{array}{llllll} v_2: & d & c & b & a & e \\ v_4: & e & c & b & a & d \\ & & & & & \end{array}$$

where a vote like v_1's ($d\ c\ a\ e\ b$) means that v_1 prefers d to c, c to a, etc. Using Borda in (C, V), a scores 6 points, b 10 points, c 13 points, d 14 points, and e 7 points, so d alone wins. Now suppose that an insincere voter, v_6 whose truthful vote is $c\ d\ b\ a\ e$, joins the election. Knowing the other voters' preferences, however, v_6 strategically casts the vote $c\ a\ e\ b\ d$, so c alone wins the election $(C, V \cup \{v_6\})$ with a score of 17, while a, b, d, and e now have 9, 11, 14, and 9 points, respectively. (Casting v_6's truthful vote would have made both c and d win the election with 17 points, but v_6 wants to make sure that her favorite candidate c is the only winner.) Thus v_6 has successfully manipulated the election.

Motivated by a famous result of Gibbard [53] and Satterthwaite [97] (which, roughly speaking, says that every reasonable voting rule is manipulable), Bartholdi et al. [6] proposed to use computational complexity to prevent manipulation from happening or being successful. They defined the *constructive manipulation* problem (CM): Given an election (C, V), a distinguished candidate $c \in C$, and a strategic voter s,[2] is it possible for s to cast a vote such that c is the winner of the election $(C, V \cup \{s\})$? For Borda, though, they showed that this problem is easy to solve: A simple greedy algorithm solves CM for Borda in polynomial time. In fact, this greedy algorithm works for every voting rule that can be represented by a scoring function that is both responsive (i.e., candidates with the highest score win) and monotonic (i.e., moving a candidate to a better position in a preference ranking cannot result in this candidate scoring fewer points). On the other hand, they showed that another voting rule, "second-order Copeland," resists manipulation in the sense that CM for it is NP-complete. Bartholdi and Orlin [5] established that for the voting rule STV ("single transferable vote") CM is NP-complete as well.[3]

Conitzer et al. [24] generalize CM in two ways: First, they allow voters to be *weighted* and, second, they consider *coalitions* of manipulators, leading to the *constructive coalitional weighted manipulation* problem (CCWM), where the weights and preferences of the honest voters but only the manipulators' weights are given (and the manipulators are free to choose their preferences strategically). For instance, suppose that, in the election from Example 1, $v_1, \dots, v_4$ are honest voters with weight 2 each, whereas v_5 and v_6 both have weight 1 and form a coalition of manipulators who wish to make their favorite candidate, c, win. Casting their truthful votes, $c\ b\ d\ a\ e$ for v_5 and $c\ d\ b\ a\ e$ for v_6, would result in d (with 29 points) beating c (with 26 points) and also a, b, and e (with even fewer points). But if v_5 and v_6 cast strategic votes (e.g.,

[1] For more details about the social-choice-theoretic properties of Borda and other voting rules, we refer to the recent book chapters by Zwicker [102] and Baumeister and Rothe [11].

[2] We often use "voter" and "vote" synonymously, i.e., we often identify a voter s with the vote cast by s.

[3] Rothe and Schend [93] note that the reduction of Bartholdi and Orlin [5] is slightly flawed but can be easily fixed.

both $c\ a\ b\ e\ d$), they would make c (still with a score of 26 points, but d now scoring only 24 points and a, b, and e even less) the Borda winner of the election.

Conitzer et al. [24] show that CCWM is (weakly) NP-complete for Borda even when there are only three candidates (and is in P for up to two candidates). Similar results for CCWM have been obtained for many other voting rules, such as plurality, veto, Copeland, maximin, and STV [24], Baldwin's and Nanson's variants of Borda [28] that we will again consider later on in Section 1.1.4, plurality with run-off and veto with run-off [40], and Bucklin and fallback voting [49] (we omit defining all these rules explicitly here, but refer to the book chapters by Conitzer and Walsh [25] and Baumeister and Rothe [11]). An interesting special case occurs for the Copeland rule[4] with three candidates: While CCWM for it is in P for the unique-winner model (which requires the distinguished candidate c to be the only winner for the manipulation attack to be successful), Faliszewski et al. [47] show that this problem is NP-complete in the nonunique-winner model (where the manipulation attack is considered successful even if c is one among several winners). The known complexity results for CCWM with respect to all other voting rules considered are the same in both winner models.

Borda is a very prominent member of a whole class of important voting rules, the so-called scoring protocols, that also contains plurality and veto. A *scoring protocol for m candidates* is defined by a scoring vector $\sigma = (\sigma_1, \ldots, \sigma_m)$ of nonnegative integers, $\sigma_1 \geq \cdots \geq \sigma_m$, where a candidate in the ith position of a vote gets σ_i points, and whoever has the most points wins. Borda is thus defined via $(m-1, m-2, \ldots, 0)$, plurality via $(1, 0, \ldots, 0)$, and veto via $(1, \ldots, 1, 0)$. Hemaspaandra and Hemaspaandra [57] established the following dichotomy result (which for the case of three candidates was also observed by Conitzer et al. [24]): CCWM is in P for plurality and the trivial scoring protocol with vector $(0, \ldots, 0)$, and is NP-complete for all other scoring protocols.

So far we have considered only the *constructive* case where the goal of the manipulator(s) is to make a given candidate win. Conitzer et al. [24] were the first to define the *destructive* variant where the goal is to block a given candidate's victory. The destructive analogue of CCWM, denoted by DCWM, has also been studied for most of the voting rules mentioned above. It turns out that DCWM is never harder than CCWM, but it can be easier. For example, Conitzer et al. [24] showed that DCWM for Borda is in P, and this holds true for each voting rule that can be represented by a scoring function that is both responsive and monotonic and whose winners can be determined in polynomial time. By contrast, they also showed that DCWM for STV (which is not monotonic) is NP-complete even for three candidates.

Coming back to constructive manipulation for Borda, we have seen that CCWM is NP-hard, yet CM is easy to solve. But what about the intermediate case, the case where voters are *unweighted* and still there is a *coalition of manipulators*? Denote this problem by CCUM. Its complexity for Borda has been a mystery for several years. Then, in 2011, two papers resolved this open question independently at about the same time: Betzler et al. [13] and Davies et al. [27] (see also [28]) showed that CCUM is NP-complete even when there are only two manipulators. Indeed, this was one of the greatest moments in computational social choice: Betzler et al. [13] presented their work in the IJCAI 2011 Distinguished Papers session, and Davies et al. [27] were honored by an AAAI 2011 Outstanding Paper Award.

Zuckerman et al. [101] considered an optimization variant of CCUM, denoted by CCUO: Given the unweighted votes of sincere voters and a distinguished candidate c, determine the minimum number of manipulators needed in order to make c win. They designed an efficient algorithm that approximates CCUO for Borda up to an additive error of one. They also studied the weighted variant, noting that a shortcoming of NP-hardness results is that they are worst-case complexity results only, thus providing a "poor obstacle against potential manipulators," as these may still be able to succeed in typical settings. Instead, Zuckerman et al. [101] took a different approach: They designed efficient heuristics, characterized "small windows" of instances where these may fail, and proved that they are correct on all other instances. For Borda, they showed that if there is a manipulation for an instance with certain weights, their heuristics will succeed when given an extra manipulator with maximal weight. Rothe and Schend [93] survey this and other approaches to dealing with challenges to complexity shields that are supposed to protect elections against manipulative attacks.

1.1.2. Control in Borda elections

While one may feel a bit uneasy about manipulators strategically changing the outcome of an election, there is actually not much one could put forward against it. After all, every voter—human or software agent—has the right to think strategically about which vote to cast; not doing so would not be smart. Electoral control, however, is better suitable than manipulation as a model of electoral fraud or vote rigging—in the sense of acts that are considered ethically unacceptable, outside the spirit of an election, or in violation of the principles of democracy. Here we assume that an external authority, called the *(election) chair*, seeks to influence the outcome of an election via exerting certain control actions. Bartholdi et al. [7] were the first to introduce control attacks (such as *constructive control by deleting voters*) and their associated decision problems (CCDV): Given an election (C, V), a distinguished candidate $c \in C$, and a nonnegative integer k, is it possible for the chair to make c win by deleting up to k votes from V? For example, in the election (C, V) considered in Example 1, with $C = \{a, b, c, d, e\}$ and $V = (v_1, \ldots, v_5)$, we have seen that d is the only Borda winner. However, by deleting just one voter, namely v_3, the chair can ensure that now c alone wins.

[4] In a Copeland election [26], each candidate who is preferred to another candidate by a majority of voters earns one point in this head-on-head contest; for each tie, they both earn half a point; and the candidates with the most points win.

The other control actions/problems studied by Bartholdi et al. [7] (some of which will be defined in detail in later sections) are *constructive control by adding voters* (CCAV), *constructive control by partition of voters* (CCPV), *constructive control by deleting candidates* (CCDC), *constructive control by adding an unlimited number of candidates* (CCAUC), *constructive control by adding candidates* (CCAC), *constructive control by partition of candidates* (CCPC), and *constructive control by run-off partition of candidates* (CCRPC). Each such control type captures a particular way of rigging elections. For example, CCDV models voter disenfranchisement; by adding spoiler candidates in CCAC, the chair seeks to weaken the rivals of her favorite candidate; and partitioning of voters in CCPV (which is formalized as a two-stage election where the electorate is partitioned to create two subelections whose winners face each other in the final run-off) is a (rather simple) model of *gerrymandering*, a common practice to achieve an advantage from suitably shifting the boundaries of voting districts.[5] Similar scenarios motivate the other control types.

The control-by-partition cases come in two variants each by using a rule that specifies how to handle ties in their first-stage subelections: Either all subelection winners move forward to the final run-off (*ties promote*, TP) or only unique subelection winners move forward (*ties eliminate*, TE). This distinction is due to Hemaspaandra et al. [59], who also introduce the *destructive* analogues of these control types: DCDV, DCAV, DCPV, DCDC, DCAC, DCAUC, DCPC, and DCRPC.[6]

Bartholdi et al. [7] classified the complexity of the constructive control problems for Condorcet voting and plurality; Hemaspaandra et al. [59] did so for their destructive variants, and also for constructive and destructive control in approval voting; Faliszewski et al. [44] studied the complexity of control for Copeland$^{\alpha}$[7]; Menton [76] and Erdélyi et al. [39] for certain variants of approval and range voting; Parkes and Xia for Schulze voting [84]; Erdélyi et al. [37] for Bucklin and fallback voting; and Maushagen and Rothe [75] for veto. Interestingly, unlike for manipulation, some voting rules are *immune* to certain control actions, which means that it is never possible for the chair to reach her goal. For example, Condorcet is immune to constructive control by adding candidates and to destructive control by deleting or partitioning candidates, and the same applies to approval and range voting. If a voting rule is *not* immune to a control type, it is *susceptible* to it, and in this case it makes sense to determine the complexity of the corresponding control problem. Among natural voting rules with a winner problem in P, normalized range voting [76] and fallback voting [37] display the broadest resistance (in the sense of NP-hardness) to control currently known: They are vulnerable (i.e., the associated control problem is in P) to only two control types (DCDV and DCAV) and resistant in all other cases. (We omit stating all related results explicitly but instead refer to the book chapters by Faliszewski and Rothe [50] and Baumeister and Rothe [11] for an overview.) How protective are NP-hardness results (showing hardness merely in the worst case) as shields against control attacks? Various approaches have been proposed to challenge such NP-hardness shields to control and, again, Rothe and Schend [93] survey these approaches and discuss how to deal with such challenges.

Summing up, while the above control scenarios have been studied intensively for many voting rules (including some exotic ones that are rarely used in practice), one of the most prominent natural voting rules, the Borda Count, has still been heavily underexplored until recently: The nine results previously known for control in Borda (marked in gray in Table 1 on page 6) are due to Russel [94], Elkind et al. [34], Loreggia et al. [73], Chen et al. [22], and Hemaspaandra and Schnoor [64]. The purpose of this paper is to fill this glaring gap, as will be outlined in Section 1.2 and will be done in full technical detail in Sections 3 and 4.

1.1.3. Online control in sequential Borda elections

In a predecessor of this work [80], we have also considered *online candidate control* in *sequential* Borda elections—a dynamic, partial-information model due to Hemaspaandra et al. [61] where the candidates show up in sequence, one after the other, the votes being gradually extended to add the current candidate in each step, and the chair must decide *right now* whether or not to exert the given control action (e.g., to either delete the current candidate now or never). That is, the chair has a "use-it-or-lose-it ability" to exert control. Extending the corresponding results for sequential plurality due to Hemaspaandra et al. [61], we show that sequential Borda is vulnerable to online constructive and destructive control by either adding or deleting candidates [80]. An obvious question related to this result (which will be presented in Section 5) is what can be shown for online *voter* control in sequential Borda elections, according to the model introduced by Hemaspaandra et al. [62].

1.1.4. Bribery in Borda elections

Another way to fiddle around with elections so as to change their outcome to one's own advantage is bribery, a model proposed by Faliszewski et al. [41] (see also [44]): A *briber* seeks to influence the outcome of an election by bribing certain

[5] To take certain restrictions (e.g., geographical constraints) into account, other models of gerrymandering and of control were considered as well, e.g., by Puppe and Tasnádi [86], Erdélyi et al. [38], Lewenberg and Lev [71], and Bachrach et al. [3].

[6] Hemaspaandra et al. [58] observed that, depending on the tie-handling rule (TP vs. TE) and the winner model (nonunique vs. unique winner) used, DCPC and DCRPC can be the same problem (see Fact 1 in Section 2). The difference between control by partition and by run-off partition of candidates is that in the latter the winners of both subelections run against each other in the final run-off, whereas in the former the winners of one subelection face all candidates of the other subelection in the final round.

[7] Copeland$^{\alpha}$ generalizes Copeland by rewarding each tie between candidates in a head-on-head contest with α points, $\alpha \in \mathbb{Q} \cap [0,1]$: Ties in Copeland^{0} are not rewarded at all; Copeland$^{0.5}$ is the common Copeland rule (see Footnote 4); and Copeland^{1}, which was proposed by Ramon Llull as early as 1299, treats ties just as wins in pairwise contests.

voters without exceeding a given budget. Bribery shares certain features with manipulation and others with control, e.g., the briber is an external actor who needs to choose which votes to affect as in control, and as in manipulation the briber needs to find suitable preference orders when changing these bribed votes. In the decision problem associated with the most basic variant of bribery, denoted by Bribery, we are given an election (C, V), a distinguished candidate $c \in C$, a budget $B \in \mathbb{N}$, and a collection $(\chi_1, \dots, \chi_n)$ of cost functions, one for each voter. For each i, $1 \le i \le n$, and each preference order over C, χ_i gives the cost of convincing the ith voter to cast this preference order instead of her original one, where we assume that keeping the original vote always has zero cost. We ask whether there is a preference profile V' such that c wins in (C, V') and the sum of the costs of changed votes doesn't exceed B. While "bribery" commonly has a rather negative connotation, it can also be positively interpreted, as, e.g., Faliszewski et al. [49] do, in terms of "campaign management" where the manager of a political campaign seeks to convince voters to change their votes and these efforts have certain costs.

For an example, look again at the election (C, V) considered in Example 1, with $C = \{a, b, c, d, e\}$, $V = (v_1, \dots, v_5)$, and the Borda winner d scoring 14 points. Assume that a (with 6 points currently) is our distinguished candidate, the budget is 2, and all voters have unit cost. Then a can be made a unique Borda winner by bribing, e.g., v_2 and v_3 to change their votes to $a\ e\ b\ c\ d$ and $a\ e\ b\ d\ c$, yielding a score of 12 for a, of 9 for b, of 11 for c, of 7 for d, and of 11 for e. If the budget were 1, though, no bribery action would be successful, as by bribing only one voter, a could gain only 3 points (giving a score of at most 9), but d could lose no more than four points (giving a score of at least 10).

Among many other results, Faliszewski et al. [41] established a dichotomy result in the class of scoring protocols: For each $\sigma = (\sigma_1, \dots, \sigma_m)$, if $\sigma_2 = \cdots = \sigma_m$ then the *weighted* variant of Bribery for σ is in P; otherwise, it is NP-complete. In particular, weighted Bribery for Borda with three or more candidates is NP-complete. Brelsford et al. [21] proved that even in the unweighted case, Bribery for Borda is NP-complete and also provided an inapproximability result for bribery.

Elkind et al. [33] defined another variant of the bribery problem, denoted Swap-Bribery (which generalizes the manipulation problem CCUM considered earlier), where the briber has to pay for each individual swap of adjacent candidates in the votes separately. They showed that Swap-Bribery for Borda (and many other voting rules) is NP-complete. That was why they also introduced the special variant Shift-Bribery, which is defined like Swap-Bribery except that each swap must involve the distinguished candidate. Still, they showed that Shift-Bribery for Borda is NP-complete, yet can be efficiently approximated to within a factor of 2 (which was generalized by Elkind and Faliszewski [32] for all scoring protocols). Recently, Maushagen et al. [74] studied the complexity of Shift-Bribery for *iterative* voting rules such as those by Baldwin [4] and Nanson [78] (which, as explained below, are variants of Borda), and showed that they are NP-complete as well. These two voting rules proceed in rounds and eliminate in each round the candidates performing worst (namely, the candidates with the lowest Borda score in Baldwin and those with scores lower than the average Borda score in Nanson), and the remaining candidates win.

The complexity of bribery has been studied for many other voting rules as well; for instance, Faliszewski et al. [44] studied bribery in Copeland$^\alpha$ elections. We again omit stating all these results and papers, referring to the book chapters by Faliszewski and Rothe [50] and Baumeister and Rothe [11] instead.

1.2. Our contribution

As pointed out above, a large variety of manipulation and bribery scenarios have been comprehensively studied for Borda elections. But what about the control complexity in Borda elections? A closer look at the two book chapters just mentioned reveals that only nine of the many control scenarios had been solved for Borda by 2016, scattered results in as many as five different papers (by contrast, the control complexity of other voting rules was typically dealt with in just one paper each). Our main contribution is to *systematically* study the control complexity of Borda and to settle all the other cases of control for Borda[8]; in particular, the technically quite demanding partition-of-candidates/voters cases that were still open.

Table 1 gives an overview of the control complexity in Borda elections in the unique-winner model; previously known results are marked in gray and our new results are marked in boldface.[9] In this table, an "R" stands for *resistance* (which in Section 2 will be defined as NP-hardness of the corresponding control problem) and "V" stands for *vulnerability* (which in Section 2 will be defined as polynomial-time solvability of the corresponding control problem). Further, we use the standard names of the control problems that correspond to the standard control scenarios (see, e.g., [11,50] and Section 1.1.2). For example, CCDC stands for "constructive control by deleting candidates" and DCDC denotes the destructive variant of this problem. Each control problem for which we provide a new result in Borda elections will be formally defined in Sections 3 and 4, and the unique-winner versus the nonunique-winner model will be discussed in Section 2. As Table 1 shows, Borda is now known to be resistant to every standard type of constructive control, whereas it is vulnerable to most of the destructive control types; resistance is known only for destructive control by (run-off) partition of candidates and by partition of voters, each in the so-called "ties-promote" (TP) model to be formally defined in Section 3.

Interestingly, we show that Borda is vulnerable to destructive control by partition of candidates with TP in the *non*unique-winner model (Theorem 8). A consequence of this result (Corollary 2) is that Borda-DCRPC-TP (which is known

[8] All of these except one (namely, Theorem 6) could be settled already in our two conference papers [80,79] from which the present work emerged. However, we here provide all proofs in full technical detail, give more examples for illustration, and we comprehensively extend our discussion.

[9] NP-completeness of CCDV has been shown in our previous paper [80] as well; however, as we learned later, this already follows from a dichotomy result of Hemaspaandra and Schnoor [64] for this problem in the class of scoring protocols.

Table 1
Control complexity in Borda elections (unique-winner model), with standard notation of control types [11,50]. “R” means resistance and “V” vulnerability.

	CCAUC	CCAC	CCDC	CCPC-TE	CCPC-TP	CCRPC-TE	CCRPC-TP	CCAV	CCDV	CCPV-TE	CCPV-TP
Borda	**R**†	R$^\$$	R$^❄$	**R**$^\clubsuit$	**R**$^\circ$	**R**$^\diamondsuit$	**R**$^\triangle$	R§	R$^\spadesuit$	**R**$^\heartsuit$	**R**$^\oplus$

(a) Constructive control. **New results are in boldface: Theorem 1 (†), Theorem 3 (◇), Theorem 4 (♣), Theorem 9 (∘), Corollary 3 (△), Theorem 10 (♡), and Theorem 11 (⊕).** Previously known results are gray and due to Russel [94] (marked by §), Elkind et al. [34] ($), Chen et al. [22] (❄), and Hemaspaandra and Schnoor [64] (♠).

	DCAUC	DCAC	DCDC	DCPC-TE	DCPC-TP	DCRPC-TE	DCRPC-TP	DCAV	DCDV	DCPV-TE	DCPV-TP
Borda	**V**‡	V$^£$	V$^£$	**V**$^\natural$	**R**$^\sharp$	**V**¶	**R**$^\square$	V§	V§	V§	**R**$^\otimes$

(b) Destructive control. **New results are in boldface: Theorem 2 (‡), Theorem 5 (♮), Corollary 1 (¶), Theorem 6 (♯), Theorem 7 (□), and Theorem 12 (⊗).** Previously known results are gray and due to Russel [94] (marked by §) and Loreggia et al. [73] (£).

to coincide with Borda-DCPC-TP in the nonunique-winner model [58]) is in P as well. By contrast, both Borda-DCPC-TP and Borda-DCRPC-TP (which differ in the unique-winner model; see Footnote 15) are NP-complete by Theorems 6 and 7. These are two of the rare cases where the complexity of control parts company depending on the winner model chosen.

Regarding online control in sequential Borda elections, in Section 5 we will show vulnerability for constructive and destructive online control by adding and by deleting candidates. As mentioned in Section 1.1.3, these results were contained in a predecessor of this work [80] already. By contrast, we will show coNP-hardness for constructive and destructive online control by adding voters, by deleting voters, and by partitioning voters (where coNP is the class of complements of NP sets). These results are new and were not contained in this predecessor [80].

1.3. A look on using Borda beyond voting

Borda Count has been used not only in voting but also to maximize social welfare when indivisible goods are to be allocated to agents with ordinal preferences [18,19,17,8,81] and for the purpose of coalition formation in hedonic games [69,92]. All three fields—preference aggregation by voting, the allocation of indivisible goods, and hedonic games—are central to certain AI applications. Voting, for example, has been employed in AI subareas as diverse as automated scheduling [56], collaborative filtering [85], computational linguistics [82], information extraction [98], planning [36], recommender systems [52], and web searching [31]. In the following, we very briefly survey some recent results on how to use Borda in the allocation of indivisible goods and in hedonic games.

1.3.1. Borda-optimal allocation of indivisible goods

Allocating indivisible goods to agents having preferences over (bundles of) goods is an important field at the intersection of AI and economics. There is substantial literature on the allocation of indivisible goods, which cannot all be cited here; instead we refer to the book chapters by Bouveret et al. [16] and Lang and Rothe [70].

Let $N = \{1, \dots, n\}$ be a set of agents and G a set of m goods. An *allocation of G to N* is a partition $(\pi_1, \dots, \pi_n)$ of G (i.e., $G = \bigcup_{i=1}^{n} \pi_i$ and $\pi_i \cap \pi_j = \emptyset$ for $i \neq j$), where π_i is the bundle of goods assigned to agent i. A common approach to how agents value their bundles is to assume additive preferences: Every agent i assigns a positive number to each good and i's utility for a bundle of goods is the sum of the corresponding numbers. Here, however, we take a different approach: We assume that agents have *ordinal* preferences over G, i.e., a ranking of the goods, and the agents' utilities are now specified by a fixed, agent-independent vector that maps ranks into scores just as in voting. In particular, Brams et al. [18] (and later Brams and King [19] and Bouveret et al. [17]) studied *Borda-optimal allocations*.[10] Baumeister et al. [8] generalized these by introducing *scoring allocation correspondences*, which informally stated proceed in three steps: First, we use a scoring vector $\sigma = (\sigma_1, \dots, \sigma_m)$ to derive from the agents' preferences a utility vector for each possible allocation π, thus specifying each agent's individual utility for π. Second, these individual utilities are aggregated via an *aggregation function* (typically, via utilitarian or egalitarian social welfare, i.e., by using the sum or the minimum of the agents' individual utilities) to obtain the collective utility of π. Third, we choose the outcomes π that maximize collective utility. (If desired, one can break ties so as to yield a *scoring allocation rule*, which always outputs only one allocation π maximizing collective utility.)

Brams et al. [18] study properties of Borda-optimal allocations such as *envy-freeness* (i.e., no agent wants to swap her bundle) and *Pareto optimality* (where an allocation is Pareto-optimal w.r.t. the agents' preferences if no other allocation

[10] Unlike for Borda in voting, they here use the scoring vector $(m, m-1, \dots, 1)$ to ensure that each good has some positive value. In voting, such a shift of scores would not matter [57]. For the allocation problem, however, Baumeister et al. [8] show that such a shift actually does matter.

can make some agent better off without making some other agent worse off). For example, they show that Borda-optimal allocations (w.r.t. the "leximin order") are always *possibly* Pareto-optimal, whereas they in general fail to be *necessarily* Pareto-optimal—notions that are closely related to the notions of possible and necessary winner in voting due to Konczak and Lang [66].

Baumeister et al. [8] study both axiomatic and computational properties of scoring allocation correspondences and rules. For example, they show that Borda scoring (with egalitarian social welfare and any tie-breaking relation) satisfies monotonicity, yet does not satisfy what they call "global monotonicity," "possible object monotonicity," and "separability." And that, given the agents' preferences, the problem of whether there is an allocation whose egalitarian social welfare exceeds a given value is NP-complete for Borda. Nguyen et al. [81] characterize *strategy-proofness*, as defined by Kelly [65], for scoring allocation correspondences with utilitarian social welfare. For Borda, their result implies that strategy-proofness holds if and only if there are no more than two goods. Kuckuck and Rothe refute a conjecture of Baumeister et al. [8] on *separability* of sequential allocation rules [67] and they study *duplication monotonicity* of scoring allocation rules [68], a notion inspired by the twin paradox [77] in voting and by false-name manipulation in weighted voting games [1,87].

1.3.2. Forming coalitions in Borda-induced FEN-hedonic games

Hedonic games, as part of cooperative game theory, model how players, each having preferences about the coalitions they can join, form coalitions. For more background, we refer to the book chapters by Aziz and Savani [2] and Elkind and Rothe [35]. One problem is how to represent hedonic games, given that each player can join exponentially many (in the number of players) coalitions. Lang et al. [69] list a number of approaches from the literature for how to deal with this problem, e.g., the *friend-* and *enemy-oriented encodings* due to Dimitrov et al. [30]. Lang et al. [69] extend their approach by also allowing *neutral* players and define *FEN-hedonic games*: Each player partitions the set of other players into friends, enemies, and neutral players and ranks her friends and enemies. They assume preferences to be monotonic w.r.t. adding friends and antimonotonic w.r.t. adding enemies and then use "bipolar responsive extensions" to lift the players' rankings of players to their partial preferences over coalitions. Rothe et al. [92] (see also Section 5.2 in the conference version by Lang et al. [69]) then employ *cardinal comparability functions* based on scoring vectors so as to extend partial to complete preference orders consistent with these bipolar responsive orders. Focusing on *Borda-induced FEN-hedonic games*, they study the complexity of the existence and the verification problem for common solution concepts; e.g., verifying "Nash stability" is in P and testing if a Nash-stable coalition structure exists is NP-complete, while verifying "core stability" is coNP-complete and testing if a core-stable coalition structure exists is $\mathrm{NP}^{\mathrm{NP}}$-complete (where $\mathrm{NP}^{\mathrm{NP}}$, the second level of the polynomial hierarchy, is the class of problems that can be solved by an NP oracle machine accessing an NP oracle; see, e.g., [83,90]).

1.4. Outline

In Section 2, we provide some notation and technical preliminaries. Section 3 is devoted to candidate control and Section 4 to voter control in Borda elections. We then turn to online control in sequential Borda elections in Section 5 before we conclude in Section 6 with outlining some open issues and possible directions for future work.

2. Preliminaries

In this section, we present our notation and technical preliminaries and give the needed background from social choice theory, computational social choice, and computational complexity.

2.1. Notation for elections

An *election* is a pair (C, V) that contains a set C of candidates and a list V of votes (a.k.a. a preference profile) describing the voters' preferences—as strict linear orders—over the candidates. We will represent a vote over C as a string that ranks the candidates from left (most preferred) to right (least preferred); for example, if $C = \{a, b, c, d\}$, a vote $c\ d\ b\ a$ means that this voter prefers c to d, d to b, and b to a. A *voting rule* determines a set of winners from each given election. Positional scoring rules are an important class of such rules, and among those we will only consider the perhaps most prominent one, the *Borda Count*, which works as follows: Given m candidates, every candidate in position i of the voters' rankings (where position 1 is the leftmost position in a vote) scores $m - i$ points, and all candidates scoring the most points win.

Let $score_{(C,V)}(x)$ denote the number of points candidate x obtains in a Borda election (C, V), and let

$$dist_{(C,V)}(x, y) = score_{(C,V)}(x) - score_{(C,V)}(y)$$

be the difference between the Borda scores of two candidates, x and y. For a subset $X \subseteq C$ of candidates, $\overrightarrow{X}$ in a vote denotes a ranking of these candidates in an arbitrary but fixed order, $\overleftarrow{X}$ denotes their ranking in reverse order, and we simply write X when the order of the candidates in X does not matter in this vote. For example, for $C = \{a, b, c, d\}$ and $X = \{b, d\}$ and assuming the lexicographic order of candidates, $c\ \overrightarrow{X}\ a$ denotes the vote $c\ b\ d\ a$, the vote $c\ \overleftarrow{X}\ a$ denotes $c\ d\ b\ a$, and $c\ X\ a$ could mean either of $c\ b\ d\ a$ and $c\ d\ b\ a$.

2.2. Some background from complexity theory

We assume that readers be familiar with the most basic notions of computational complexity. For our purposes, it will suffice to know the complexity classes P (deterministic polynomial time) and NP (nondeterministic polynomial time) as well as the notions of NP-*hardness* and NP-*completeness*, based on the *polynomial-time many-one reducibility*. Whenever we speak of a problem A being *reducible* to a problem B, we mean this notion of reducibility: There exists a polynomial-time computable, total function φ such that for each input a, $a \in A$ if and only if $\varphi(a) \in B$. For more background on complexity theory, the reader is referred to the standard textbooks by, e.g., Garey and Johnson [51], Papadimitriou [83], and Rothe [90].

2.3. Susceptibility, immunity, vulnerability, and resistance

The control types considered here will be formally defined in Sections 3 and 4, and we refer to the book chapters by Faliszewski and Rothe [50] and Baumeister and Rothe [11] (and to the references therein) for all other standard control types and for real-world scenarios that motivate them.

A voting rule is said to be *susceptible to a type of control* (e.g., constructive control by adding candidates) if there is some election for which the chair can reach her goal (e.g., turning a nonwinning candidate into a winner) by exerting this type of control. If a voting rule is not susceptible to a control type, it is said to be *immune to it*. Borda is susceptible to each standard control type, in particular to those considered here. A voting rule that is susceptible to some type of control is said to be *vulnerable to it* if the associated control problem can be solved in polynomial time (i.e., is in P), and it is said to be *resistant to it* if the associated control problem is NP-hard. We mention in passing that since all problems considered in Sections 3 and 4 are in NP, each resistant control problem in fact is NP-complete.

2.4. Unique- versus nonunique-winner model

Our control problems will be defined in Sections 3 and 4 in the *unique-winner model* (see also Table 1). That is, a constructive (destructive) control action is viewed as being successful only if the designated candidate can be made a *unique* winner (*not a unique* winner) by this action. We note, however, that using essentially the same constructions and slightly modifying the arguments in our proofs, most of our results also work in the *nonunique-winner model*, which means that for a constructive (destructive) control action to be successful, it is enough to make the designated candidate only *a* winner (she can be made *not even a* winner) by this action. The only two exceptions are destructive control by partition and by run-off partition of candidates in the ties-promote model (to be defined in Section 3) to which Borda will be shown resistant in the unique-winner model (Theorems 6 and 7), yet vulnerable in the nonunique-winner model (Theorem 8 and Corollary 2).

In our proofs, we will sometimes use the following result due to Hemaspaandra et al. [58], which shows that some of the destructive candidate control cases (to be defined in the next section) can collapse depending on the chosen winner model. The notation for control problems used in Fact 1 will be formally introduced in Section 3 where this fact will be applied in two places; it can now safely be skipped and looked up later. Recall that decision problems are simply languages of strings that encode exactly the yes-instances of the problems; so an equality between two control problems in Fact 1 means that Hemaspaandra et al. [58] have proven these two sets of strings to be the same.

Fact 1 *(Hemaspaandra et al. [58]). In the unique-winner model, it holds that* DCRPC-TE = DCPC-TE. *In the nonunique-winner model, it holds that* DCRPC-TE = DCPC-TE *and* DCRPC-TP = DCPC-TP.

3. Complexity of candidate control in Borda elections

In this section, we solve all open problems for candidate control in Borda elections, starting with constructive control by adding an unlimited number of candidates.

3.1. Borda-CCAUC and Borda-DCAUC

Elkind et al. [34] showed that Borda is resistant to constructive control by adding a *limited* number of candidates (i.e., a bound k on the number of candidates that may be added is part of the problem instance), and Loreggia et al. [73] showed that Borda is vulnerable to the destructive variant of this control type (see Table 1). Originally, however, Bartholdi et al. [7] defined control by adding candidates in an *unlimited* variant where no such bound is given. The definition of the limited variant is due to Faliszewski et al. [44], who also proved that the two variants of the problem can have different complexity classifications: Two special cases of Copeland$^{\alpha}$ elections (recall Footnote 7 in Section 1)—namely, Copeland^{0} and Copeland^{1}, the latter a.k.a. Llull elections [55]—are resistant to the constructive, limited variant (the corresponding problem denoted by CCAC), whereas they are vulnerable to the constructive, unlimited variant, which we define now for Borda. In the problem Borda-Constructive-Control-by-Adding-an-Unlimited-Number-of-Candidates (Borda-CCAUC) we ask, given a set C of candidates, an additional set A of candidates, $C \cap A = \emptyset$, a set V of voters with preferences over $C \cup A$, and a distinguished candidate $p \in C$, whether there is a subset $A' \subseteq A$ such that p is the unique Borda winner of $(C \cup A', V)$.[11]

We will show resistance of Borda to this control type. The proof of Theorem 1 will make use of a reduction from Exact-Cover-by-3-Sets (X3C) that is well known to be NP-complete [51]:

Exact-Cover-by-3-Sets (X3C)	
Input:	Given a set $X = \{x_1, \ldots, x_{3k}\}$ and a family of subsets of X, $\mathscr{S} = \{S_1, \ldots, S_n\}$, each with three elements.
Question:	Does there exist an exact cover of X, i.e., a subfamily $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = k$ such that each element $x_i \in X$ occurs in exactly one subset $S_j \in \mathscr{S}'$?

Our reduction from X3C to Borda-CCAUC will also employ Lemma 1, which was proven by Elkind et al. [34, Lemma B.3] and allows us to construct votes conveniently.[12]

Lemma 1 *(Elkind et al. [34]). Let $C = \{c_1, \ldots, c_{2t-1}, d\}$, $t \geq 2$, be a set of candidates and let $A = \{a_1, \ldots, a_s\}$ be a set of spoiler candidates. Let $L = 2t - 1$ and $M_{|A'|} = L(2|A'| + |C| - 1)$ for every $A' \subseteq A$. Then there is a polynomial-time computable preference profile $\mathscr{R} = (R_1, \ldots, R_{2L})$ over $C \cup A$ such that for each $A' \subseteq A$ the Borda scores in the election $(C \cup A', \mathscr{R})$ are as follows:*

(a) *For each $c_i \in C$, $score(c_i) = M_{|A'|} + 1$;*
(b) *$score(d) = M_{|A'|} - L$; and*
(c) *for each $a_i \in A'$, $score(a_i) \leq M_{|A'|} - 2L$.*

Theorem 1. *Borda is resistant to constructive control by adding an unlimited number of candidates.*

Proof. To show NP-hardness, we provide a reduction from X3C to Borda-CCAUC. Let $(X, \mathscr{S})$ be a given X3C instance with $X = \{x_1, \ldots, x_m\}$, where $m = 3k$ for some $k > 1$, and $\mathscr{S} = \{S_1, \ldots, S_n\}$, where $S_i \subseteq X$ and $|S_i| = 3$ for each i, $1 \leq i \leq n$. Without loss of generality, we assume that k is even and $k > 2$ (this can be achieved by duplicating the instance[13] if necessary). Construct from $(X, \mathscr{S})$ a Borda-CCAUC instance $((C, V), A, p)$ as follows. Let $C = X \cup \{u, p\}$ with p being the distinguished candidate and $A = \{a_1, \ldots, a_n\}$ a set of spoiler candidates. Define V to consist of the following votes:

1. For each i, $1 \leq i \leq n$, there are two votes: $\overrightarrow{S_i}\ u\ p\ \overrightarrow{X \setminus S_i}\ A$ and $\overleftarrow{X \setminus S_i}\ p\ u\ a_i\ \overleftarrow{S_i}\ A \setminus \{a_i\}$.
2. There are three votes of the form $u\ \overrightarrow{A}\ p\ \overrightarrow{X}$ and three votes of the form $\overleftarrow{X}\ p\ u\ \overleftarrow{A}$. (Note that the gap between the score of u and p is proportional to the number of added candidates, which ensures that if the number of candidates crosses a certain limit, then u overtakes p to become the winner.)
3. All votes obtained by applying Lemma 1 to the candidate set C with each x_i taking the role of a c_i, p that of c_{3k+1}, and u that of d. (Here, we need k to be even.)

Note that p ranks ahead of every $a_j \in A'$ in all but three votes in the second group of voters. The point deficit from those three votes is always offset by the other votes in this group, so we can disregard the points of every $a_j \in A$ from now on, since p always defeats them. Regarding the point differences of p to all other candidates, we will later need the following lemma.

Lemma 2. *Let $((C, V), A, p)$ be the constructed Borda-CCAUC instance. For the point differences of p to the other candidates in the election $(C \cup A', V)$ for any $A' \subseteq A$, we have*

(a) *$dist(p, u) = 3k + 2 - 3|A'|$ and*
(b) *$dist(p, x_i) = |\{a_j \in A' \mid x_i \in S_j\}|$.*

Proof of Lemma 2. We show the two assertions as follows.

(a) From the votes in group 1, p and u gain the same number of points. From the votes in group 2, u gains $|A'| + 1$ more points than p in three votes and p gains 1 point more than u in the three other votes. From the last group

[11] For convenience, whenever we have a list V of votes over a set $C \cup A$ of candidates and then consider an election with fewer candidates, $C \cup A'$ with $A' \subset A$, we use $(C \cup A', V)$ to denote the election with the votes in V tacitly assumed to be restricted to $C \cup A'$, meaning that none of the candidates in $A \setminus A'$ appear in the votes of V while the other candidates are ranked in the same order as before.

[12] The original lemma by Elkind et al. [34] is slightly more general in that they consider nonnegative integers $\ell_1, \ldots, \ell_{2t-1}$ with $L = \sum_{i=1}^{2t-1} \ell_i$. For our purpose, it is enough to set $\ell_1 = \cdots = \ell_{2t-1} = 1$, so $L = 2t - 1$.

[13] In order to duplicate an instance $(X, \mathscr{S})$ of X3C, we first clone X so that we obtain $X' = \{x'_1, \ldots, x'_m\}$ with $|X| = |X'|$ and $X \cap X' = \emptyset$, and construct a clone $\mathscr{S}'$ of $\mathscr{S}$ such that $\mathscr{S}' = \{\{x'_i, x'_j, x'_k\} \subseteq X' \mid \{x_i, x_j, x_k\} \in \mathscr{S}\}$. Then we obtain the duplicated instance $(\widehat{X}, \widehat{\mathscr{S}}) = (X \cup X', \mathscr{S} \cup \mathscr{S}')$. It is obvious that $(\widehat{X}, \widehat{\mathscr{S}})$ is a yes-instance of X3C if and only if $(X, \mathscr{S})$ is a yes-instance of X3C and if $|X| = m = 3k$ with k being uneven then $|\widehat{X}| = |X| + |X'| = 3 \cdot (2k) = 3k'$ with k' being even.

of votes and Lemma 1, it follows that p gains $L+1$ with $L=3k+1$ more points than u. Summing up, we have $dist(p,u)=3k+2-3|A'|$.

(b) It is obvious that p and candidates from X gain the same number of points from the votes in the voter groups 2 and 3. Let $x_i \in X$. Notice that for every $j, 1 \le j \le n$, with $x_i \notin S_j$, the two votes corresponding to j in voter group 1 give p and x_i the same number of points and for every $j, 1 \le j \le n$, with $x_i \in S_j$, the two votes corresponding to j in voter group 1 give p one point more than x_i only if $a_j \in A'$, and the same number of points otherwise. Summing up, we have $dist(p,x_i)=|\{a_j \in A' \mid x_i \in S_j\}|$.

This completes the proof of the lemma. ❑ Lemma 2

From Lemma 2 it immediately follows that p is not winning (C,V) (with $A'=\emptyset$) alone.

We claim that $(X,\mathscr{S})$ is a yes-instance of X3C if and only if $((C,V),A,p)$ is a yes-instance of Borda-CCAUC.

From left to right, suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Let $A'=\{a_j \in A \mid S_j \in \mathscr{S}'\}$. Then from Lemma 2 we have $dist_{(C\cup A',V)}(p,x_i)=|\{a_j \in A' \mid x_i \in S_j\}|=1$ for every $x_i \in X$, since every $x_i \in X$ is contained in one element of the exact cover $\mathscr{S}'$ of X exactly once. Furthermore, from Lemma 2 we have $k=|\mathscr{S}'|=|A'|$. Thus $dist_{(C\cup A',V)}(p,u)=3k+2-3k=2$, so p defeats every candidate and is the only Borda winner of $(C\cup A',V)$.

From right to left, suppose that p can be made the only Borda winner by adding the candidates of a subset $A' \subseteq A$. Therefore, p defeats every candidate in $(C\cup A',V)$, so we have $dist_{(C\cup A',V)}(p,u)>0$ and $dist_{(C\cup A',V)}(p,x_i)>0$ for every $x_i \in X$ (recall that p always defeats every $a_j \in A'$). Since from Lemma 2 we have $dist_{(C\cup A',V)}(p,x_i)=|\{a_j \in A' \mid x_i \in S_j\}|>0$ for every $x_i \in X$, the subfamily $\mathscr{S}'=\{S_j \in S \mid a_j \in A'\}$ covers X. Thus we have $|\mathscr{S}'| \ge k$, as there are $3k$ elements in X and every subset of $\mathscr{S}$ contains three elements. Furthermore, from Lemma 2 we have $dist_{(C\cup A',V)}(p,u)=3k+2-3|A'|>0$, so $|\mathscr{S}'|=|A'| \le k$. Overall, we have that $\mathscr{S}'$ covers X and $|\mathscr{S}'|=k$, which means that $\mathscr{S}'$ is an exact cover of X. ❑

For the destructive variant, we show that Borda-DCAUC is in P.

Theorem 2. *Borda is vulnerable to destructive control by adding an unlimited number of candidates.*

Proof. To show the theorem, we will reduce Borda-DCAUC to Borda-DCAC. Then P membership of Borda-DCAUC follows immediately from Borda-DCAC being in P, which was proven by Loreggia et al. [73].

Let $((C,V),A,p)$ be an instance of Borda-DCAUC. We trivially map it to the corresponding Borda-DCAC instance $((C,V),A,p,|A|)$, i.e., we set the limit of how many candidates of A may be added to (C,V) to the highest possible value, $|A|$. Since we thus can potentially add all candidates of A to the election in $((C,V),A,p,|A|)$ it is immediately clear that $((C,V),A,p,|A|)$ is a yes-instance of Borda-DCAC if and only if $((C,V),A,p)$ is a yes-instance of Borda-DCAUC.[14] ❑

3.2. Borda-CCRPC-TE and Borda-CCPC-TE

In the problem Borda-Constructive-Control-by-Run-Off-Partition-of-Candidates-TE (Borda-CCRPC-TE) we ask, given an election (C,V) and a distinguished candidate $p \in C$, whether the candidate set C can be partitioned into two subsets C_1 and C_2 such that p is the unique Borda winner of the final run-off among the Borda winners of subelections (C_1,V) and (C_2,V), where only unique subelection winners move forward in the ties-eliminate (TE) model.

The proof of Theorem 3 below makes use of a reduction from the standard NP-complete satisfiability problem (3-SAT) [51]:

3-Satisfiability (3-SAT)	
Input:	Given a boolean formula φ in 3-CNF (i.e., with exactly three literals per clause).
Question:	Does there exist a satisfying truth assignment to the variables of φ?

For a boolean formula φ, we denote by $\#_i$ the number of literals occurring in the ith clause that are negated variables.

Theorem 3. *Borda is resistant to constructive control by run-off partition of candidates in the ties-eliminate model.*

Proof. To show NP-hardness, we now provide a reduction from 3-SAT to Borda-CCRPC-TE. Given a 3-SAT instance $\varphi(x_1,\dots,x_n)$, construct a Borda-CCRPC-TE instance $((C,V),p)$ as follows. Let $X=\{x_1,x_2,\dots,x_n\}$ be the set of variables and let $K=\{K_1,\dots,K_m\}$ be the set of clauses of φ, where $K_i=(\ell_i^{(1)} \vee \ell_i^{(2)} \vee \ell_i^{(3)})$, $1 \le i \le m$. Furthermore, let $D=\{d_1,\dots,d_6\}$ and $D_i=\{d_1,\dots,d_i\} \subseteq D$. Define the candidate set by $C=X\cup K\cup\{p,r,r^*\}\cup D$ with p being the distinguished candidate the chair wants to make a unique winner. Define V to consist of the following votes:

[14] Recall, however, that in general control by adding a limited number of candidates may behave differently from control by adding an unlimited number of candidates. For example, as mentioned earlier, for Copeland^0 and Copeland^1 elections, CCAC and CCAUC—the constructive analogues of DCAC and DCAUC—are known to differ in terms of their computational complexity (assuming $P \neq NP$).

1. For each i, $1 \le i \le m$, there are two votes:
$$\overrightarrow{C \setminus (\{p, K_i\} \cup D)}\ p\ D_{2\#_i}\ K_i\ D \setminus D_{2\#_i} \text{ and } K_i\ d_6\ p\ \overleftarrow{C \setminus (\{p, K_i\} \cup D)}\ D_5.$$
2. For each i, $1 \le i \le m$, and for each literal $\ell_i^{(1)}$, $\ell_i^{(2)}$, and $\ell_i^{(3)}$, there are four votes: either
   - two votes $K_i\ x_j\ p\ \overrightarrow{C \setminus (\{K_i, x_j, p\} \cup D)}\ D$ and
   - two votes $\overleftarrow{C \setminus (\{K_i, x_j, p\} \cup D)}\ p\ K_i\ x_j\ D$

   if $\ell_i^{(k)} = \overline{x}_j$ is a negated variable, or
   - two votes $\overrightarrow{C \setminus (\{K_i, x_j, p\} \cup D)}\ p\ x_j\ K_i\ D$ and
   - two votes $K_i\ p\ \overleftarrow{C \setminus (\{K_i, x_j, p\} \cup D)}\ x_j\ D$

   if $\ell_i^{(k)} = x_j$ is a positive variable.
3. There are m votes of the form $r^*\ r\ \overrightarrow{K}\ \overrightarrow{D}\ p\ X$ and m votes of the form $r\ p\ \overleftarrow{D}\ \overleftarrow{K}\ r^*\ X$.

Since $dist_{(C,V)}(p, r) = m(-6 - m - 2) = -m(m + 8) < 0$, p does not win in (C, V). Note that p and r score the same number of points in the first two groups of votes. Later on, we will also need the following lemma.

Lemma 3. *Let $((C, V), p)$ be the constructed Borda-CCRPC-TE instance. In the election $(\{p\} \cup D \cup K \cup X', V)$ with $X' \subseteq X$, p beats all candidates from K if for every $K_i \in K$, X' contains at least one variable candidate corresponding to a positive variable in clause K_i, or $X \setminus X'$ contains at least one variable candidate corresponding to a negated variable in clause K_i, or both.*

Proof of Lemma 3. Consider a clause candidate K_i. In the first group of votes, p scores $2\#_i - 1$ points more than candidate K_i (recall that $\#_i$ is the number of negated variables in clause K_i). In the second group of votes, p gains two more points with respect to candidate K_i for each positive variable in clause K_i, and p loses two points with respect to candidate K_i for each negated variable in clause K_i. Since p and K_i score the same number of points in the third group of votes, we have

$$dist_{(C,V)}(p, K_i) = -2\#_i + 2(3 - \#_i) + (2\#_i - 1) = 5 - 2\#_i.$$

Assuming that one variable candidate $x_j \notin X'$, if x_j is a negated variable in clause K_i then p gains two points with respect to candidate K_i, and if x_j is a positive variable in clause K_i then p loses two points with respect to K_i. Further, if $X' \subseteq X$ is the set of candidates obtained by removing from X all variable candidates corresponding to positive variables in clause K_i, then

$$dist_{(\{p\} \cup D \cup K \cup X', V)}(p, K_i) = 5 - 2\#_i - 2(3 - \#_i) = -1$$

because p is losing as many points with respect to K_i as there are positive variables in clause K_i. That is, p is defeated by K_i if (1) all variable candidates corresponding to positive variables in clause K_i are not in X' and (2) all variable candidates corresponding to negated variables in clause K_i are in X'. Note that, for p to defeat K_i, either (1) or (2), or both, must be false. ❑ Lemma 3

The following example illustrates Lemma 3.

Example 2. Let $K_1 = (x_1 \vee x_2 \vee \overline{x}_3) \in K$ be a clause over the variables $\{x_1, x_2, x_3\}$. Since there is only one negated variable in this clause, we have $\#_1 = 1$. Therefore, p scores one point more than K_1 from the first voter group:

$$\overrightarrow{C \setminus (\{p, K_1\} \cup D)}\ p\ D_2\ K_1\ D \setminus D_2 \text{ and } K_1\ d_6\ p\ \overleftarrow{C \setminus (\{p, K_1\} \cup D)}\ D_5.$$

Then the following 12 votes are in the second voter group:

- two votes $\overrightarrow{C \setminus (\{K_1, x_1, p\} \cup D)}\ p\ x_1\ K_1\ D$,
- two votes $K_1\ p\ \overleftarrow{C \setminus (\{K_1, x_1, p\} \cup D)}\ x_1\ D$,
- two votes $\overrightarrow{C \setminus (\{K_1, x_2, p\} \cup D)}\ p\ x_2\ K_1\ D$,
- two votes $K_1\ p\ \overleftarrow{C \setminus (\{K_1, x_2, p\} \cup D)}\ x_2\ D$,
- two votes $K_1\ x_3\ p\ \overrightarrow{C \setminus (\{K_1, x_3, p\} \cup D)}\ D$, and
- two votes $\overleftarrow{C \setminus (\{K_1, x_3, p\} \cup D)}\ p\ K_1\ x_3\ D$.

From the first four votes in this group, p gains two points on K_1 only if x_1 is participating in the election; from the second four votes, p gains two points on K_1 only if x_2 is participating in the election; and from the last four votes p loses two points on K_1 only if x_3 is participating. Therefore, p is beaten by K_1 if and only if x_1 and x_2 are not participating and x_3 is participating in the election, which corresponds to setting the variables x_1 and x_2 to false and x_3 to true such that the clause K_1 is not satisfied.

We show that φ is a yes-instance of 3-SAT if and only if $((C, V), p)$ is a yes-instance of Borda-CCRPC-TE.

From left to right, suppose there is a satisfying truth assignment to the variables of $\varphi(x_1, \dots, x_n)$, say α. Let $X_+ \subseteq X$ denote the set of variables set to *true* under α, and let $X_- \subseteq X$ denote the set of variables set to *false* under α. Partition C into $C_1 = \{p\} \cup D \cup K \cup X_+$ and $C_2 = \{r, r^*\} \cup X_-$. The Borda winners of subelection (C_2, V) are r and r^*, since they score more points than the candidates in X_- due to the third voter group and the same number of points in the other two voter groups. Due to TE, no candidate proceeds to the final run-off from this subelection. In subelection (C_1, V), p defeats all candidates from D, since p scores more points than these candidates in the first voter group and the same number of points in the other two voter groups. p also defeats all candidates from X_+, since p scores at least $m(m + 5)$ points more than any candidate in X_+ in the third voter group, at most m points fewer than any candidate from X_+ in the second voter group (which is the case if some positive variable occurs in all clauses), and the same number of points in the first voter group.

What about the clause candidates? The truth assignment (giving rise to X_+ and X_-) satisfies φ, so each clause K_i of φ is satisfied. Thus, for every i, $1 \le i \le m$, at least one positive variable in K_i is assigned to *true* or at least one negated variable in K_i is assigned to *false*. In the former case, the corresponding variable candidate is in X_+ and thus in the same subelection as p; in the latter case, the corresponding variable candidate is in X_- and thus not in the same subelection as p. By Lemma 3, p scores more points than K_i. Summing up, since p defeats all other candidates in her subelection and no one moves forward to the final run-off from the other subelection, p alone is the overall Borda winner.

From right to left, suppose that p is the unique overall Borda winner for some partition of the candidates. This implies that p also is the unique Borda winner of one subelection. Since r scores more points than p due to the third voter group, p and r must be in different subelections (regardless of who else participates in them). Without loss of generality, assume that p is in C_1 and r is in C_2.

Consider C_2 first. r cannot be the unique Borda winner in subelection (C_2, V), since otherwise p would not win the run-off against r. Therefore, there must be candidates that either tie or defeat r in (C_2, V). Clause candidates, variable candidates, and candidates from D lose too many points in the third voter group (that cannot be made up for in the first and second voter groups) to tie-or-defeat r. Only candidate r^* remains. However, r^* cannot be the unique Borda winner of subelection (C_2, V), since p and r^* would score the same number of points in the run-off, contradicting that p is the *unique* run-off winner. Thus there must be a tie between r and r^* in (C_2, V), which prevents them both from proceeding to the run-off due to the TE model. Therefore, neither candidates from D nor from K can be in C_2, for otherwise the balance of points between r and r^* would be perturbed due to the third voter group. Variable candidates, however, may be in C_2, since they get fewer points than either r and r^* and would not interfere with their point balance. Thus C_1 contains p and all candidates from D and K and some variable candidates. Let X_+ denote the set of variable candidates in C_1. Note that p defeats the candidates in D by the first voter group and the candidates in X_+ by the third voter group. Since p also defeats each clause candidate K_i, the variable candidates must be distributed among C_1 and C_2 according to Lemma 3. Now, if we assign the value *true* to all variables corresponding to variable candidates in X_+ and the value *false* to all variables corresponding to variable candidates not in X_+, we obtain a truth assignment satisfying $\varphi(x_1, \dots, x_n)$. ❑

Borda-Constructive-Control-by-Partition-of-Candidates-TE (Borda-CCPC-TE) is defined as follows. Given an election (C, V) and a distinguished candidate $p \in C$, we ask whether the candidate set C can be partitioned into two subsets C_1 and C_2 such that p is the unique Borda winner of the final election in which the Borda winner of subelection (C_1, V)—if there exists one (again, in model TE, only unique subelection winners move forward)—faces all candidates from C_2.

Theorem 4. *Borda is resistant to constructive control by partition of candidates in the ties-eliminate model.*

Proof. To show NP-hardness, since instances of Borda-CCRPC-TE and Borda-CCPC-TE are defined identically, we can use the same construction as in the proof of Theorem 3 that yields a reduction from 3-SAT to Borda-CCRPC-TE. That is, given a 3-SAT instance $\varphi(x_1, \dots, x_n)$, construct the same candidates and votes as in the proof of Theorem 3. Note that the argument on the point balance of p and a clause candidate K_i (given right after the construction in that proof) still holds.

To show correctness of the construction, we only outline the most important arguments to highlight the slight differences to the argumentation in that previous proof. To prove the equivalence from left to right, suppose there is a satisfying truth assignment α to the variables of $\varphi(x_1, \dots, x_n)$. Partition C into C_1 and C_2 so that C_1 contains r, r^*, and all variable candidates that are set to *false* in α, and C_2 contains all the other candidates. Note that r and r^* tie in subelection (C_1, V) and are thus eliminated by the tie-handling rule. Candidates in C_2 get a bye to the final run-off in which p then beats all other candidates (in particular, the clause candidates) from C_2 because α is a satisfying truth assignment.

For the right-to-left direction, suppose that p is the unique overall Borda winner for some partition of the candidates. Candidate r had to be eliminated in the subelection; otherwise, r would have beaten p in the run-off. This can only be achieved by r^*, who can tie (but not beat) r in the subelection if the candidates in D, K, and p are not participating.

Thus C_1 contains r, r^*, and some variable candidates, and C_2 contains p, all candidates from D and K, and the remaining variable candidates. All candidates from C_2 advance directly to the run-off, and in subelection (C_1, V) all winners are tieing and, therefore, are eliminated by the tie-handling rule. Since p beats all clause candidates in the run-off, the variable candidates must have been distributed among C_1 and C_2 according to the above argument. Assigning every variable candidate in C_2 to *true* and all the others to *false*, we obtain a satisfying truth assignment. ❑

3.3. Borda-DCPC-TE and Borda-DCRPC-TE

We now turn to the destructive variants of the previous two problems. Unlike in the constructive case, we can give a polynomial-time algorithm for Borda-DCPC-TE.

Theorem 5. *Borda is vulnerable to destructive control by partition of candidates in the ties-eliminate model.*

Proof. Our algorithm uses the result of Loreggia et al. [73] that Borda-DCDC is in P (see Table 1). Given an election (C, V) and a distinguished candidate $p \in C$, the algorithm works as follows:

1. If p is not a unique Borda winner, accept immediately because control is possible via the trivial partition $(C, \emptyset)$.
2. Otherwise, if $|C| = 1$, control is impossible since p is the only candidate and always wins, so reject.
3. Now, let $k = |V| - 2$. If $((C, V), p, k)$ is a yes-instance of Borda-DCDC, which can be checked in polynomial time [73], accept; otherwise, reject.

This algorithm runs in polynomial time and is correct. In step 1, if p is not a unique Borda winner, p is at most tied with some candidates, so she can be either beaten or eliminated by the tie-handling rule in a subelection. Correctness of step 2 is obvious. For step 3, if the constructed instance is a yes-instance, there is a subset $C' \subseteq C$, $|C'| \le k = |V| - 2$, $p \notin C'$, so that p is at most tied in the election $(C \setminus C', V)$. Therefore, we can eliminate p in the subelection by partitioning C into $C \setminus C'$ and C', so control is possible. If the instance is a no-instance, p cannot be beaten or tied even if all but one candidate other than p are deleted from the election. That means that p is the sole winner of $(C \setminus C', V)$ for every $C' \subseteq C$ with $|C'| \le k = |V| - 2$ and $p \notin C'$; so p cannot be eliminated in a subelection. In this case, p wins any subelection and reaches the run-off. There may be a set of other candidates that reached the final round, say $C^\star \subseteq C$, $p \notin C^\star$. If some of those candidates beats or at least ties with p in this run-off, destructive control would still have been achieved. But this cannot happen because then p could have been eliminated in a subelection by partitioning C into $\{p\} \cup C^\star$ and $C \setminus (\{p\} \cup C^\star)$. As stated above, however, this is impossible since the constructed Borda-DCDC instance is a no-instance, so p alone wins the run-off and control is impossible. ❑

By Fact 1, Borda-DCPC-TE is the same as Borda-DCRPC-TE in the unique-winner model, which gives the following corollary.

Corollary 1. *Borda is vulnerable to destructive control by run-off partition of candidates in the ties-eliminate model.*

3.4. Borda-DCPC-TP and Borda-DCRPC-TP

Next, we consider the same two problems as above but with the ties-promote (TP) instead of the ties-eliminate rule, which means that *all* subelection winners move forward to the final round. While we focus on the unique-winner model as our default in all other cases (simply noting that minor changes of our proofs yield the same result also in the nonunique-winner model), we here explicitly distinguish between the unique- and the nonunique-winner model because destructive control by partition and by run-off partition of candidates with TP are the only cases considered here where the control complexity in Borda elections varies with the winner model chosen: We show that, in the unique-winner model, Borda is resistant to control by partition of candidates with TP (Theorem 6) and to control by run-off partition of candidates with TP (Theorem 7), yet is vulnerable in the nonunique-winner model (Theorem 8 and Corollary 2).

These are two of the very rare cases where the complexity of a voting problem differs depending on the winner model. Only very few other such examples come to mind. Firstly, while Conitzer et al. [24] have shown that for Copeland elections with three candidates, constructive coalitional weighted manipulation in the unique-winner model is solvable in polynomial time, Faliszewski et al. [47] established NP-completeness of the same problem in the nonunique-winner model. Secondly, Hemaspaandra et al. [60] have shown that online weighted manipulation in the nonunique-winner model for plurality can be solved in polynomial time, yet in the unique-winner model its constructive variant is coNP-hard and its destructive variant is NP-hard.

3.4.1. Unique-winner model

We start with the NP-hardness result for Borda-DCPC-TP in the unique-winner model.

Let us introduce some useful notation that will be employed in the proofs of Theorems 6, 15, and 16. In these proofs, votes are always created in pairs. For a set of candidates C and two candidates $c_1, c_2 \in C$, we denote by $W(c_1, c_2)$ the two votes $c_1\ c_2\ \overrightarrow{C \setminus \{c_1, c_2\}}$ and $\overleftarrow{C \setminus \{c_1, c_2\}}\ c_1\ c_2$. For such a pair, under the Borda rule c_1 gains two points on c_2 and one point on each of the remaining candidates, whereas c_2 loses two points on c_2 and one point on each of the remaining candidates. All other candidates gain the same number of points. Note that if one of c_1 and c_2 is not participating in the election, all candidates gain the same number of points from the two votes of $W(c_1, c_2)$. Furthermore, for the point balance of any two candidates x and y, only those pairs $W(c_1, c_2)$ matter where $\{x, y\} \cap \{c_1, c_2\} \neq \emptyset$.

Theorem 6. *Borda is resistant to destructive control by partition of candidates in the ties-promote model.*

Proof. To show NP-hardness, we provide a reduction from X3C to Borda-DCPC-TP. Let $(X, \mathscr{S})$ be a given X3C instance with $X = \{x_1, \ldots, x_m\}$, where $m = 3k$ for some $k > 5$, and $\mathscr{S} = \{S_1, \ldots, S_n\}$, where $S_i \subseteq X$ and $|S_i| = 3$ for each i, $1 \le i \le n$. We assume that every $x_i \in X$ appears in exactly three subsets $S_j \in \mathscr{S}$ (thus $n = 3k$ as well). This restricted version of X3C was proven to be NP-complete by Gonzalez [54]. The assumption that $k > 5$ can be achieved by cloning and merging instances with smaller k.

Construct from $(X, \mathscr{S})$ a Borda-DCPC-TP instance $((C, V), p)$ as follows. Let $C = \{p, c, r\} \cup B \cup G$ with $B = \{b_1, \ldots, b_{3k}\}$ and $G = \{g_1, \ldots, g_{3k}\}$.

We now construct the following pairs of votes:

number	vote	for
5	$W(p, b_i)$	$1 \le i \le 3k$
$3k$	$W(p, c)$	
7	$W(c, b_i)$	$1 \le i \le 3k$
1	$W(r, c)$	
$3k$	$W(p, r)$	
5	$W(r, g_i)$	$1 \le i \le 3k$
21	$W(g_i, c)$	$1 \le i \le 3k$
3	$W(p, g_i)$	$1 \le i \le 3k$

Additionally, for every $S_j = \{x_s, x_t, x_u\} \in \mathscr{S}$ we construct the following 36 votes: six pairs $W(b_s, g_j)$, six pairs $W(b_t, g_j)$, and six pairs $W(b_u, g_j)$.

We now introduce additional notation to conveniently count point balances of elections with a subset of candidates $C' \subseteq C$.

- Let $\rho_{C'} = 1$ if $r \in C'$ and $\rho_{C'} = 0$ if $r \notin C'$.
- Let $\zeta_{C'} = 1$ if $c \in C'$ and $\zeta_{C'} = 0$ if $c \notin C'$.
- For every $b_i \in B$ and $C' \subseteq C$ with $G' = C' \cap G$, let $\ell_{b_i}^{C'}$ be the number of times x_i is covered by the set $\mathscr{S}' \subseteq \mathscr{S}$ that corresponds to G' (i.e., for every $g_i \in G'$, there is $S_i \in \mathscr{S}'$). Note that $1 \le \ell_{b_i}^{C'} \le 3$ and $|G'| = |\mathscr{S}'| \ge \ell_{b_i}^{C'}$.
- For every $g_i \in G$ and $C' \subseteq C$ with $B' = C' \cap B$ and $X' = \{x_i \in X \mid b_i \in B'\}$, let $\ell_{g_i}^{C'} = |S_i \cap X'|$.

From the constructed votes we have the following point balances for an election (C', V) with $C' \subseteq C$, $B' = C' \cap B$ and $G' = C' \cap G$:

$$
\begin{aligned}
dist_{(C',V)}(p, c) &= 6k - 2|B'| + (3k+1)\rho_{C'} + 24|G'|, \\
dist_{(C',V)}(p, r) &= 6k - 2|G'| + 5|B'| + (3k-1)\zeta_{C'}, \\
dist_{(C',V)}(p, b_i) &= 5(|B'|+1) + (3k+7)\zeta_{C'} + 3k\rho_{C'} + 3|G'| - 6\ell_{b_i}^{C'} \text{ for every } b_i \in B', \text{ and} \\
dist_{(C',V)}(p, g_i) &= 3(|G'|+1) + (3k-21)\zeta_{C'} + (3k+5)\rho_{C'} + 6\ell_{g_i}^{C'} \text{ for every } g_i \in G'.
\end{aligned}
$$

We can see that p can only be tied with r in $C' = \{p, r\} \cup G$ or with c in $C' = \{p, c\} \cup B$, and beats both of them otherwise. Also, p always beats every b_i since

$$dist_{(C',V)}(p, b_i) \ge 5(|B'|+1) + 3|G'| - 6\ell_{b_i}^{C'} \ge 10 - 3\ell_{b_i}^{C'} > 0.$$

Furthermore, p always beats every g_i since

$$dist_{(C',V)}(p, g_i) \ge 3(|G'|+1) + (3k-21)\zeta_{C'} \ge 6 + 3k - 21 = 3k - 15 > 0.$$

Here we need the requirement $k > 5$.

Before we proceed with the proof of correctness, we provide some intuition on how the reduction is set up. The votes are chosen in such a way that p can only be prevented from winning alone if p is tied with other candidates in the final-round election. Otherwise, p could simply be eliminated in the subelections. There are only two subsets of possible "tie" candidates (referring to cases (a) and (b) later on) for which p can be tied with in the final-round election, so all other candidates need to be eliminated in the subelections (since we are in the TP model, those candidates need to be actually beaten in order to be eliminated). The distinguished candidate p cannot be used to conveniently eliminate those other candidates, as the remaining candidates from case (a) that are not used to tie with p in the final-round election are the "tie" candidates from case (b), and vice versa. So, depending on which case we try to carry out, we need some of the "tie" candidates to beat all other candidates in the subelection while also tieing among themselves in order to reach the final-round election. Although the two cases seem to be symmetrical, we can show that only case (a) can lead to success. Then the candidates from B

correspond to elements from X and candidates from G correspond to subsets of X in $\mathscr{S}$. Using some subset of X in $\mathscr{S}$ to cover elements of X corresponds to using the corresponding candidate of G to eliminate candidates in the subelection.

Now proceeding with the proof of correctness, we claim that $(X, \mathscr{S})$ is a yes-instance of X3C if and only if $((C, V), p)$ is a yes-instance of Borda-DCPC-TP.

From right to left, suppose $((C, V), p)$ is a yes-instance of Borda-DCPC-TP, via a successful partition of C into C_1 and C_2 that prevents p from being the only winner of the final-round election in which the Borda winners of (C_1, V) and all candidates of C_2 participate. From the point balances and discussion above we see that p cannot be beaten in any subelection. So p must be tied in the final election. This can only happen if either (a) $\{r\} \cup G$ or (b) $\{c\} \cup B$ reach the final round while either $\{c\} \cup B$ (in case (a)) or $\{r\} \cup G$ (in case (b)) are eliminated in the first round.

First, assume that p is participating in the subelection (C_1, V). In case (a), $\{c\} \cup B$ need to be eliminated in (C_1, V). If no other candidate is participating in this subelection (i.e., if $C_1 = \{p, c\} \cup B$), c together with p would proceed to the final run-off. However, neither r nor any $g_i \in G$ can participate in (C_1, V) since they would then be eliminated in the first round but are needed in the final round to prevent p from being the only winner. Thus we cannot eliminate $\{c\} \cup B$ without eliminating candidates from $\{r\} \cup G$. In case (b), $\{r\} \cup G$ need to be eliminated in (C_1, V) (i.e., $\{p, r\} \cup G \subseteq C_1$). With a similar argument as in case (a) we can show that we cannot eliminate all candidates $\{r\} \cup G$ without eliminating candidates from $\{c\} \cup B$. All in all, if p participates in (C_1, V) then p cannot be tied in the final round and wins the election alone. Therefore, we conclude that $p \in C_2$.

Again, there are the same two cases as above, (a) and (b), that could lead to a tie in the final run-off. First, we have the following point balances for elections (C', V) with $C' \subseteq (C \setminus \{p\})$, $B' = C' \cap B$, and $G' = C' \cap G$:

$$\begin{aligned}
dist_{(C',V)}(r, c) &= -7|B'| + 26|G'| + 2, \\
dist_{(C',V)}(r, b_i) &= 8\zeta_{C'} + 5|G'| - 6\ell_{b_i}^{C'} \text{ for every } b_i \in B', \\
dist_{(C',V)}(r, g_i) &= 5(|G'| + 1) - 20\zeta_{C'} + 6\ell_{g_i}^{C'} \text{ for every } g_i \in G', \\
dist_{(C',V)}(c, b_i) &= 7(|B'| + 1) + \rho_{C'} - 6\ell_{b_i}^{C'} \text{ for every } b_i \in B', \\
dist_{(C',V)}(c, g_i) &= 7|B'| + 4\rho_{C'} - 21(|G'| + 1) + 6\ell_{g_i}^{C'} \text{ for every } g_i \in G', \text{ and} \\
dist_{(C',V)}(g_i, b_j) &= 28\zeta_{C'} - 5\rho_{C'} - 6(\ell_{g_i}^{C'} + \ell_{b_j}^{C'}) \text{ for every } g_i \in G' \text{ and for every } b_j \in B'.
\end{aligned}$$

In case (a), $\{c\} \cup B$ need to be eliminated in (C_1, V). We need to add other candidates from $\{r\} \cup G$ to the subelection, so all candidates from $\{c\} \cup B$ are eliminated and all added candidates proceed to the final round. There are three possible ways of adding such candidates: (1) only add r, (2) add r and some candidates $G' \subseteq G$, and (3) add only some $G' \subseteq G$. In case (1), c would beat r since

$$dist_{(C',V)}(r, c) = -21k + 2 < 0.$$

In case (2), we have for every $g_i \in G'$ that

$$dist_{(C',V)}(r, g_i) = 5(|G'| + 1) - 20 + 18 = 5|G'| + 3 > 0,$$

so every $g_i \in G'$ would also be eliminated. This leaves only case (3). For every $g_i \in G'$, we have that

$$dist_{(C',V)}(c, g_i) = 21k - 21(|G'| + 1) + 18 = 21k - 21|G'| - 3.$$

Therefore, we need $|G'| \geq k$ in order for c to be beaten. Furthermore, we have for every $g_i \in G'$ and every $b_j \in B$ that

$$dist_{(C',V)}(g_i, b_j) = 28 - 6(3 + \ell_{b_j}^{C'}) = 10 - 6\ell_{b_j}^{C'}.$$

Therefore, to beat all b_j we need $\ell_{b_j}^{C'} \leq 1$ to hold for every $b_j \in B$. This, however, means that $\mathscr{S}' = \{S_i \in \mathscr{S} \mid g_i \in G'\}$ covers every $x_i \in X$ at most once. Since we need $|\mathscr{S}'| = |G'| \geq k$, this is possible only if $\mathscr{S}'$ is an exact cover of X.

In case (b), $\{r\} \cup G$ need to be eliminated in the subelection (C_1, V) (i.e., $\{r\} \cup G \subseteq C_1$). Therefore, we again need to add some candidates from $\{c\} \cup B$ to C_1 but they cannot be eliminated themselves in (C_1, V). From the point balances above we have that r always beats c in this case (since $dist_{(C',V)}(r, c) = -7|B'| + 78k + 2 \geq -21k + 78k + 2 > 0$) and always beats any $b_i \in B$ (since $dist_{(C',V)}(r, b_i) = 15k - 6\ell_{b_i}^{C'} \geq 15k - 18 > 0$). Therefore, this case cannot lead to a tie of c with p in the final round.

From left to right, suppose there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. We now show that $((C, V), p)$ is a yes-instance of Borda-DCPC-TP. Let $G' = \{g_i \in G \mid S_i \in \mathscr{S}'\}$. Then we partition C into $C_1 = \{c\} \cup B \cup G'$ and $C_2 = \{p, r\} \cup (G \setminus G')$. In the subelection (C_1, V), we have the following point differences:

$$\begin{aligned}
dist_{(C_1,V)}(c, g_i) &= 21k - 21(|G'| + 1) + 6\ell_{g_i}^{C_1} \text{ for every } g_i \in G' \text{ and} \\
dist_{(C_1,V)}(g_i, b_j) &= 28 - 6(\ell_{g_i}^{C_1} + \ell_{b_j}^{C_1}) \text{ for every } g_i \in G' \text{ and for every } b_j \in B.
\end{aligned}$$

Note that $|G'| = |\mathscr{S}'| = k$ and $\ell_{b_j}^{C_1} = 1$ since $\mathscr{S}'$ is an exact cover of X, and $\ell_{g_i}^{C_1} = 3$ since all candidates from B are participating in the election. Therefore, it holds that $dist_{(C_1,V)}(c, g_i) < 0$ and $dist_{(C_1,V)}(g_i, b_j) > 0$ for every $g_i \in G'$ and $b_j \in B$. Furthermore, we have $dist_{(C_1,V)}(g_i, g_j) = 0$ for every $g_i, g_j \in G'$. Therefore, all candidates from G' win (C_1, V) and proceed to the final round. This leaves p, r, and all candidates from G in the final election. From the arguments and discussion above it follows that p is tied with r in this election. Therefore, p is prevented from winning alone. ❑

Next, we consider the analogous control type with two first-round subelections and show that Borda-DCRPC-TP is NP-hard as well.

Theorem 7. *Borda is resistant to destructive control by run-off partition of candidates in the ties-promote model.*

Proof. To show NP-hardness, we provide a reduction from X3C to Borda-DCRPC-TP. Let $(X, \mathscr{S})$ be a given X3C instance with $X = \{x_1, \dots, x_m\}$, where $m = 3k$ for some $k > 1$, and $\mathscr{S} = \{S_1, \dots, S_n\}$, where $S_i \subseteq X$ and $|S_i| = 3$ for each i, $1 \le i \le n$. Note that we again assume that every $x_i \in X$ appears in exactly three subsets $S_j \in \mathscr{S}$; hence, $n = 3k$ as well. Construct from $(X, \mathscr{S})$ a Borda-DCRPC-TP instance $((C, V), p)$ as follows. Let $C = \{p, r\} \cup X \cup \mathscr{S}$ with p being the distinguished candidate. For every $x_i \in X$, let $\mathscr{S}_{x_i} = \{S_j \mid x_i \in S_j\} \subseteq \mathscr{S}$. Note that $|\mathscr{S}_{x_i}| = 3$ for every $x_i \in X$. Define V to consist of the following votes:

1. There are $3k + 1$ votes of the form $p \ \overrightarrow{\mathscr{S}} \ r \ \overrightarrow{X}$ and $3k + 1$ votes of the form $p \ \overleftarrow{\mathscr{S}} \ r \ \overleftarrow{X}$. Intuitively, the voters in this group ensure that candidates from $\mathscr{S}$ appearing in an election will make p beat r.
2. There is a vote $r \ \overrightarrow{X} \ p \ \overrightarrow{\mathscr{S}}$ and a vote $r \ \overleftarrow{X} \ p \ \overleftarrow{\mathscr{S}}$. Intuitively, the voters in this group ensure that candidates from X appearing in an election will give r enough points to make up for the points gap from voters in the first voter group if none of $\mathscr{S}$ are part of the election.
3. For every $x_i \in X$, there are $(3k + 1)(3k + 2)$ votes $x_i \ \mathscr{S}_{x_i} \ \overrightarrow{X \setminus \{x_i\}} \ r \ p \ \overrightarrow{\mathscr{S} \setminus \mathscr{S}_{x_i}}$ and there are $(3k + 1)(3k + 2)$ votes $p \ \mathscr{S}_{x_i} \ \overleftarrow{\mathscr{S} \setminus \mathscr{S}_{x_i}} \ r \ \overleftarrow{X \setminus \{x_i\}} \ x_i$. Intuitively, the voters in this group ensure that we need an exact cover in order to have the correct candidates in the final election. In particular, we need this group to be so large that covering some candidate from X more than once would prevent r from reaching the final election.
4. There are $3k(3k+1)(3k+2)$ votes $p \ \overrightarrow{X} \ r \ \overrightarrow{\mathscr{S}}$ and $3k(3k+1)(3k+2)$ votes $r \ \overleftarrow{X} \ p \ \overleftarrow{\mathscr{S}}$. Intuitively, the voters in this group ensure that candidates from $\mathscr{S}$ never have a chance to win, so we can disregard their scores.

Before we proceed to prove that the reduction is correct, we need the following two lemmas.

Lemma 4. *For subsets $X' \subseteq X$ and $\mathscr{S}' \subseteq \mathscr{S}$, p is the unique Borda winner of the election (C', V) with $C' = \{p, r\} \cup X' \cup \mathscr{S}'$ if $|X'| < 3k$ or $|\mathscr{S}'| > 0$, and a Borda winner only tied with r otherwise.*

Proof of Lemma 4. For subsets $X' \subseteq X$ and $\mathscr{S}' \subseteq \mathscr{S}$, we have the following point balances in the election (C', V) with $C' = \{p, r\} \cup X' \cup \mathscr{S}'$:

- $dist_{(C',V)}(p, r) = (2|\mathscr{S}'| + 2)(3k + 1) - 2(|X'| + 1) + |X'| \cdot |\mathscr{S}'|(3k + 1)(3k + 2)$.
- For each $x_i \in X'$, $dist_{(C',V)}(p, x_i) \ge (3k + 1)(2|\mathscr{S}'| + 2) + 3k(|X'| + 1) > 0$.
- For each $S_j \in \mathscr{S}'$, $dist_{(C',V)}(p, S_j) \ge (3k + 1)(3k + 2)((3k - 3)|X'| + 6k + 3k|\mathscr{S}'| - 9) > 0$.

We can see that p always beats all $x_i \in X'$ and $S_j \in \mathscr{S}'$ and ties r only if $X' = X$ and $\mathscr{S}' = \emptyset$. Note that even when r is removed from the election, p is the only Borda winner for any $X' \subseteq X$ and $\mathscr{S}' \subseteq \mathscr{S}$. ❑ Lemma 4

From Lemma 4 we see that p is the only Borda winner of election (C, V).

Lemma 5. *For subsets $X' \subseteq X$ and $\mathscr{S}' \subseteq \mathscr{S}$, p is the unique Borda winner of the election (C', V) with $C' = \{p\} \cup X' \cup \mathscr{S}'$.*

Proof of Lemma 5. For subsets $X' \subseteq X$ and $\mathscr{S}' \subseteq \mathscr{S}$, we have the following point balances in the election (C', V) with $C' = \{p, r\} \cup X' \cup \mathscr{S}'$:

- For each $x_i \in X'$, $dist_{(C',V)}(p, x_i) \ge (3k + 1)(2|\mathscr{S}'|) + 3k(|X'| + 1) > 0$.
- For each $S_j \in \mathscr{S}'$, $dist_{(C',V)}(p, S_j) \ge (3k + 1)(3k + 2)((3k - 3)|X'| + 3k + 3k|\mathscr{S}'| - 6) > 0$.

We can see that p always beats all $x_i \in X'$ and $S_j \in \mathscr{S}'$. ❑ Lemma 5

Intuitively, from Lemma 4 and Lemma 5 it follows that p can only be prevented from winning alone if r, all of X, and none of $\mathscr{S}$ reach the final-round election. Furthermore, p needs to be in a different subelection than r and all of X, for otherwise not all of them would reach the final-round election. Then we need to add specific candidates from $\mathscr{S}$ (i.e., an exact cover of X) to the subelection with r and all of X in order to have them tied with each other.

Proceeding with the proof of Theorem 7, we claim that $(X, \mathscr{S})$ is a yes-instance of X3C if and only if $((C, V), p)$ is a yes-instance of Borda-DCRPC-TP.

From left to right, suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. We partition C into $C_1 = \{p\} \cup (\mathscr{S} \setminus \mathscr{S}')$ and $C_2 = \{r\} \cup X \cup \mathscr{S}'$. By Lemma 5, p is the only Borda winner of (C_1, V) and reaches the final round. In the second subelection, r ties every $x_i \in X$ since

$$dist_{(C_2,V)}(r, x_i) = (3k+1)^2 + (3k+1) - (3k+1)(3k+2) = 0.$$

Furthermore, r beats every $S_j \in \mathscr{S}'$ since

$$dist_{(C_2,V)}(r, S_j) > -(3k+1)(3k+2)(4k) + 3k(3k+1)(3k+2)(5k) > 0.$$

Due to the ties-promote model, r and every $x_i \in X$ reach the final round. Since p, r, and all candidates in X participate in the final round, but no candidates in $\mathscr{S}$ do, from Lemma 4 we can conclude that p is tied with r and thus prevented from being a unique Borda winner.

From right to left, suppose that p can be prevented from being the unique Borda winner by partitioning the set of candidates. From Lemma 4 and Lemma 5 we can conclude that r, all candidates in X and no candidate in $\mathscr{S}$ reach the final round. Furthermore, p cannot participate in a subelection with r or some candidates X as this would prevent at least one of them from reaching the final round. Without loss of generality, assume that $p \in C_1$ and $\{r\} \cup X \subseteq C_2$. It is easy to see that p is the only Borda winner of (C_1, V) and so reaches the final round. In $(\{r\} \cup X, V)$, r beats every $x_i \in X$ by $(3k+1)(3k+2)$ points. For every $S_j \in \mathscr{S}$ that is added to C_2, every $x_i \in S_j$ gains $(3k+1)(3k+2)$ points on r. For r and all candidates from X to proceed to the final round, candidates $\mathscr{S}' \subseteq \mathscr{S}$ need to be added to C_2 so that every $x_i \in X$ is contained in exactly one element of $\mathscr{S}'$. Therefore, $\mathscr{S}'$ is an exact cover of X. Note also that r beats all those candidates $\mathscr{S}'$ in (C_2, V). ❑

3.4.2. Nonunique-winner model

While we focus on the unique-winner model by default, in this section we make an exception and assume the *nonunique-winner model*. In stark contrast with Theorem 6 (which shows that Borda-DCPC-TP in the unique-winner model is NP-hard), we now show that Borda-DCPC-TP in the *non*unique-winner model is in P.

Theorem 8. *In the nonunique-winner model, Borda is vulnerable to destructive control by partition of candidates in the ties-promote model.*

Proof. To prove P membership of the problem, the algorithm from the proof of Theorem 5 can be used with some slight modifications. Let (C, V) be a given election and $p \in C$ the distinguished candidate. Apart from the trivial cases, we only need to check whether there is a candidate who beats p in an election with a subset of candidates $C' \subseteq C \setminus \{p\}$, which can be done in polynomial time by slightly modifying an algorithm of Loreggia et al. [73]. If this is the case, we can prevent p from winning by eliminating her in the subelection $(C' \cup \{p\}, V)$. Otherwise, p is a winner in every election $(C' \cup \{p\}, V)$ with $C' \subseteq C \setminus \{p\}$, so control is impossible. ❑

By Fact 1, Borda-DCPC-TP and Borda-DCRPC-TP are identical in the nonunique-winner model.[15] Therefore, Theorem 8 implies Corollary 2, which again is somewhat surprising in light of Theorem 7 (which shows that Borda-DCRPC-TP in the unique-winner model is NP-hard).

Corollary 2. *In the nonunique-winner model, Borda is vulnerable to destructive control by run-off partition of candidates in the ties-promote model.*

3.5. Borda-CCPC-TP and Borda-CCRPC-TP

Finally, we turn to the constructive variants of the above two problems.

---

[15] In the unique-winner model, DCPC-TP and DCRPC-TP are not the same problem: Hemaspaandra et al. [58] design a rather artificial voting rule for which these two problems differ. In fact, one can also prove this claim for a very natural voting rule: Borda. To see that Borda-DCPC-TP and Borda-DCRPC-TP differ in the unique-winner model, consider three candidates, a, b, and c, and let V contain three votes: $a\ b\ c$, $a\ b\ c$, and $b\ c\ a$. It is clear that a is tied with b for the first place if c participates in the election, while c is lagging behind them in points. Therefore, in Borda-DCPC-TP, we can prevent a from winning alone by the trivial partition $(\emptyset, \{a, b, c\})$, i.e., by giving all candidates a bye to the final round. However, in Borda-DCRPC-TP, either b or c is always eliminated in the first round while a always reaches the run-off: Since there are three candidates in total, there must be a first-round subelection with two or three candidates. In $(\{a, b, c\}, V)$, a and b win and c is eliminated, so a and b proceed to the run-off; in $(\{a, b\}, V)$, a wins and b is eliminated, so a faces c (who trivially wins the other first-round subelection) in the run-off; in $(\{a, c\}, V)$, a wins and c is eliminated, so a faces b (who trivially wins the other first-round subelection) in the run-off; and in $(\{b, c\}, V)$, b wins and c is eliminated, so b faces a (who trivially wins the other first-round subelection) in the run-off. In each case, only a wins the run-off and thus cannot be prevented from winning alone in Borda-DCRPC-TP.

Theorem 9. *Borda is resistant to constructive control by partition of candidates in the ties-promote model.*

Proof. To show NP-hardness, we provide a reduction from X3C to Borda-CCPC-TP. Let $(X, \mathscr{S})$ be a given X3C instance with $X = \{x_1, \ldots, x_m\}$, where $m = 3k$ for some $k > 1$, and $\mathscr{S} = \{S_1, \ldots, S_n\}$, where $S_i \subseteq X$ and $|S_i| = 3$ for each i, $1 \le i \le n$. Again, we assume that every $x_i \in X$ appears in exactly three subsets $S_j \in \mathscr{S}$, so $n = 3k$ as well. Construct from $(X, \mathscr{S})$ a Borda-CCPC-TP instance $((C, V), p)$ as follows. Let $C = \{p, r, r^*\} \cup X \cup \mathscr{S}$ with p being the distinguished candidate. Define V to consist of the following votes:

1. There are $2k$ votes of the form $r\ \overrightarrow{X}\ p\ r^*\ \overrightarrow{\mathscr{S}}$ and $2k$ votes of the form $r\ \overleftarrow{X}\ p\ r^*\ \overleftarrow{\mathscr{S}}$.
2. There is one vote $r^*\ \overrightarrow{\mathscr{S}}\ r\ p\ \overrightarrow{X}$ and one vote $r^*\ \overleftarrow{\mathscr{S}}\ r\ p\ \overleftarrow{X}$.
3. For every $S_i = \{x', x'', x'''\} \in \mathscr{S}$, there are
   - $(2k-1)(3k+3)+1$ votes of the form $\overrightarrow{X \setminus S_i}\ r\ r^*\ p\ S_i\ x'\ x''\ x'''\ \overrightarrow{\mathscr{S} \setminus \{S_i\}}$ and
   - $(2k-1)(3k+3)+1$ votes of the form $x'''\ x''\ x'\ p\ r^*\ r\ S_i\ \overleftarrow{X \setminus S_i}\ \overleftarrow{\mathscr{S} \setminus \{S_i\}}$.

It is easy to see that p is beaten by r in every possible subelection and therefore is not winning in (C, V). The intuitive idea is that r is eliminated by r^* in the subelection and, then, p is able to win the final-round election. For r^* to beat r in the subelection, none of X can be part of the subelection, and we also need a specific number of candidates from $\mathscr{S}$ to participate. In order for p to win the final-round election afterwards, we need to carefully choose candidates from $\mathscr{S}$ to participate in the subelection, as they are eliminated, so that the remaining candidates from $\mathscr{S}$ that reach the final-round election enable p to beat all candidates from X. This can only be achieved if the candidates from $\mathscr{S}$ that are given a bye to the final round correspond to an exact cover of X.

We claim that $(X, \mathscr{S})$ is a yes-instance of X3C if and only if $((C, V), p)$ is a yes-instance of Borda-CCPC-TP.

From left to right, suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Then partition C into $C_1 = \{r, r^*\} \cup \mathscr{S} \setminus \mathscr{S}'$ and $C_2 = \{p\} \cup \mathscr{S}' \cup X$. All candidates in C_2 are directly qualified for the final election. Furthermore, r^* is the unique Borda winner of the subelection (C_1, V), since $dist_{(C_1,V)}(r^*, r) = -4k + 4k + 2 = 2$ and $dist_{(C_1,V)}(r^*, S_i) > 0$ for every $S_i \in \mathscr{S} \setminus \mathscr{S}'$. In the final election (C', V) with $C' = \{p, r^*\} \cup \mathscr{S}' \cup X$, we have that p beats every candidate in $\mathscr{S}'$ and the candidate r, since $dist_{(C',V)}(p, r^*) = 4k - 2k - 2 > 0$ (recall that we required $k > 1$). Because $\mathscr{S}'$ is an exact cover, every x_i is contained in exactly one element of $\mathscr{S}'$. Therefore,

$$dist_{(C',V)}(p, x_i) = -2k(3k+1) + (3k+1) + (2k-1)(3k+1) + 1 = 1$$

and p is the unique Borda winner of the final election.

From right to left, suppose that p can be made the only Borda winner by partitioning the candidates. Since p is beaten by r in every possible subelection, r and p need to be in different parts of the partition (say, r is in (C_1, V)), and r needs to be eliminated in the subelection (C_1, V). It is easy to see that r beats all candidates from X and $\mathscr{S}$ in all possible subelections as well. Therefore, $r^* \in C_1$. For subsets $X' \subseteq X$ and $\mathscr{S}' \subseteq \mathscr{S}$, the point balance of r^* and r is

$$dist_{(\{r,r^*\} \cup X' \cup \mathscr{S}', V)}(r^*, r) = -4k|X'| - 4k + 2|\mathscr{S}'| + 2.$$

In order for r^* to beat r, no candidate from X may be in the subelection and at least $2k$ candidates from $\mathscr{S}$ need to participate in it. This leaves candidates $C_1 = \{r^*, p\} \cup X \cup \mathscr{S}'$ with $\mathscr{S}' \subseteq \mathscr{S}$ and $|\mathscr{S}'| \le k$ in the final election. Note that p beats all candidates in $\mathscr{S}'$ and the candidate r^*, since $|\mathscr{S}'| \le k$. Without any $S_i \in \mathscr{S}'$, p has a point deficit of $-2k(3k+1) + (3k+1)$ on every $x_j \in X$. For every $S_i \in \mathscr{S}'$, p gains $(2k-1)(3k+1)+1$ points on every $x_j \in S_i$. Therefore, for p to beat all candidates in X, every $x_i \in X$ needs to be in at least one element of $\mathscr{S}'$. Since $|\mathscr{S}'| \le k$, we have that $\mathscr{S}'$ is an exact cover. ❑

A slight modification of the proof of Theorem 9 yields Corollary 3.

Corollary 3. *Borda is resistant to constructive control by run-off partition of candidates in the ties-promote model.*

4. Complexity of control by partition of voters in Borda elections

In this section we solve the only three problems that still were open for voter control in Borda elections (recall Table 1): constructive control by partition of voters when ties promote or ties eliminate and destructive control by partition of voters when ties promote. We start with the one open case concerning the TE model.

4.1. Borda-CCPV-TE

In Borda-Constructive-Control-by-Partition-of-Voters-TE (Borda-CCPV-TE) we ask, given an election (C, V) and a candidate p in C, whether V can be partitioned into V_1 and V_2 such that p is the unique Borda winner of the two-stage election where only unique Borda winners of subelections (C, V_1) and (C, V_2) proceed to the final run-off.

Theorem 10. *Borda is resistant to constructive control by partition of voters in the ties-eliminate model.*

Proof. To show NP-hardness, we provide a reduction from X3C to Borda-CCPV-TE. Let $(X, \mathscr{S})$ be a given X3C instance with $X = \{x_1, \dots, x_m\}$, where $m = 3k$ for some $k > 1$, and $\mathscr{S} = \{S_1, \dots, S_n\}$, where $S_i \subseteq X$ and $|S_i| = 3$ for each i, $1 \le i \le n$. We again assume that every $x_i \in X$ appears in exactly three subsets $S_j \in \mathscr{S}$. From this restriction it follows that $n = 3k$ as well. Construct from $(X, \mathscr{S})$ a Borda-CCPV-TE instance $((C, V), p)$ as follows. First, we construct a large but still polynomial number of buffer candidates $B = B_1 \cup B_2 \cup \dots \cup B_{6k+3}$ with

- B_{2i}, $1 \le i \le 3k$, each containing $18k^2 + 12k - 1$ candidates;
- B_{2i-1}, $1 \le i \le 3k$, each containing $27k^2 + 18k + 4$ candidates;
- B_{6k+1} containing $36k^3 + 6k^2 - 12k$ candidates;
- B_{6k+2} containing $135k^3 + 90k^2 + 12k$ candidates; and
- B_{6k+3} containing $18k^3 + 30k^2 + 12k - 1$ candidates.

Note that all B_i, $1 \le i \le 6k + 3$, are pairwise disjoint. Let $C = \{p, r, r^*\} \cup X \cup B$ be the set of candidates with p being the distinguished candidate. Define $V = V_1 \cup V_2 \cup V_3 \cup V_4$ to consist of the following four groups of votes:

1. V_1 contains a single vote of the form $r\ B_{6k+1}\ r^*\ B_{6k+2}\ p\ \overrightarrow{X}\ B \setminus (B_{6k+1} \cup B_{6k+2})$.
2. V_2 contains a single vote of the form $r\ B_{6k+3}\ r^*\ \overleftarrow{X}\ p\ B \setminus B_{6k+3}$.
3. V_3 contains a vote v_j of the form
$$X \setminus S_j\ p\ B_{2j-1}\ r^*\ B_{2j}\ r\ x'\ x''\ x'''\ B \setminus (B_{2j-1} \cup B_{2j})$$
for every $S_j = \{x', x'', x'''\} \in \mathscr{S}$.
4. V_4 contains $3k$ votes of the form $r\ \overleftarrow{X}\ p\ r^*\ B$.

Note that in the way these votes are set up, every buffer candidate $b_j \in B$ is behind some candidate from $C \setminus B$ in every vote (as a matter of fact, b_j is behind every candidate from $C \setminus B$ in all votes but one). This lets us conveniently disregard all buffer candidates, since they are eliminated in all possible subelections and can never reach the final run-off. Note further that in the following, for better readability, all points balances are simplified to their shortest forms.

Note that p is not winning in (C, V), since $dist_{(C,V)}(p, r) = -54k^3 - 45k^2 - 3k - 3 < 0$. The general idea of the reduction is that p can beat r^* in a 1-on-1 final-round election (due to the many votes favoring p over r^* in voter groups 3 and 4) but loses to r if they reach the final-round election. Therefore, the votes must be partitioned so that p wins one subelection alone and r^* the other. We will show that in order to achieve this, we need to set the votes in groups 1, 2, and 4 to the subelections in a certain specific way. Then we need at least $2k$ votes of group 3 for r^* to reach the finale while the remaining k votes need to be enough for p to win the other subelection, in particular beating the candidates of X, which is only possible if there is an exact cover of X.

In the following, we will provide a detailed description on why we chose to construct a specific number of buffer candidates. First, consider voter group 4. The reason to have this group is to ensure that p loses against r in the 1-on-1 final-round election due to the $3k$ votes in voter group 3 favoring p over r. In order to make this group as irrelevant to the partitioning process of the subelections as possible, the score difference in the other votes (for elections that include buffer candidates) of the relevant candidates should always be at least as high as the score difference from voter group 4 (i.e., $\gamma = 3k(3k + 2) = 9k^2 + 6k$). For voter group 1, we want this vote to be in the subelection in which p does not win, so B_{6k+2} should contain at least $3k(|B_{2i}| + |B_{2i-1}|) - 1$ candidates so that the score deficit of p to r cannot be compensated by the votes from group 3. Since the vote is now in the subelection in which r^* must win against r, we need their score difference (i.e., $B_{6k+1} + 1$) to be compensated by at least $2k$ votes from group 3, so we must have $|B_{6k+1}| = (2k - 1)(|B_{2i}| + 1)$. Due to the issue with voter group 4 discussed above, we set $|B_{2i}| = 2\gamma - 1$, which gives $|B_{6k+1}| = (2k - 1)2\gamma$. Now, regarding the vote in group 2, this vote is supposed to be in the subelection that p must win in order to give p a score deficit on the candidates in X, so we must prevent it from being set to the other subelection. Therefore, we have $|B_{6k+3}| = (k + 1)(|B_{2i}| + 1) - 1 = (k + 1)2\gamma - 1$, which implies that putting the vote in the subelection together with the vote from group 1 would make it hopeless for r^* to win against r. Lastly, regarding B_{2i-1}, we expect k votes from group 3 to be in the subelection that p alone wins, so we need to compensate (1) the deficit on r from the vote in group 2, (2) votes from group 4 being in the subelection, and (3) the deficit on candidates from X. This is achieved by setting $|B_{2i-1}| = 2\gamma + 4 + \gamma$.

We claim that $(X, \mathscr{S})$ is a yes-instance of X3C if and only if $((C, V), p)$ is a yes-instance of Borda-CCPV-TE.

From left to right, suppose there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Let $\widehat{V} = \{v_j \in V_3 \mid S_j \in \mathscr{S}'\}$. Partition V into $V' = V_1 \cup (V_3 \setminus \widehat{V}) \cup V_4$ and $V'' = V_2 \cup \widehat{V}$. In the subelection (C, V'), r^* beats every other candidate, since

$$dist_{(C,V')}(r^*, r) = 9k^2 + 6k - 1 > 0,$$
$$dist_{(C,V')}(r^*, p) = 81k^3 + 53k^2 + 2k - 2 > 0, \text{ and}$$
$$dist_{(C,V')}(r^*, x_i) \ge 171k^3 + 15k^2 - 55k - 4 > 0 \text{ for every } x_i \in X.$$

In the other subelection, (C, V''), p is the only Borda winner, since

$$\begin{aligned}
dist_{(C,V'')}(p, r^*) &= 27k^3 + 18k^2 + 2k - 1 > 0,\\
dist_{(C,V'')}(p, r) &= 27k^3 - 10k - 1 > 0, \text{ and}\\
dist_{(C,V'')}(p, x_i) &\geq 42k^2 + 33k + 3 > 0 \text{ for every } x_i \in X.
\end{aligned}$$

In the final-stage election $(\{p, r^*\}, V)$, p is the only Borda winner, since

$$dist_{(\{p,r^*\},V)}(p, r^*) = 6k - 2 > 0.$$

For the converse, suppose there is no exact cover. We now show that p cannot be made the only Borda winner by partitioning the votes. Since no buffer candidate reaches the final round and due to the ties-eliminate model, the only possible final-stage elections with p participating are $(\{p, c\}, V)$ with $c \in \{r, r^*\} \cup X$ and $(\{p\}, V)$ (i.e., in the latter case p alone wins one subelection and two or more candidates tie for winner in the other subelection). It is easy to see that p wins alone only if r^* wins the other subelection alone or no one qualifies from the other subelection. Consider any partition (V', V'') of V. Without loss of generality, assume that $V_1 \subseteq V'$. Then p cannot win (C, V'), since the deficit of $171k^3 + 96k^2 + 2$ to r cannot be made up for, not even with all the votes from V_3 from which p would gain only $135k^3 + 90k^2 + 15k$ points on r. Therefore, p can only win (C, V''). For p to beat every $x_i \in X$ in (C, V''), there need to be votes $\widehat{V} \subseteq V_3$ in V'' such that, for every $x_i \in X$, there is a $v_j \in \widehat{V}$ with $x_i \in S_j$. Otherwise, p would be behind x_i in every vote of V''. Since there is no exact cover, we need at least $k + 1$ such votes from $\widehat{V}$ to ensure that p is not beaten by some candidate $x_i \in X$ in (C, V''). In (C, V'), the point deficit resulting from $V_1 \subseteq V'$ of p and candidates from X to r cannot be made up for by at most $2k - 1$ votes from V_3. Notice that all votes from V_2 and V_4 have r at the top position, so even if we assume that none of those votes are in V', with only in the best case $2k - 1$ votes of V_3 being in V', we have $dist_{(C,V')}(r^*, r) \leq -1$. It follows that r^* cannot beat or tie r in (C, V'), which leads to r reaching the final round. Therefore, without an exact cover, either p cannot reach the final round or r does reach it as well. In both cases, p does not win it. ❑

4.2. Borda-CCPV-TP and Borda-DCPV-TP

Lastly, we consider the same problem as above but with the ties-promote (TP) instead of the ties-eliminate (TE) model, and its destructive variant also with TP. We start with the former.

Theorem 11. *Borda is resistant to constructive control by partition of voters in the ties-promote model.*

Proof. To show NP-hardness, we can use exactly the same reduction as in the proof of Theorem 10 that showed NP-hardness of Borda-CCPV-TE.

In the proof of correctness, it was shown there that if the X3C instance is a yes-instance then we can make p be the only winner of one subelection and r^* be the only winner of the other subelection, leading to p alone winning the final run-off.

In the converse direction, it was shown that if the X3C instance is a no-instance, then either p does not reach the final round, or if we can make p win one subelection then inevitably r reaches the final round as well via the other subelection, which leads to p not winning the final round. ❑

Theorem 12. *Borda is resistant to destructive control by partition of voters in the ties-promote model.*

Proof. To show NP-hardness, we provide a reduction from X3C to Borda-DCPV-TP. Let $(X, \mathscr{S})$ be a given X3C instance with $X = \{x_1, \dots, x_m\}$, where $m = 3k$ for some $k > 1$, and $\mathscr{S} = \{S_1, \dots, S_n\}$, where $S_i \subseteq X$ and $|S_i| = 3$ for each i, $1 \leq i \leq n$. Again, we assume that every $x_i \in X$ appears in exactly three subsets $S_j \in \mathscr{S}$ (recall that this implies $n = 3k$ as well). Construct from $(X, \mathscr{S})$ a Borda-DCPV-TP instance $((C, V), p)$ as follows.

We start by constructing a large but still polynomial number of buffer candidates $B = B_1 \cup B_2 \cup \cdots \cup B_{3k+2} \cup \{\widehat{b}_1, \widehat{b}_2, \widehat{b}_3\}$ with B_i, where $1 \leq i \leq 3k$ and $S_i = \{x_p, x_q, x_r\}$ with $p < q < r$, containing $rk - 3$ candidates; B_{3k+1} containing $6k + 3$ candidates; and B_{3k+2} containing $9k^2$ candidates. Note that all B_i, $1 \leq i \leq 3k + 2$, and $\{\widehat{b}_1, \widehat{b}_2, \widehat{b}_3\}$ are pairwise disjoint.

Let $C = \{p, r, r^*\} \cup X \cup B$ with p being the distinguished candidate. For a more convenient construction of votes, we introduce additional notation. If we write $\overrightarrow{X_{|\{x_i,x_j,x_\ell\}}}$ for some $\{x_i, x_j, x_\ell\} \subseteq X$, the candidates of X appear in the vote in the usual order but x_i, x_j, and x_ℓ are replaced with $\widehat{b}_1$, $\widehat{b}_2$, and $\widehat{b}_3$. It is important to note that when $\overrightarrow{X}$ appears in a vote, the candidates of X are ordered from lowest to highest index, whereas they are ordered from highest to lowest index in case of $\overleftarrow{X}$.

Now, define $V = V_1 \cup V_2 \cup V_3 \cup V_4$ to consist of the following four groups of votes:

1. V_1 contains $3k + 2$ votes of the form $p\ B_{3k+1}\ r\ \overleftarrow{X}\ B \setminus B_{3k+1}$.

2. V_2 contains two votes of the form $r\ \widehat{b}_1\ \overrightarrow{X}\ p\ B \setminus \{\widehat{b}_1\}$.
3. For every $S_j = \{x_p, x_q, x_r\} \in \mathscr{S}$ with $p < q < r$, V_3 contains a vote v_j that is constructed in the following way: Set x_r on position one, then $(r-q)k-1$ buffer candidates from B_j, then x_q (at position $(r-q)k+1$), then $(q-p)k-1$ buffer candidates from B_j, then x_p (at position $(r-p)k+1$), then the remaining $pk-1$ buffer candidates from B_j, and from position $rk+1$ onwards the vote has the form $p\ r\ \overrightarrow{X_{|\{x_p,x_q,x_r\}}}\ B \setminus (B_j \cup \{\widehat{b}_1, \widehat{b}_2, \widehat{b}_3\})$.
4. V_4 contains $3k+1$ votes of the form $r\ p\ B_{3k+2}\ \overrightarrow{X}\ B \setminus B_{3k+2}$ and $3k+1$ votes of the form $p\ r\ B_{3k+2}\ \overleftarrow{X}\ B \setminus B_{3k+2}$.

The votes in V_3 are set up in such a way that for a vote $v_j \in V_3$, if $x_i \in S_j$ then $dist_{(C,\{v_j\})}(r, x_i) = -(ik+1)$, and if $x_i \notin S_j$ then $dist_{(C,\{v_j\})}(r, x_i) = i$. This is illustrated by the following Example 3.

Example 3. Let $k = 2$ and $S_1 = \{x_2, x_3, x_5\} \in \mathscr{S}$. Then B_1 consists of $5 \cdot 2 - 3 = 7$ candidates, denoted by $b_1, \ldots, b_7$. With $p = 2$, $q = 3$ and $r = 5$ the following vote for the third voter group is constructed:

$$x_5\ b_7\ b_6\ b_5\ x_3\ b_4\ x_2\ b_3\ b_2\ b_1\ p\ r\ x_1\ \widehat{b}_1\ \widehat{b}_2\ x_4\ \widehat{b}_3\ x_6\ B \setminus (B_1 \cup \{\widehat{b}_1, \widehat{b}_2, \widehat{b}_3\}).$$

Then we have the following point balances of r and the candidates in X:

i	1	2	3	4	5	6
$dist(r, x_i)$	1	−5	−7	4	−11	6

Note that every buffer candidate $b_i \in B$ is behind one candidate from $C \setminus B$ in all votes. Therefore, no buffer candidate b_i ever survives a subelection and we can disregard their scores.

Lemma 6. *For any partition of votes (V', V''), p always is the unique Borda winner of one subelection.*

Proof. Let (V', V'') be a partition of V. Since V_1 contains $3k+2$ votes, there is at least one part of the partition with at least k votes of V_1, let us say V'. From these votes, p is ahead of r by at least $k(6k+4)$ points and ahead of every $x_i \in X$ by at least $k(6k+5)$ points in (C, V'). Even if the other votes in V' all rank r ahead of p, r can only gain at most $3k+1+6k+4$ points on p, which is not enough to at least tie p. For each $x_i \in X$, if the other votes in V' all rank x_i ahead of p then x_i can only gain at most $2 + 3k^2 + 1$ points on p, which is not enough to at least tie p. Therefore, p is the unique Borda winner of (C, V'). ❑ Lemma 6

In election (C, V), p is the unique Borda winner because

$$dist_{(C,V)}(p, r) = (3k+2)(6k+4) - 2(3k+2) + 3k > 0 \text{ and}$$
$$dist_{(C,V)}(p, x_i) = 54k^3 + 54k^2 + (42-3i)k + (9-3i) > 0 \text{ for every } x_i \in X.$$

The intuitive idea of the reduction is that p can only be tied for winning with r in the final-round election if r and all of X proceed from the subelections. Since p wins at least one subelection alone, it follows that r and all of X must tie for winning the other subelection. This is only possible if we balance the score of r and all of X by compensating the score deficit of candidates of X on r from votes of group 2 with votes from group 3. We need voter group 4 to ensure that no candidate from X can beat p in a final-round election, and voter group 1 ensures that p wins at least one subelection alone and can at most be tied with r in the final-round election.

We claim that $(X, \mathscr{S})$ is a yes-instance of X3C if and only if $((C, V), p)$ is a yes-instance of Borda-DCPV-TP.

From left to right, suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Let $\widehat{V} = \{v_j \in V_3 \mid S_j \in \mathscr{S}'\}$. Partition V into $V' = V_1 \cup \widehat{V} \cup V_4$ and $V'' = V_2 \cup (V_3 \setminus \widehat{V})$

In the second subelection (C, V''),

$$dist_{(C,V'')}(r, p) = 6k + 4 - 2k = 4k + 4 > 0 \text{ and}$$
$$dist_{(C,V'')}(r, x_i) = 2i + 2 - 2(ik+1) + (2k-2)i = 0 \text{ for every } x_i \in X,$$

since $\mathscr{S}'$ is an exact cover, so every $x_i \in X$ is covered exactly twice in $S \setminus \mathscr{S}'$. Therefore, r and all $x_i \in X$ tie and proceed to the final round according to the TP rule. From Lemma 6 it follows that p wins (C, V') alone. Then r ties p in the final election, since $dist_{(\{p,r\}\cup X,V)}(p, r) = 2 \cdot 3k - (2|X|) = 0$.

From right to left, suppose that p can be prevented from being the only Borda winner by partitioning the votes. From Lemma 6 it follows that p always reaches the final election. Since no buffer candidate survives the subelections, for a subset $X' \subseteq X$ the only possible final elections are $(\{p, r\} \cup X', V)$ and $(\{p\} \cup X', V)$. In $(\{p, r\} \cup X', V)$,

$$dist_{(\{p,r\}\cup X',V)}(p, r) = 2 \cdot 3k - (2|X'|) \text{ and}$$
$$dist_{(\{p,r\}\cup X',V)}(p, x_i) \geq 2(3k+2) - 11 > 0 \text{ for every } x_i \in X'.$$

Therefore, p is not the only winner if r and all candidates in X reach the final run-off. In $(\{p\} \cup X', V)$,

$$dist_{(\{p\}\cup X',V)}(p, x_i) \geq (3k+2) - 11 + (3k+1) > 0$$

for every $x_i \in X'$. Therefore, p wins the final election alone if r or any x_i fail to reach the final round. From Lemma 6 it follows that r and all $x_i \in X$ need to tie in one subelection. Without loss of generality, assume that this subelection is (C, V'). Then no vote from V_1 can be in V' (or else p would beat r), and no vote from V_4 can be in V' (or else p would beat at least one $x_i \in X$). Thus V' consists of votes from V_2 and V_3. If there is no vote from V_2 in V' then r is beaten by p; otherwise, r always beats p. We will now show that in both cases (only one vote of V_2 is in V' and both votes of V_2 are in V'), an exact cover needs to exist in order for p and all candidates of X to tie in (C, V').

Case 1: If V' contains one vote from V_2 then r scores $i+1$ points more than every $x_i \in X$. To tie r and all x_i, there need to be votes $\widehat{V} \subseteq V_3$ in V'. Let $\mathscr{S}' = \{S_j \mid v_j \in \widehat{V}\} \subseteq \mathscr{S}$ and $S_{x_i} = \{S_j \in \mathscr{S}' \mid x_i \in S_j\}$. $|S_{x_i}|$ indicates how many times x_i is hit by $\mathscr{S}'$.

- If $|S_{x_i}| = 0$ for a $x_i \in X$ then $dist_{(C,V')}(r, x_i) > 0$. Therefore, $|\widehat{V}| = |\mathscr{S}'| \geq k$.
- If $|S_{x_i}| = 1$ then
$$dist_{(C,V')}(r, x_i) = -(ik+1) + i + 1 + (|\mathscr{S}'| - 1)i = 0$$
for $|\mathscr{S}'| = k$ and all i.
- If $|S_{x_i}| = 2$ then
$$dist_{(C,V')}(r, x_i) = -2(ik+1) + i + 1 + (|\mathscr{S}'| - 2)i = -(2k-1)i - 1 + (|\mathscr{S}'| - 2)i \neq 0$$
for $i \neq 1$.
- If $|S_{x_i}| = 3$ then
$$dist_{(C,V')}(r, x_i) = -3(ik+1) + i + 1 + (|\mathscr{S}'| - 3)i \leq -(3k-1)i - 2 + (3k-3)i = -2i - 2 < 0.$$

Therefore, if V' consists of one vote of V_2, r and all $x_i \in X$ can only tie if there exists an exact cover.

Case 2: If V' contains both votes from V_2 then r scores $2i+2$ points more than every $x_i \in X$.

- If $|S_{x_i}| = 0$ for a $x_i \in X$ then $dist_{(C,V')}(r, x_i) > 0$.
- If $|S_{x_i}| = 1$ then
$$\begin{aligned} dist_{(C,V')}(r, x_i) &= -(ik+1) + 2i + 2 + (|\mathscr{S}'| - 1)i \\ &= -(k-2)i + 1 + (|\mathscr{S}'| - 1)i \geq -(k-2)i + 1 + (k-1)i = 1 + i > 0. \end{aligned}$$
- If $|S_{x_i}| = 2$ then
$$dist_{(C,V')}(r, x_i) = -2(ik+1) + 2i + 2 + (|\mathscr{S}'| - 2)i = -(2k-2)i + (|\mathscr{S}'| - 2)i = 0$$
for $|\mathscr{S}'| = 2k$ and all i. This would mean that $\mathscr{S} \setminus \mathscr{S}'$ is an exact cover.
- If $|S_{x_i}| = 3$ then
$$dist_{(C,V')}(r, x_i) = -3(ik+1) + 2i + 2 + (|\mathscr{S}'| - 3)i \leq -(3k-2)i - 1 + (3k-3)i = -i - 1 < 0.$$

Therefore, if V' consists of two votes of V_2, r and all $x_i \in X$ can only tie if there exists an exact cover.

This completes the proof. ❑

5. Online control in sequential Borda elections

Finally, we turn to online control in sequential Borda elections. Hemaspaandra et al. proposed frameworks to model online manipulation [60], online voter control [62], and online candidate control [61] in sequential elections. We start with online control of candidate-sequential Borda elections.

5.1. Online candidate control in sequential Borda elections

We first describe the model and the related problems that are due to Hemaspaandra et al. [61], who also provide motivating examples for these control scenarios in detail, ranging from TV singing/dancing talent shows to university faculty-hiring processes. Specifically, we restrict ourselves to formalizing *online constructive control by deleting candidates for Borda*, denoted by online-Borda-CCDC. The corresponding problem for adding candidates (online-Borda-CCAC) and their destructive counterparts (online-Borda-DCDC and online-Borda-DCAC) can be defined analogously. Capturing the election chair's "moment of decision," an input to online-Borda-CCDC encodes the history of the sequential election process up to a given point in time and specifically consists of:

- the candidate set C,
- the set of voters V,
- the chair's ideal ranking σ of the candidates,
- a distinguished candidate $d \in C$,
- an order in which the candidates will show up, with flags indicating who the current candidate is and which of the previous candidates have been deleted,
- the voters' preferences masked down to the still-standing (i.e., already revealed but not deleted) candidates, and
- a nonnegative integer bound k on how many deletions are left for the chair to use.

The question the chair now faces is whether she has a "forced win" by either deleting (if possible) or not deleting the current candidate right in this moment (she will never again have this choice about this current candidate), where by "forced win" we mean whether the set $\{c \mid c \geq_\sigma d\}$ will contain a Borda winner eventually, *no matter what voter preferences will be revealed about the future candidates who have not shown up yet*. A more formal way of defining the notion of a "forced win" would be to phrase the question as follows: Does there exist a decision about the current candidate such that for all possible voter preferences that might be revealed until the next moment of decision for the chair, there is a decision about the next candidate such that ... such that the set $\{c \mid c \geq_\sigma d\}$ will contain a Borda winner?

Briefly, for online-Borda-CCAC the input now contains a "certainly in the election" set of candidates and a (disjoint) set of "potentially additional" candidates (which we also refer to as *spoiler* candidates), who may be added to the election by the chair as soon as they are revealed. The presentation order refers to both candidate sets and the rest stays the same (so for already revealed spoiler candidates, there is a flag indicating whether they have been added or not, etc.).

The destructive variants of online-Borda-CCDC and online-Borda-CCAC have the same input as their constructive counterparts but here we ask whether the chair can make sure that no candidate of $\{c \mid d \geq_\sigma c\}$ is a Borda winner at the end of the voting process. For *destructive* online control by deleting candidates, Hemaspaandra et al. [61] distinguish the *non-hand-tied chair model* where the chair may delete some but not all candidates "d or worse" and the *hand-tied chair model* where the chair may never delete any candidate "d or worse."

Theorem 13. *Each of online-Borda-*CCDC*, online-Borda-*DCDC *(both in the non-hand-tied and the hand-tied chair model), online-Borda-*CCAC*, and online-Borda-*DCAC *belongs to* P*.*

Proof. We restrict ourselves to giving the proof details for the result that online-Borda-CCDC is in P. The other proofs are similar and we will only outline the key differences.

Given an input to online-Borda-CCDC as described above (using the same notation, e.g., d denoting the distinguished candidate and σ the chair's ideal ranking of the candidates), we give a polynomial-time algorithm that decides whether the chair has a forced win by either deleting or not deleting the current candidate, c. In fact, since the chair is facing these two options (to delete or not to delete c—unless the number of allowed deletions is used up already in which case c must be left in) now, our algorithm (to be described below) will be run twice, first pretending the chair's decision were to leave c in, then pretending the chair's decision were to remove c, and if at least one run yields a forced win for the chair, the input is accepted; otherwise it is rejected.

We call each $e \in C$ with $e \geq_\sigma d$ a *good* candidate and each $e \in C$ with $d >_\sigma e$ a *bad* candidate. Let b be the number of future (i.e., as yet unrevealed) bad candidates and let g be the number of future good candidates. Recall that all votes at this point are masked down to the still-standing candidates (but will be gradually extended when new candidates show up). Our polynomial-time algorithm now works as follows.

If there is no voter, every candidate still standing in the end is a Borda winner with score zero, so we accept if there is a good candidate among those, and otherwise we reject. Further, in case all candidates have been revealed in the current moment, we simply determine their scores, and we accept if a good candidate has the highest score; otherwise, we reject. So from now on we may assume that there is at least one voter and not all candidates have shown up yet.

We now determine the scores of all already revealed but not deleted candidates. If no good candidate has currently the highest score, the chair does not have a forced win: It may happen, for instance, that all future candidates will be ranked below all currently revealed candidates in the completed votes in the end, which would mean that all currently revealed candidates score the same number of points more than they have now, and they each score more points than any future candidate, since these are ranked lower in each vote, so it is still true that no good candidate is a Borda winner, and we reject. (In this case, it doesn't matter whether future candidates will be deleted or not.)

Consider now the case that at least one good candidate is currently winning. Let $k \geq 0$ be the number of deletions left for the chair to use.

If $k < b$ then there is at least one future bad candidate that cannot be removed. In the worst case, one such candidate ends up in the top positions of all completed votes in the end and thus is the only Borda winner, so the chair does not have a forced win and we reject.

If $k \geq b$, however, all future bad candidates can be deleted by the chair, who then is left with $k - b \geq 0$ remaining possible deletions. If none of the previously revealed candidates is bad, only good candidates remain in the election and at least one of them will be a Borda winner in the end, so the chair has a forced win and we accept.

Therefore, we now consider the final case that a good candidate currently has the highest score, all future bad candidates can be deleted with $k - b \geq 0$ possible deletions remaining for the chair, and at least one bad candidate was previously revealed and not deleted (and so will remain in the election). For each future good candidate who will not be deleted, every other candidate can score at most one additional point in each vote, depending on their relative position in the votes. In order to spoil a forced win for the chair, some previously revealed and not deleted bad candidate would have to score enough additional points due to the $g - (k - b)$ future good candidates that cannot be deleted so as to have more points than each good candidate in the end.

Let i be a bad candidate still in the election and let s be a future good candidate. When s is being revealed, then i makes up one point with respect to a good candidate j if there is a vote of the form $\cdots\ i\ \cdots\ s\ \cdots\ j\ \cdots$. That is, if such a bad candidate's deficit regarding the good candidates is not too large and there are sufficiently many votes of this form for all good candidates, the bad candidate can still become a unique Borda winner in the worst case (for the chair). We may assume that the revealed good candidates that won't be deleted, call them $s_1, \ldots, s_{g-(k-b)}$, occur directly behind the bad candidate i in these votes: $\cdots\ i\ s_1\ \cdots\ s_{g-(k-b)}\ \cdots$. For a good candidate j, let $v_{i,j}$ be the number of votes in which i precedes j. Then i can make up $v_{i,j} \cdot (g - (k - b))$ points with respect to j in the worst case. From the remaining $|V| - v_{i,j}$ votes, both i and j would gain the same number of points. Therefore, all we need to check is whether there is a bad candidate i still in the race such that for all good candidates j currently in the election, $score(i) + v_{i,j} \cdot (g - (k - b)) > score(j)$, where $score(h)$ denotes candidate h's current score. If so, i can become a unique Borda winner in the end, which spoils the chair's forced win, so we reject. Otherwise, for each bad candidate there is a good candidate whose score is at least as high in the end, even if the candidates still to be revealed and not deleted will be in the worst positions for this good candidate: The chair has a forced win and we accept. This completes the description of our polynomial-time algorithm for online-Borda-CCDC and the (implicit) proof of correctness.

For online-Borda-CCAC, a key difference is that we have two types of future candidates: *qualified* candidates who are certainly in the election and *spoiler* candidates that may or may not be added by the chair. Note that the chair does not need to add any spoiler candidates (only the number of added candidates is limited to k). In fact, we may assume that the chair will never add future spoiler candidates, as even future good candidates can potentially reduce other good candidates' chances of winning. On the contrary, the chair has no control over qualified candidates meaning every qualified bad candidate, who has not been revealed yet, will make the chair's goal impossible to reach in the worst case. Therefore, instead of "future bad candidates that cannot be removed" we speak of "future bad candidates that are not spoiler candidates." Now consider the case in our algorithm that a good candidate is currently winning (and no future bad candidate can enter the election). Instead of "future good candidates that cannot be removed" we need to check if "future good candidates that are qualified" can make a bad candidate defeat all other good candidates in the worst case.

For the destructive variants, we consider the candidates in $\{c \mid c >_\sigma d\}$ to be good candidates and candidates in $\{c \mid d \geq_\sigma c\}$ to be bad candidates. Additionally, a good candidate needs to defeat all bad candidates for the chair's goal to be reached (in the constructive variant it was sufficient for a good candidate to at least tie all bad candidates). This entails slight changes to the algorithm of the respective constructive variant (e.g., the case that required a good candidate to be currently winning now requires that no bad candidate is currently winning, or in the last case, the inequality is no longer required to be strict).

For online-Borda-DCDC, special attention needs to be paid to the *hand-tied chair model* (where the chair cannot delete any bad candidate) and the *non-hand-tied chair model* (where the chair may delete all but one bad candidate). In the case that no bad candidate is currently winning, if $b = 0$ (i.e., if there is no future bad candidate) nothing needs to be done. Otherwise (i.e., if $b > 0$), reject the input in the *hand-tied chair model*, and in the *non-hand-tied chair model* reject the input only if there is no bad candidate still standing. The input is rejected in those cases because there would be future, bad candidates that the chair cannot delete (recall that one of those candidates could be ranked on top of all votes in the worst case and thus would be the only winner). ❑

5.2. Online voter control in sequential Borda elections

Again, we start by describing the model proposed by Hemaspaandra et al. [62], now focusing on the formalization of *online constructive control by deleting voters for Borda*, denoted by online-Borda-CCDV. We are given a basic *online voter control setting* (*OVCS*), (C, u, V, σ, d), consisting of a set C of candidates, the current voter u, an election snapshot $V = (V_{<u}, u, V_{u<})$ for C and u, the chair's ideal preference order σ on C, and a distinguished candidate $d \in C$. Here, the earlier voters $V_{<u}$ have already cast their votes (i.e., their preference orders over C), now it is u's turn to cast a vote (which the chair of course knows), and the future voters $V_{u<}$ will cast their (as yet unknown) votes in the specified order. The chair's goal is to make sure, by exerting an appropriate control action (here: possibly deleting u), that the ultimate election resulting from all the chair's decisions and the votes cast by the voters will have a set of winners that has a nonempty intersection with $\{c \mid c \geq_\sigma d\}$ (i.e., with the candidates the chair likes at least as much as d). In online-Borda-CCDV, every voter before u is marked as either being deleted or not and we are also given a deletion upper bound k (i.e., the chair is allowed to delete at most k voters). Given an OVCS (C, u, V, σ, d) and k, the question is whether—no matter what votes the future voters after u will cast—it is possible for the chair to reach her goal by her current decision as to whether or not to delete u and by her future decisions regarding deleting future voters (no more than k in total), each being made with the chair's then-in-hand knowledge about what votes will have been cast by then.

In the corresponding destructive case, online-Borda-DCDV, the only difference is that the chair's goal now is to ensure that none of the candidates in $\{c \mid d \geq_\sigma c\}$ wins the ultimate election.

In the corresponding cases for adding voters, online-Borda-CCAV and online-Borda-DCAV, k is now an upper bound on how many (as yet unregistered) voters may be added to the election by the chair, and each voter before u is marked as either originally registered or not (note that only originally unregistered voters can be added); in particular, u must be unregistered for the chair to be able to make a decision regarding adding u.

Finally, in the corresponding cases for partition of voters, online-Borda-CCPV and online-Borda-DCPV, each voter before u is marked as being either in the left side of the partition or in the right side, and the chair's decision now refers to whether the current voter u will be assigned to either the left or the right side of the partition. (As in the offline-control analogues, Borda-CCPV and Borda-DCPV, this voter partition is used for a two-stage election, where a tie-handling rule—TE or TP—determines which winners of the two first-stage subelections will proceed to the final round.[16])

5.2.1. Online control by adding and by deleting voters

We now show that constructive and destructive online voter control by either adding or deleting voters are coNP-hard problems. To show Theorem 14, we first define the problem Borda-Constructive-Coalitional-Unweighted-Manipulation (Borda-CCUM; recall this problem from Section 1.1.1): Given an election (C, V), a distinguished candidate $p \in C$, and an integer t (the number of manipulators), we ask whether the manipulators can cast votes V', $|V'| = t$, such that p is a Borda winner of the election $(C, V \cup V')$. Additionally, we denote by Borda-CCUM$_{\text{UW}}$ the unique-winner variant of Borda-CCUM in which a manipulation is considered successful only if the manipulators succeed in making p a *unique* winner of the election. Borda-CCUM is NP-complete even if $t = 2$, which was shown independently by Davies et al. [27] (see also [28]) and Betzler et al. [13]. The proof by Davies et al. [27,28] can be slightly modified to show that Borda-CCUM$_{\text{UW}}$ is NP-complete as well. We will reduce from the complement of this problem to show coNP-hardness of constructive and destructive online control by adding and deleting voters in sequential Borda elections. Since each of these problems clearly is in NP, we even have coNP-completeness.

Theorem 14. *Each of online-Borda-*CCDV*, online-Borda-*DCDV*, online-Borda-*CCAV*, and online-Borda-*DCAV *is* coNP*-hard.*

Proof. To show that online-Borda-CCDV is coNP-hard, we reduce the complement of Borda-CCUM$_{\text{UW}}$ to this problem. Let $((C', V'), p, t)$ be a Borda-CCUM$_{\text{UW}}$ instance. We construct an OVCS (C, u, V, σ, d) as follows. Let $C = C'$ be the set of candidates, $V_{<u} \cup V_u = V'$ the election snapshot for C and the current voter u, the chair's preference order $\sigma = C \setminus \{p\}\ p$, and d is set to the last candidate in $C \setminus \{p\}$ (in the chair's ideally desired order σ). The chair's goal is to ensure that one candidate from $C \setminus \{p\}$ wins the election. Furthermore, suppose that $V_{u<}$ contains t voters, the deletion limit k is set to 0, and all voters from $V_{<u}$ are marked as undeleted.

Since $k = 0$, the voter u and all t future voters from $V_{u<}$ cannot be deleted from the election. Therefore, the chair has a forced win if and only if there do not exist t votes such that p is a unique winner of (C, V). This is the case if and only if (C', V', p, t) is a no-instance of Borda-CCUM$_{\text{UW}}$.

For the destructive variant, online-Borda-DCDV, we reduce the complement of Borda-CCUM (in the nonunique-winner model) to this problem and construct the same instance as above, except that we now set $d = p$ so that the chair's goal is to prevent p from winning. With the same argument as above, the chair has a forced win if and only if there do not exist t votes such that p is a winner of (C, V), which is the case if and only if (C', V', p, t) is a no-instance of Borda-CCUM.

Note that coNP-hardness of online-Borda-CCAV and online-Borda-DCAV can be shown similarly. A key difference is that in the constructed instance $V_{<u} = V'$ and V_u can be set to any vote since V_u can never be added to the election. Furthermore, all votes except V_u are marked as registered voters. ❑

5.2.2. Online control by partition of voters

We now turn to online control by partition of voters, starting with the ties-promote rule (which, recall Footnote 16, fits more naturally with the nonunique-winner model). To show that both the constructive and the destructive problem is coNP-hard, we will reduce from the complement of the NP-complete problem Permutation-Sum, which is defined as follows:

Permutation-Sum	
Input:	Given a nondecreasingly ordered sequence of n integers, $X_1 \leq \cdots \leq X_n$ with $\sum_{i=1}^{n} X_i = n(n+1)$.
Question:	Do there exist two permutations, π and δ, of $\{1, \ldots, n\}$ such that $\pi(i) + \delta(i) = X_i$ for every i, $1 \leq i \leq n$?

Permutation-Sum was proven to be NP-complete by Yu et al. [100] and was used both by Betzler et al. [13] and by Davies et al. [27,28] to prove NP-hardness of Borda-CCUM.

[16] Hemaspaandra et al. [62] note that the TP model fits more naturally with the nonunique-winner model in which they define their online control problems: For a constructive online control action to be successful, it is enough that some candidate "d or better" wins.

Theorem 15. *Both online-Borda-*CCPV-TP *and online-Borda-*DCPV-TP *are* coNP*-hard.*

Proof. To show that online-Borda-CCPV-TP is coNP-hard, we will reduce the complement of Permutation-Sum to this problem. Given a Permutation-Sum instance $(X_1, \dots, X_n)$, we construct an instance of online-Borda-CCPV-TP as follows. The set of candidates is $\{p, a, x_1, \dots, x_n, x_{n+1}\}$. The current voter is u. The voters prior to u on the left side of the partition vote as follows:

number	vote	for
$n+3$	$W(x_i, a)$	$1 \le i \le n+1$
1	$W(p, x_i)$	$1 \le i \le n+1$
$n+2$	$p\ x_1\ \cdots\ x_n\ x_{n+1}\ a$	
$n+2$	$x_{n+1}\ x_n\ \cdots\ x_1\ p\ a$	

The voters prior to u on the right side of the partition cast the following votes:

number	vote	for
$3n+8-X_i$	$W(x_i, a)$	$1 \le i \le n$
$3n+8$	$W(x_{n+1}, a)$	
1	$p\ a\ x_1\ \cdots\ x_n\ x_{n+1}$	
$n+2$	$p\ x_1\ \cdots\ x_n\ x_{n+1}\ a$	
$n+2$	$x_{n+1}\ x_n\ \cdots\ x_1\ p\ a$	

Now, u casts the vote $p\ a\ x_{n+1}\ x_n\ \cdots\ x_1$, and there are another two voters coming up after u. The chair's ideal preference is $\sigma = x_1\ \cdots\ x_n\ x_{n+1}\ a\ p$, and the distinguished candidate is x_{n+1} (i.e., the chair wins if and only if one of $x_1, \dots, x_n, x_{n+1}$ wins the election). We will need the following lemma.

Lemma 7. *For any partition of u and the two future voters, if p reaches the final round, then p alone wins the election, and otherwise at least one of $x_1, \dots, x_n, x_{n+1}$ wins.*

Proof of Lemma 7. First note that a cannot win in any of the two subelections, since a already has a deficit of more than $2n+4$ points on every other candidate in both subelections (including u's vote) and can gain no more than $2n+2$ points on any candidate from the remaining two voters. This implies that if p is in the final election, only a subset $X' \subseteq \{x_1, \dots, x_n, x_{n+1}\}$ of candidates can move forward to the final round. From the votes of the voters up to and including u, p gains $2|X'|+2$ points more than any candidate from X'. Since there are only $|X'|+1$ candidates remaining, p can lose at most $2|X'|$ points on another candidate from the two voters after u. Therefore, p alone wins the final election.

If p did not reach the final round, the candidates from $\{x_1, \dots, x_n, x_{n+1}\}$ must have won both subelections ahead of p (recall that a is always eliminated) and are promoted to the final run-off according to the ties-promote model. Therefore, if only a subset of $\{x_1, \dots, x_n, x_{n+1}\}$ participates in the final election (and not p), at least one of them wins. ❑ Lemma 7

Proceeding with the proof of Theorem 15, we will now show that there are no two permutations, π and δ, of $\{1, \dots, n\}$ such that $\pi(i) + \delta(i) = X_i$ if and only if at least one of $x_1, \dots, x_n, x_{n+1}$ is a winner of the constructed election.

From left to right, suppose that there are no two permutations, π and δ, of $\{1, \dots, n\}$ satisfying $\pi(i) + \delta(i) = X_i$. Since $\sum_{i=1}^{n} X_i = n(n+1)$, we have that for every two permutations, π and δ, of $\{1, \dots, n\}$, there is some i, $1 \le i \le n$ with $\pi(i) + \delta(i) > X_i$. Set u and the two voters after u to the right side of the partition. We will show that p will be eliminated in both sides of the partition.

From the left side, all candidates from $\{x_1, \dots, x_n, x_{n+1}\}$ proceed to the final round since every such candidate scores one point more than p.

On the right side, disregarding the two voters after u for now, every $x_i \in \{x_1, \dots, x_n\}$ scores $2n+4-X_i$ points more than p and x_{n+1} scores $2n+4$ points more than p. Now, regarding the two voters after u, in order for p to not be beaten by x_{n+1}, p needs to be in the top position of both votes and x_{n+1} must be ranked last. We may assume that a is on the second place in both votes since p beats a anyway. Then every $x_i \in \{x_1, \dots, x_n\}$ gains between 1 and n points from the two voters after u. Let $\pi(i)$ be the points candidate x_i gains from the first voter after u, and let $\delta(i)$ be the points candidate x_i gains from the second voter after u. Note that $\pi = (\pi(1), \dots, \pi(n))$ and $\delta = (\delta(1), \dots, \delta(n))$ are permutations of $\{1, \dots, n\}$. Then p gains $2n+4-(\pi(i)+\delta(i))$ points on every x_i. For the sake of contradiction, assume that p can at least tie with all candidates x_i, $1 \le i \le n$. Then it must hold that

$$2n+4-(\pi(i)+\delta(i)) \ge 2n+4-X_i$$

for every i, $1 \le i \le n$. This gives $\pi(i)+\delta(i) \le X_i$ for every i, $1 \le i \le n$, which contradicts the assumption that there are no two permutations, π and δ, of $\{1, \dots, n\}$ such that $\pi(i)+\delta(i) = X_i$. Therefore, p is beaten by at least one candidate of $\{x_1, \dots, x_n\}$ and is eliminated in the right side of the partition. Therefore, p will not reach the final round and at least one of $x_1, \dots, x_n, x_{n+1}$ wins according to Lemma 7, achieving the chair's goal.

From right to left, suppose that there are two permutations, π and δ, of $\{1,\ldots,n\}$ such that $\pi(i)+\delta(i)=X_i$. Let the voters after u cast the votes $p\ a\ x_{\pi^{-1}(n)}\ \cdots\ x_{\pi^{-1}(1)}\ x_{n+1}$ and $p\ a\ x_{\delta^{-1}(n)}\ \cdots\ x_{\delta^{-1}(1)}\ x_{n+1}$. Note that every $x_i \in \{x_1,\ldots,x_n\}$ gains $\pi(i)+\delta(i)$ points from those votes.

We will show that, no matter how the voters are partitioned, p always reaches the final round (and thus, by Lemma 7, none of $x_1,\ldots,x_n,x_{n+1}$ wins, so the chair's goal cannot be achieved).

If one of u or the two voters after u is set to the left side of the partition, p will always gain at least one point more than any other candidate. Therefore, she will at least tie with every other candidate and proceed to the final round. On the other hand, if all of these three voters are set to the right side of the partition, the point balance of p and x_{n+1} is

$$n+4+(2n+4)-(3n+8)=0$$

and the point balance of p and every $x_i \in \{x_1,\ldots,x_n\}$ is

$$-(3n+8-X_i)+(n+4)+(2n+4-(\pi(i)+\delta(i)))=X_i-(\pi(i)+\delta(i))=0.$$

Therefore, p at least ties with every other candidate and thus proceeds to the final round.

For the destructive case, set the chair's distinguished candidate to p, so the chair's goal is now reached if p is not a winner of the election. It can be shown again, using Lemma 7, that at least one of $\{x_1,\ldots,x_n,x_{n+1}\}$ is a winner of the constructed election if and only if p is not a winner. Therefore, the reduction above can be used to show coNP-hardness of online-Borda-DCPV-TP as well. ❑

Finally, we show that the above proof can be slightly modified so as to also work for online control by partition of voters with the ties-eliminate rule.

Theorem 16. *Both online-Borda-*CCPV-TE *and online-Borda-*DCPV-TE *are* coNP*-hard.*

Proof. We will only outline the key differences in the reduction and in the proof of correctness.

Consider the voters prior to u. On the left side of the partition, instead of $n+3$ pairs of votes $W(x_i,a)$ for every i, $1 \le i \le n+1$, there are now $n+2$ pairs of votes $W(x_i,a)$ for each i, $1 \le i \le n$, and $n+3$ pairs of votes $W(x_{n+1},a)$; this implies that p ties all candidates $x_1,\ldots,x_n$ and misses one point on x_{n+1}. On the right side of the partition, there are now $3n+7-X_i$ instead of $3n+8-X_i$ pairs of votes $W(x_i,a)$ for $1 \le i \le n$, and $3n+7$ instead of $3n+8$ pairs of votes $W(x_{n+1},a)$; thus every candidate $x_1,\ldots,x_{n+1}$ scores one point fewer than before while p's score remains unchanged.

Since there may not be a first-round subelection winner with the TE rule, Lemma 7 does no longer hold and we cannot use it. Other than that the proof proceeds almost as before.

For the left-to-right direction, the same strategy (put u and both future votes to the right side of the partition) can be used to make at least one of $x_1,\ldots,x_{n+1}$ a winner of the final round. Note that from the left side of the partition now only x_{n+1} proceeds to the final round. On the right of the partition, with the same argumentation as before it can be shown that p now cannot defeat (instead of tie-or-defeat) all candidates $x_1,\ldots,x_{n+1}$ if the Permutation-Sum instance is a no-instance. Therefore, either no candidate (if there is a tie for the win) or a candidate from $x_1,\ldots,x_n$ proceeds to the final round from the right side of the partition. Then only x_{n+1} and possibly another candidate from $x_1,\ldots,x_n$ are in the final round.

For the right-to-left direction, again, let the voters after u cast the votes $p\ a\ x_{\pi^{-1}(n)}\ \cdots\ x_{\pi^{-1}(1)}\ x_{n+1}$ and $p\ a\ x_{\delta^{-1}(n)}\ \cdots\ x_{\delta^{-1}(1)}\ x_{n+1}$. Then none of u or the two future voters can be set to the left side of the partition, or p wins this first-round subelection and afterwards also wins the final round alone. On the other hand, if all three votes are set to the right side of the partition, p beats all candidates $x_1,\ldots,x_{n+1}$ by one point and wins the final round alone as well.

For the destructive variant recall that for the left-to-right direction p never wins with the mentioned strategy and for the right-to-left direction p always is the only Borda winner. If p is the chair's distinguished candidate, this is what we need to show that the reduction is correct in the destructive case. ❑

6. Conclusions and future work

We have summarized recent results on how Borda has been used in collective decision making, ranging from voting (our main focus) to other fields (allocating indivisible goods to agents and hedonic games). This wide range of applicability is quite astonishing, considering how simple and elegant Borda's rule is. We have also surveyed the most central models of strategic behavior in computational social choice (namely, the most common manipulation, control, and bribery scenarios in elections), a true success story within AI, and we have mentioned some of the most important complexity results for other voting rules alongside Borda.

Our main technical contribution is that we have solved the remaining open problems about the complexity of standard control scenarios in Borda elections (recall Table 1). This closes the most glaring gap regarding the computational complexity of manipulative attacks on Borda elections. In particular, complementing previous results, we have now shown that Borda is resistant to every standard type of constructive control, whereas it is vulnerable to most of the destructive control types. We have also identified two of the rare cases where the complexity of a control problem in the unique-winner model parts company from that in the nonunique-winner model. In addition, we have provided the first results on online control in sequential Borda elections. While we have shown vulnerability for some cases of online candidate control, we have established coNP-hardness lower bounds for all cases of online voter control.

As future work for control in Borda elections, we propose

- to provide a parameterized complexity analysis of the cases where resistance is known (a few cases in this direction were already studied by, e.g., Liu and Zhu [72] and Chen et al. [22]),
- to study weighted elections for the vulnerable cases (see, e.g., [43]), and
- to settle the missing cases of online control in Borda elections: all cases of online control by partition of candidates.

Another challenging task is to settle the complexity of control for all scoring rules, ideally by establishing dichotomy results in the style of Hemaspaandra et al. [57,63,64], Betzler and Dorn [12], and Baumeister and Rothe [10]. Establishing hardness in *typical* settings rather than merely worst-case hardness results is still a great challenge in manipulation, control, and bribery. Furthermore, structured domains such as single-peaked or single-crossing elections have been extensively studied recently also for Borda [14,20,46,99], and we propose to continue this line of research especially regarding our cases involving partitioning of candidates or voters.

More generally, we propose to keep looking for new applications of this vintage voting rule in other fields and domains. For example, Rey and Rothe [88] have recently studied the complexity of structural control scenarios (such as adding and deleting players) in weighted voting games. How can we model control for Borda-induced FEN-hedonic games [92]?

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This paper merges and extends some previous papers, and we thank the AAAI'17, ICTCS'17, COMSOC'18, and AAAI'19 reviewers for their helpful comments. This work was supported in part by DFG grants RO-1202/14-2, RO-1202/15-1, and RO-1202/21-1.

References

[1] H. Aziz, Y. Bachrach, E. Elkind, M. Paterson, False-name manipulations in weighted voting games, J. Artif. Intell. Res. 40 (2011) 57–93.
[2] H. Aziz, R. Savani, Hedonic games, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 356–376, chapter 15.
[3] Y. Bachrach, O. Lev, Y. Lewenberg, Y. Zick, Misrepresentation in district voting, in: Proceedings of the 25th International Joint Conference on Artificial Intelligence, AAAI Press/IJCAI, July 2016, pp. 81–87.
[4] J. Baldwin, The technique of the Nanson preferential majority system of election, Trans. Proc. R. Soc. Vic. 39 (1926) 42–52.
[5] J. Bartholdi III, J. Orlin, Single transferable vote resists strategic voting, Soc. Choice Welf. 8 (4) (1991) 341–354.
[6] J. Bartholdi III, C. Tovey, M. Trick, The computational difficulty of manipulating an election, Soc. Choice Welf. 6 (3) (1989) 227–241.
[7] J. Bartholdi III, C. Tovey, M. Trick, How hard is it to control an election?, Math. Comput. Model. 16 (8/9) (1992) 27–40.
[8] D. Baumeister, S. Bouveret, J. Lang, N. Nguyen, T. Nguyen, J. Rothe, A. Saffidine, Positional scoring-based allocation of indivisible goods, J. Auton. Agents Multi-Agent Syst. 31 (3) (2017) 628–655.
[9] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Computational aspects of approval voting, in: J. Laslier, R. Sanver (Eds.), Handbook on Approval Voting, Springer, 2010, pp. 199–251, chapter 10.
[10] D. Baumeister, J. Rothe, Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules, Inf. Process. Lett. 112 (5) (2012) 186–190.
[11] D. Baumeister, J. Rothe, Preference aggregation by voting, in: J. Rothe (Ed.), Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, in: Springer Texts in Business and Economics, Springer-Verlag, 2015, pp. 197–325, chapter 4.
[12] N. Betzler, B. Dorn, Towards a dichotomy for the possible winner problem in elections based on scoring rules, J. Comput. Syst. Sci. 76 (8) (2010) 812–836.
[13] N. Betzler, R. Niedermeier, G. Woeginger, Unweighted coalitional manipulation under the Borda rule is NP-hard, in: Proceedings of the 22nd International Joint Conference on Artificial Intelligence, AAAI Press/IJCAI, July 2011, pp. 55–60.
[14] N. Betzler, A. Slinko, J. Uhlmann, On the computation of fully proportional representation, J. Artif. Intell. Res. 47 (2013) 475–519.
[15] J. Borda, Mémoire sur les élections au scrutin, Histoire de L'Académie Royale des Sciences, Paris, 1781, English translation appears in the paper by de Grazia [29].
[16] S. Bouveret, Y. Chevaleyre, N. Maudet, Fair allocation of indivisible goods, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 284–310, chapter 12.
[17] S. Bouveret, U. Endriss, J. Lang, Fair division under ordinal preferences: computing envy-free allocations of indivisible goods, in: Proceedings of the 19th European Conference on Artificial Intelligence, IOS Press, Aug. 2010, pp. 387–392.
[18] S. Brams, P. Edelman, P. Fishburn, Fair division of indivisible items, Theory Decis. 55 (2) (2003) 147–180.
[19] S. Brams, D. King, Efficient fair division: help the worst off or avoid envy?, Ration. Soc. 17 (4) (2005) 387–421.
[20] F. Brandt, M. Brill, E. Hemaspaandra, L. Hemaspaandra, Bypassing combinatorial protections: polynomial-time algorithms for single-peaked electorates, J. Artif. Intell. Res. 53 (July 2015) 439–496.
[21] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, I. Schnoor, Approximability of manipulating elections, in: Proceedings of the 23rd AAAI Conference on Artificial Intelligence, AAAI Press, July 2008, pp. 44–49.

[22] J. Chen, P. Faliszewski, R. Niedermeier, N. Talmon, Elections with few voters: candidate control can be easy, in: Proceedings of the 29th AAAI Conference on Artificial Intelligence, AAAI Press, Jan. 2015, pp. 2045–2051.
[23] Y. Chevaleyre, U. Endriss, J. Lang, N. Maudet, A short introduction to computational social choice, in: Proceedings of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science, in: Lecture Notes in Computer Science, vol. 4362, Springer-Verlag, Jan. 2007, pp. 51–69.
[24] V. Conitzer, T. Sandholm, J. Lang, When are elections with few candidates hard to manipulate?, J. ACM 54 (3) (2007) 14.
[25] V. Conitzer, T. Walsh, Barriers to manipulation in voting, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 127–145, chapter 6.
[26] A. Copeland, A "reasonable" social welfare function, Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences University of Michigan, 1951.
[27] J. Davies, G. Katsirelos, N. Narodytska, T. Walsh, Complexity of and algorithms for Borda manipulation, in: Proceedings of the 25th AAAI Conference on Artificial Intelligence, AAAI Press, Aug. 2011, pp. 657–662.
[28] J. Davies, G. Katsirelos, N. Narodytska, T. Walsh, L. Xia, Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules, Artif. Intell. 217 (2014) 20–42.
[29] A. de Grazia, Mathematical deviation of an election system, Isis 44 (1–2) (1953) 41–51.
[30] D. Dimitrov, P. Borm, R. Hendrickx, S. Sung, Simple priorities and core stability in hedonic games, Soc. Choice Welf. 26 (2) (2006) 421–433.
[31] C. Dwork, R. Kumar, M. Naor, D. Sivakumar, Rank aggregation methods for the web, in: Proceedings of the 10th International World Wide Web Conference, ACM Press, 2001, pp. 613–622.
[32] E. Elkind, P. Faliszewski, Approximation algorithms for campaign management, in: Proceedings of the 6th International Workshop on Internet & Network Economics, in: Lecture Notes in Computer Science, vol. 6484, Springer-Verlag, Dec. 2010, pp. 473–482.
[33] E. Elkind, P. Faliszewski, A. Slinko, Swap bribery, in: Proceedings of the 2nd International Symposium on Algorithmic Game Theory, in: Lecture Notes in Computer Science, vol. 5814, Springer-Verlag, October 2009, pp. 299–310.
[34] E. Elkind, P. Faliszewski, A. Slinko, Cloning in elections: finding the possible winners, J. Artif. Intell. Res. 42 (2011) 529–573.
[35] E. Elkind, J. Rothe, Cooperative game theory, in: J. Rothe (Ed.), Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, in: Springer Texts in Business and Economics, Springer-Verlag, 2015, pp. 135–193, chapter 3.
[36] E. Ephrati, J. Rosenschein, A heuristic technique for multi-agent planning, Ann. Math. Artif. Intell. 20 (1–4) (1997) 13–67.
[37] G. Erdélyi, M. Fellows, J. Rothe, L. Schend, Control complexity in Bucklin and fallback voting: a theoretical analysis, J. Comput. Syst. Sci. 81 (4) (2015) 632–660.
[38] G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, More natural models of electoral control by partition, in: Proceedings of the 4th International Conference on Algorithmic Decision Theory, in: Lecture Notes in Artificial Intelligence, vol. 9346, Springer-Verlag, September 2015, pp. 396–413.
[39] G. Erdélyi, M. Nowak, J. Rothe, Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control, Math. Log. Q. 55 (4) (2009) 425–443.
[40] G. Erdélyi, C. Reger, 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, IFAAMAS, May 2019, pp. 846–854.
[41] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, How hard is bribery in elections?, J. Artif. Intell. Res. 35 (2009) 485–532.
[42] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, Using complexity to protect elections, Commun. ACM 53 (11) (2010) 74–82.
[43] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, Weighted electoral control, J. Artif. Intell. Res. 52 (2015) 507–542.
[44] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Llull and Copeland voting computationally resist bribery and constructive control, J. Artif. Intell. Res. 35 (2009) 275–341.
[45] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, A richer understanding of the complexity of election systems, in: S. Ravi, S. Shukla (Eds.), Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz, Springer, 2009, pp. 375–406, chapter 14.
[46] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, The shield that never was: societies with single-peaked preferences are more open to manipulation and control, Inf. Comput. 209 (2) (2011) 89–107.
[47] P. Faliszewski, E. Hemaspaandra, H. Schnoor, Copeland voting: ties matter, in: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, May 2008, pp. 983–990.
[48] P. Faliszewski, A. Procaccia, AI's war on manipulation: are we winning?, AI Mag. 31 (4) (2010) 53–64.
[49] P. Faliszewski, Y. Reisch, J. Rothe, L. Schend, Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting, J. Auton. Agents Multi-Agent Syst. 29 (6) (2015) 1091–1124.
[50] P. Faliszewski, J. Rothe, Control and bribery in voting, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 146–168, chapter 7.
[51] M. Garey, D. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 1979.
[52] S. Ghosh, M. Mundhe, K. Hernandez, S. Sen, Voting for movies: the anatomy of recommender systems, in: Proceedings of the 3rd Annual Conference on Autonomous Agents, ACM Press, 1999, pp. 434–435.
[53] A. Gibbard, Manipulation of voting schemes: a general result, Econometrica 41 (4) (1973) 587–601.
[54] T. Gonzalez, Clustering to minimize the maximum intercluster distance, Theor. Comput. Sci. 38 (1985) 293–306.
[55] G. Hägele, F. Pukelsheim, The electoral writings of Ramon Llull, Stud. Lulliana 41 (97) (2001) 3–38.
[56] T. Haynes, S. Sen, N. Arora, R. Nadella, An automated meeting scheduling system that utilizes user preferences, in: Proceedings of the 1st International Conference on Autonomous Agents, ACM Press, 1997, pp. 308–315.
[57] E. Hemaspaandra, L. Hemaspaandra, Dichotomy for voting systems, J. Comput. Syst. Sci. 73 (1) (2007) 73–83.
[58] E. Hemaspaandra, L. Hemaspaandra, C. Menton, Search versus decision for election manipulation problems, ACM Trans. Comput. Theory 12 (1) (2020) 3:1–3:42.
[59] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Anyone but him: the complexity of precluding an alternative, Artif. Intell. 171 (5–6) (2007) 255–285.
[60] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, The complexity of online manipulation of sequential elections, J. Comput. Syst. Sci. 80 (4) (2014) 697–710.
[61] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, The complexity of controlling candidate-sequential elections, Theor. Comput. Sci. 678 (2017) 14–21.
[62] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, The complexity of online voter control in sequential elections, J. Auton. Agents Multi-Agent Syst. 31 (5) (2017) 1055–1076.
[63] E. Hemaspaandra, L. Hemaspaandra, H. Schnoor, A control dichotomy for pure scoring rules, in: Proceedings of the 28th AAAI Conference on Artificial Intelligence, AAAI Press, July 2014, pp. 712–720.
[64] E. Hemaspaandra, H. Schnoor, Dichotomy for pure scoring rules under manipulative electoral actions, in: Proceedings of the 22nd European Conference on Artificial Intelligence, IOS Press, August/September 2016, pp. 1071–1079.
[65] J. Kelly, Strategy-proofness and social choice functions without single-valuedness, Econometrica 45 (2) (1977) 439–446.
[66] K. Konczak, J. Lang, Voting procedures with incomplete preferences, in: Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling, July/August 2005, pp. 124–129.
[67] B. Kuckuck, J. Rothe, Sequential allocation rules are separable: refuting a conjecture on scoring-based allocation of indivisible goods, in: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, July 2018, pp. 650–658.

[68] B. Kuckuck, J. Rothe, Duplication monotonicity in the allocation of indivisible goods, AI Commun. 32 (4) (2019) 253–270.
[69] J. Lang, A. Rey, J. Rothe, H. Schadrack, L. Schend, Representing and solving hedonic games with ordinal preferences and thresholds, in: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, May 2015, pp. 1229–1237.
[70] J. Lang, J. Rothe, Fair division of indivisible goods, in: J. Rothe (Ed.), Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division, in: Springer Texts in Business and Economics, Springer-Verlag, 2015, pp. 493–550, chapter 8.
[71] Y. Lewenberg, O. Lev, Divide and conquer: using geographic manipulation to win district-based elections, in: U. Grandi, J. Rosenschein (Eds.), Proceedings of the 6th International Workshop on Computational Social Choice, Toulouse, France, June 2016.
[72] H. Liu, D. Zhu, Parameterized complexity of control by voter selection in maximin, Copeland, Borda, Bucklin, and approval election systems, Theor. Comput. Sci. 498 (2013) 115–123.
[73] A. Loreggia, N. Narodytska, F. Rossi, B. Venable, T. Walsh, Controlling elections by replacing candidates or votes (extended abstract), in: Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, May 2015, pp. 1737–1738.
[74] C. Maushagen, M. Neveling, J. Rothe, A. Selker, Complexity of shift bribery in iterative elections, in: Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, July 2018, pp. 1567–1575.
[75] C. Maushagen, J. Rothe, Complexity of control by partitioning veto elections and of control by adding candidates to plurality elections, Ann. Math. Artif. Intell. 82 (4) (2018) 219–244.
[76] C. Menton, Normalized range voting broadly resists control, Theory Comput. Syst. 53 (4) (2013) 507–531.
[77] H. Moulin, Condorcet's principle implies the no show paradox, J. Econ. Theory 45 (1) (1988) 53–64.
[78] E. Nanson, Methods of election, Trans. Proc. R. Soc. Vic. 19 (1882) 197–240.
[79] M. Neveling, J. Rothe, Closing the gap of control complexity in Borda elections: solving ten open cases, in: Proceedings of the 18th Italian Conference on Theoretical Computer Science, vol. 1949, Sept. 2017, pp. 138–149, CEUR-WS.org.
[80] M. Neveling, J. Rothe, Solving seven open problems of offline and online control in Borda elections, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence, AAAI Press, Feb. 2017, pp. 3029–3035.
[81] N. Nguyen, D. Baumeister, J. Rothe, Strategy-proofness of scoring allocation correspondences for indivisible goods, Soc. Choice Welf. 50 (1) (2018) 101–122.
[82] K. Oflazer, G. Tür, Morphological disambiguation by voting constraints, in: Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics, ACL/Morgan Kaufmann, 1997, pp. 222–229.
[83] C. Papadimitriou, Computational Complexity, second edition, Addison-Wesley, 1995.
[84] D. Parkes, L. Xia, A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs, in: Proceedings of the 26th AAAI Conference on Artificial Intelligence, AAAI Press, July 2012, pp. 1429–1435.
[85] D. Pennock, E. Horvitz, C. Giles, Social choice theory and recommender systems: analysis of the axiomatic foundations of collaborative filtering, in: Proceedings of the 17th National Conference on Artificial Intelligence, AAAI Press, July/August 2000, pp. 729–734.
[86] C. Puppe, A. Tasnádi, Optimal redistricting under geographical constraints: why "pack and crack" does not work, Econ. Lett. 105 (1) (2009) 93–96.
[87] A. Rey, J. Rothe, False-name manipulation in weighted voting games is hard for probabilistic polynomial time, J. Artif. Intell. Res. 50 (2014) 573–601.
[88] A. Rey, J. Rothe, Structural control in weighted voting games, B. E. J. Theor. Econ. 18 (2) (2018) 1–15.
[89] M. Risse, Why the count de Borda cannot beat the Marquis de Condorcet, Soc. Choice Welf. 25 (2005) 95–113.
[90] J. Rothe, Complexity Theory and Cryptology. An Introduction to Cryptocomplexity, EATCS Texts in Theoretical Computer Science, Springer-Verlag, 2005.
[91] J. Rothe, Borda count in collective decision making: a summary of recent results, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, AAAI Press, January/February 2019, pp. 9830–9836.
[92] J. Rothe, H. Schadrack, L. Schend, Borda-induced hedonic games with friends, enemies, and neutral players, Math. Soc. Sci. 96 (2018) 21–36.
[93] J. Rothe, L. Schend, Challenges to complexity shields that are supposed to protect elections against manipulation and control: a survey, Ann. Math. Artif. Intell. 68 (1–3) (2013) 161–193.
[94] N. Russel, Complexity of control of Borda count elections, Master's thesis, Rochester Institute of Technology, 2007.
[95] D. Saari, Capturing the 'Will of the people', Ethics 113 (2003) 333–334.
[96] D. Saari, Which is better: the Condorcet or Borda winner? Soc. Choice Welf. 27 (1) (2006) 107–129.
[97] M. Satterthwaite, Strategy-proofness and Arrow's conditions: existence and correspondence theorems for voting procedures and social welfare functions, J. Econ. Theory 10 (2) (1975) 187–217.
[98] G. Sigletos, G. Paliouras, C. Spyropoulos, M. Hatzopoulos, Combining information extraction systems using voting and stacked generalization, J. Mach. Learn. Res. 6 (2005) 1751–1782.
[99] Y. Yang, On the complexity of Borda control in single-peaked elections, in: Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems, IFAAMAS, May 2017, pp. 1178–1186.
[100] W. Yu, H. Hoogeveen, J. Lenstra, Minimizing makespan in a two-machine flow shop with delays and unit-time operations is NP-hard, J. Sched. 7 (5) (2004) 333–348.
[101] M. Zuckerman, A. Procaccia, J. Rosenschein, Algorithms for the coalitional manipulation problem, Artif. Intell. 173 (2) (2009) 392–412.
[102] W. Zwicker, Introduction to the theory of voting, in: F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia (Eds.), Handbook of Computational Social Choice, Cambridge University Press, 2016, pp. 23–56, chapter 2.

CHAPTER 4

TOWARDS COMPLETING THE PUZZLE: COMPLEXITY OF CONTROL BY REPLACING, ADDING, AND DELETING CANDIDATES OR VOTERS

4.1 Summary

In this chapter we study various open problems regarding electoral control, therefore taking a step towards completing the puzzle of the complexity of electoral control problems for the most important voting rules. In particular, we initiate and complete the study of the standard control cases for plurality with runoff and veto with runoff.

We also study another special case of $\mathcal{E}$-CONSTRUCTIVE-MULTIMODE-CONTROL which models electoral control by replacing candidates or voters that was introduced by Loreggia et al. [102]. For replacement control the election chair may alter the set of candidates or set of voters while keeping the size of both sets the same as in the original election. For example, if a candidate is removed from the election, one candidate from the set of unregistered candidates must be added subsequently, thus replacing the removed candidate with the added candidate. To obtain the corresponding control problems we restrict $\mathcal{E}$-CONSTRUCTIVE-MULTIMODE-CONTROL as follows.

- For $\mathcal{E}$-CONSTRUCTIVE-CONTROL-BY-REPLACING-VOTERS we set $\ell_{AV} = \ell_{DV}$, $\ell_{AC} = \ell_{DC} = 0$, and $D = \emptyset$; and require in the question that $|V'| = |W'|$.
- For $\mathcal{E}$-CONSTRUCTIVE-CONTROL-BY-REPLACING-CANDIDATES we set $\ell_{AC} = \ell_{DC}$, $\ell_{AV} = \ell_{DV} = 0$, and $W = \emptyset$; and require in the question that $|C'| = |D'|$.

The complexity of replacement control problems are studied for Copeland$^{\alpha}$, maximin, k-veto, plurality/veto with runoff, Condorcet, fallback, and (normalized) range voting. We find that the complexity of replacement control always matches the complexity of the corresponding control by adding or deleting problem with the highest complexity. For example, plurality with runoff is in P for constructive control by adding candidates, constructive control by deleting candidates, and constructive control by replacing candidates. If for a voting rule the complexity of control by adding and control by deleting candidates or voters differs, then the complexity of control by replacing candidates or voters matches the higher complexity as is the case of maximin and constructive candidate control. It was shown by Loreggia et al. [102] that this is not necessarily the case. Interestingly, Condorcet and range voting are both immune against (constructive and destructive) control by adding candidates but in combination with control by deleting candidates they become susceptible to control.

4.2 Publication – Erdélyi, Neveling, Reger, Rothe, Yang and Zorn [58]

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):1–48, 2021.

Preliminary versions of this paper were published in the proceedings of the *18th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS'19, see [60]) and of the *15th International Computer Science Symposium in Russia* (CSR'20, see [121]).

4.3 Personal Contribution

The writing was done jointly with my coauthors. Theorems 7, 8, and 20, and Theorems 22 – 25 are to be attributed to my contribution. Theorem 19 was done jointly with my coauthors.

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

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 α, 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*) τ maps each election (C, V) to a subset $W \subseteq C$ of the candidates, called the *τ winners* (or simply the *winners* if τ 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 α 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 ℓ 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 τ-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	$\lvert V'\rvert = \lvert W'\rvert, \ell_{AV} = \ell_{DV}, \ell_{AC} = \ell_{DC} = 0, D = \emptyset$
Replacing candidates	$\lvert C'\rvert = \lvert D'\rvert, \ell_{AC} = \ell_{DC}, \ell_{AV} = \ell_{DV} = 0, W = \emptyset$

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 τ.

τ-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 ℓ_{AV}, ℓ_{DV}, ℓ_{AC}, and ℓ_{DC}, with $\ell_{AV} \le \lvert W\rvert$, $\ell_{DV} \le \lvert V\rvert$, $\ell_{AC} \le \lvert D\rvert$, and $\ell_{DC} \le \lvert C\rvert$.

Question: Are there $V' \subseteq V$, $W' \subseteq W$, $C' \subseteq C \setminus \{c\}$, and $D' \subseteq D$ such that $\lvert V'\rvert \le \ell_{DV}$, $\lvert W'\rvert \le \ell_{AV}$, $\lvert C'\rvert \le \ell_{DC}$, $\lvert D'\rvert \le \ell_{AC}$, and c is a τ 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 τ-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 τ 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 ℓ to denote the integer(s) in the input that is not necessarily required to be 0. For example, when considering CCRV, we use ℓ to denote $\ell_{AV} = \ell_{DV}$.

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 κ 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_α and b_β 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 τ be a voting rule satisfying IBC. Then τ-CCRC is* NP*-hard if τ-CCDC is* NP*-hard, and τ-DCRC is* NP*-hard if τ-DCDC is* NP*-hard.*

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 α with $0 \le \alpha \le 1$, we have the following result.

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

However, for each rational α 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 α 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 ℓ' with $1 \le \ell' \le \ell$ so that d beats c by replacing ℓ' candidates. For a pair (d, ℓ'), 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 ℓ' 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 ℓ' 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* α *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 α 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 κ (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 2κ registered votes, among which κ of them are of the form $c\ c'\ \overrightarrow{C \setminus \{c, c'\}}$ and the other κ 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 κ 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 κ 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 κ votes. This means that replacing at most κ 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 κ 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 κ 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 κ 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 κ 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	–	2κ	2κ	2κ
d	$3\kappa + 1$	–	$4\kappa + 1$	$3\kappa + 1$
$u' \in U$	$3\kappa + 1$	κ	$*$	$\leq \kappa$

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 $\neq$ 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 τ be an unanimous voting rule that satisfies IBC. If τ-CCAC is* NP-*hard, then τ-CCRC is* NP-*hard.*

Due to this lemma and the facts that (1) maximin is unanimous; (2) maximin satisfies IBC; and (3) 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 κ votes of the form c U d; and

- there are κ 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 $S \in \mathcal{S}$, 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 κ 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 κ voters if and only if $\mathcal{S}$ contains an exact set cover of U.

($\Rightarrow$) Assume that U admits an exact set cover $\mathcal{S}' \subseteq \mathcal{S}$. Let $W' = \{v(S) \mid S \in \mathcal{S}'\}$ be the set of the unregistered votes corresponding to this exact set cover. Clearly, $|W'| = |\mathcal{S}'| = \kappa$. Let V' be a multiset of κ 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 $\mathcal{S}'$ 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 2κ 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 κ with respect to V, and we are allowed to replace at most κ votes, the maximin score of u in E' can be at most 2κ. 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 $\mathcal{S}' = \{S \in \mathcal{S} \mid v(S) \in W'\}$. 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 κ 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 κ 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 κ 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 $\mathcal{S}'$ 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 $\mathcal{S}$ contains u. In light of the above construction of the unregistered votes, all the κ votes in W' rank this particular candidate u above c, resulting in the maximin score of c in $\hat{E}$ being at most 2κ, 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 3κ 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 κ registered votes of the form $d\ U\ c$ with κ unregistered votes corresponding to an exact set cover of U, the maximin scores of c and d are, respectively, $2\kappa + 1$ and 2κ, 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 κ 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 2κ. 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 κ 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 τ-Exact-Destructive-Control-by-Adding-and-Deleting-Candidates, denoted by τ-EDCAC+DC, where τ is a voting rule. In particular, this problem is a variant of τ-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* ℓ_{DC} candidates and adds *exactly* ℓ_{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 ℓ_{AC} candidates from D and deletes exactly ℓ_{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 $C_1 = \{c'' \in C \setminus \{c, c', d\} : N(c', c'') \leq k\}$. 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 ℓ_{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 $k \geq 2$. We complement these results by showing that 2-Veto-CCRV is polynomial-time solvable and k-Veto-CCRV is NP-complete for $k \geq 3$, achieving a dichotomy result for 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 $(C, V \cup W)$, ℓ, 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: $|V^c| \leq \min(\ell, |W| - |W^c|)$.

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 ℓ', 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 ℓ' 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 ℓ voters in V vetoing c with ℓ 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 ℓ 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 ℓ 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 $(U, \mathscr{S})$ of RX3C, where $|U| = |\mathscr{S}| = 3\kappa$, 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 κ voters, i.e., $\ell = \kappa$.

We claim that c can be made a 3-veto winner by replacing at most κ 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 κ votes corresponding to $\mathscr{S}'$ from W with κ voters in V vetoing c, c has $(n + \kappa) - \kappa = n$ vetoes, every $u \in U$ has $(n - 1) + 1 = n$ vetoes, and each 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 ℓ 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 κ, 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 κ 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 κ 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 $U = \{u_1, \ldots, u_s\}$, $s \geq 1$, $\mathscr{S} = \{S_1, \ldots, S_t\}$, $t \geq 1$, and integer κ, $1 \leq \kappa < s$ (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

$$\begin{aligned} C' &= \{c'_1, \ldots, c'_{k-1}\}, \\ X &= \{x_1, \ldots, x_{k-1}\}, \text{ and} \\ Y &= \{y_1, \ldots, y_\kappa\}, \end{aligned}$$

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 κ candidates from C with candidates from U.

($\Rightarrow$) Assume there is a hitting set $U' \subseteq U$ of $\mathscr{S}$ of size κ (since $\kappa < s$, if U' is a hitting set of size less than κ, 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 κ candidates. Since the κ 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 κ 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 κ 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 κ added candidates from U need to be a hitting set of $\mathscr{S}$. Also note that with the κ added candidates from U, c also ties d (who lost κ 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 κ, $1 \leq \kappa \leq s$.

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 κ candidates from C with candidates from U.

($\Rightarrow$) Assume there is a hitting set $U' \subseteq U$ of $\mathscr{S}$ of size κ (since $\kappa < s$, if U' is a hitting set of size less than κ, 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 κ candidates. We first argue why we must remove all κ 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 κ candidates from Y need to be removed, for otherwise c does not gain any vetoes. Then κ 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 κ 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].

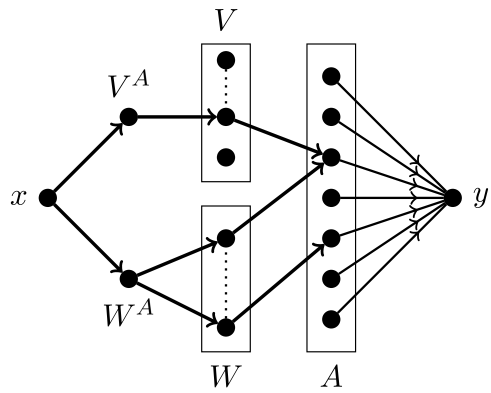
Our results in this section are summarized in Table 7.

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, τ-Exact Constructive Control by Adding and Deleting Voters, denoted by τ-ECCAV+DV, is polynomial-time solvable, where τ is either plurality with runoff or veto with runoff. In this exact variant of τ-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 τ is PRun, and be the number of voters in V vetoing d if τ 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* $\tau \in \{\text{PRun}, \text{VRun}\}$, τ *-ECCAV+DV is in* P.

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

$\tau =$ **PRun.** Our algorithm tries to find a candidate $d \in C \setminus \{c\}$ and four nonnegative integers ℓ^c_{AV}, ℓ^d_{AV}, ℓ^c_{DV}, and ℓ^d_{DV} such that $\ell^c_X + \ell^d_X \le \ell_X$ for $X \in \{\text{AV}, \text{DV}\}$. This candidate d is supposed to be the one who competes with c in the runoff stage. Moreover, ℓ^c_{AV} (respectively, ℓ^d_{AV}) is supposed to be the number of voters added from W that approve c (respectively, d), and ℓ^c_{DV} (respectively, ℓ^d_{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 ℓ_{AV} votes from W of which ℓ^c_{AV} (respectively, ℓ^d_{AV}) approve c (respectively, d), and delete exactly ℓ_{DV} votes from V of which ℓ^c_{DV} (respectively, ℓ^d_{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 $X \in \{\text{AV}, \text{DV}\}$, 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 ℓ^A_{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 ℓ^A_{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 $X \in \{\mathrm{AV}, \mathrm{DV}\}$, ℓ^c_{X} and ℓ^d_{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 $\mathrm{Y} \in \{\mathrm{CCAV}, \mathrm{CCDV}, \mathrm{CCRV}, \mathrm{DCAV}, \mathrm{DCDV}, \mathrm{DCRV}\}$ and for each $\mathrm{X} \in \{\mathrm{PRun}, \mathrm{VRun}\}$, 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* $\mathrm{Y} \in \{\mathrm{CCAV}, \mathrm{CCDV}, \mathrm{CCRV}, \mathrm{DCAV}, \mathrm{DCDV}, \mathrm{DCRV}\}$, *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 κ 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 κ 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 run-off 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 κ 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 κ 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 $\mathscr{S}' \subseteq \mathscr{S}$ of U. After adding the candidates in $\mathscr{S}'$, candidate q has one veto, every $S \in \mathscr{S}'$ has at least $6\kappa + 11$ vetoes, every $u \in U$ has $6\kappa + 3 - 2 = 6\kappa + 1$ vetoes, and c has 6κ 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 $\mathscr{S}' \subseteq \mathscr{S}$ of at most κ unregistered candidates to make c a winner under veto with runoff. Observe first that $\mathscr{S}'$ must contain exactly κ candidates, since otherwise c would have at least $6\kappa + 3$ vetoes, while at least one candidate in U would have at most $6\kappa + 3 - 2 = 6\kappa + 1$ vetoes. Hence, this candidate in U and q would be the two candidates going to the runoff stage. Then, from $|\mathscr{S}'| = \kappa$, it follows that c has 6κ vetoes after adding candidates in $\mathscr{S}'$. If $\mathscr{S}'$ is not an exact 3-set cover, there must be a candidate $u \in U$ occurring in at least two subsets of $\mathscr{S}'$. 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 $\mathscr{S}'$ 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 $(U, \mathscr{S})$ with $|U| = |\mathscr{S}| = 3\kappa$ 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 $U = \{u_1, u_2, \dots, u_{3\kappa}\}$. Without loss of generality, assume that $\kappa \geq 4$.

PRun-CCDC. From $(U, \mathscr{S})$, we create the following instance of PRun-CCDC. Let $C = \{c, q\} \cup U \cup \mathscr{S}$ 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 2κ votes of the form $q\ u_1\ u_2\ \dots\ u_{3\kappa}\ \mathscr{S}\ c$.
- We create $\kappa + 1$ votes of the form $q\ u_{3\kappa}\ u_{3\kappa-1}\ \dots\ u_1\ \mathscr{S}\ c$.
- For each $u \in U$, we create $3\kappa - 3$ votes of the form $u\ U \setminus \{u\}\ \mathscr{S}\ c\ q$.
- For each $S \in \mathscr{S}$, we create three votes of the form $S\ c\ C \setminus (S \cup \{c, q\})\ q$.
- For each $S = \{u_x, u_y, u_z\} \in \mathscr{S}$, 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 $\mathscr{S}' \subseteq \mathscr{S}$. After deleting the candidates in $\mathscr{S}'$, q has $2\kappa + \kappa + 1 = 3\kappa + 1$ approvals, c has 3κ approvals, every remaining $S \in \mathscr{S} \setminus \mathscr{S}'$ 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 κ 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 κ 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 3κ 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 κ 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 3κ 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 κ 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 κ 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 3κ. 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 κ 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 κ 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 κ 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 κ 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 κ registered candidates and rank them before all the other candidates in all votes. More importantly, we rank all the κ 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 κ unregistered candidates, and rank them directly after q in all votes (i.e., q and these κ candidates are ranked consecutively in all votes with q being the first one among them). The relative order among these κ 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 $U = \{u_1, \dots, u_{3\kappa}\}$, $\kappa \geq 2$ (which may be assumed, as RX3C is trivially solvable when $\kappa = 1$), and $\mathscr{S} = \{S_1, \dots, S_{3\kappa}\}$. 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 κ votes from V with votes from W.

($\Rightarrow$) Assume there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of U. We remove κ 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 κ 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* κ 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 κ removed votes, c's deficit to every other candidate is now decreased to $\kappa - 3$. However, none of the κ 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 κ new votes to rank each of the 3κ 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 ℓ. 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 ℓ votes from V preferring c to d and let $W' \subseteq W$ contain at most ℓ 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 ℓ. 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 ℓ, 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 ℓ 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 κ votes from V with as many votes from W.

($\Rightarrow$) Suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of U. Remove κ 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 κ votes from V with as many votes from W. Since c has no approvals in (C, V) and we can only add at most κ 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 κ 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 κ 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 κ 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 ℓ. 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 ℓ 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 ℓ 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 ith 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 ℓ 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 ℓ 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 ℓ, and we ask whether c can be made a fallback winner by deleting at most ℓ 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 ℓ 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 $D = \{d_1, \dots, d_\ell\}$, $D \cap C = \emptyset$, 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 ℓ 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 ℓ candidates from C if and only if c can be made a fallback winner of $(C \cup D, V')$ by deleting up to ℓ 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 ℓ. 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 ℓ 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 ℓ 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 $E = \{e_1, \ldots, e_\kappa\}$ 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 κ 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 ℓ 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 ℓ 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 5

THE COMPLEXITY OF CLONING CANDIDATES IN MULTIWINNER ELECTIONS

5.1 Summary

We study how multiwinner elections can be tampered with by cloning candidates. For that we adapt the model for cloning candidates introduced by Elkind, Faliszewski, and Slinko [53] to multiwinner elections and define the following decision problems for a multiwinner voting rule $\mathscr{R}$.

$\mathscr{R}$-POSSIBLE-CLONING-GC	
Input:	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.
Question:	Is there a cloning vector $K = (K_1,\dots,K_m)$ with $\sum_{c_i \in C} \rho_i(K_i) \le B$ such that p (or one of her clones) is in a winning committee under $\mathscr{R}$ in at least one cloned multiwinner election E_K resulting from E via K?

$\mathscr{R}$-NECESSARY-CLONING-GC is defined analogously but we require that the cloning vector makes p (or one of her clones) part of a winning committee in *all* (instead of "at least one") cloned multiwinner elections resulting from E via the cloning vector.

Just like Elkind, Faliszewski, and Slinko [53] we study three cost models ZERO-COST, UNIT-COST, and GENERAL-COST. The corresponding problems with ZERO-COST and UNIT-COST are denoted by replacing "GC" with "ZC" and "UC", respectively, in the problem names. Our model is more focused than the singlewinner variant of Elkind, Faliszewski, and Slinko [53] in that we do not take on a probabilistic viewpoint in regards to when we view a cloning action as successful and only capture the extreme points by viewing a cloning action as successful if the distinguished candidate (or one of her clones) is in a winning committee in *at least one* or in *all* cloned multiwinner elections resulting from a cloning action.

We study the complexity of $\mathscr{R}$-POSSIBLE-CLONING-$\{$GC,UC,ZC$\}$ and $\mathscr{R}$-NECESSARY-CLONING-$\{$GC,UC,ZC$\}$ for the multiwinner voting rules defined in Chapter 2. In order to have polynomial-time winner determination for STV we use lexicographic tie-breaking for ties between candidates and arbitrary tie-breaking for ties between voters. Even with those simple tie-breaking rules both decision problems are intractable for STV: Possible cloning is NP-hard for STV in all cost models and coNP-hard for necessary cloning in all cost models. SNTV is easy for possible cloning and trivial for necessary cloning. Surprisingly, possible cloning with ZERO-COST and UNIT-COST is easy for k-Borda while it is NP-hard for Bloc. All other cases for Bloc and k-Borda are NP-hard. The reduction that is used to show NP-hardness of k-Borda-NECESSARY-CLONING-ZC also holds for $k = 1$ (i.e., the singlewinner variant of k-Borda) and therefore solves a problem left open by Elkind, Faliszewski, and Slinko [53].

The Chamberlin–Courant voting rules that we study have NP-hard winner determination meaning the decision problems defined above are trivially NP-hard for all cost models as well so we investigate the parameterized complexity of the problems with the number of candidates and the number of voters as parameters.

5.2 Publication – Neveling and Rothe [119]

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, 2020.

5.3 Personal Contribution

The writing was done jointly with Jörg Rothe. Modeling and technical parts are to be attributed to my contribution.

The Complexity of Cloning Candidates in Multiwinner Elections

Marc Neveling and Jörg Rothe
Institut für Informatik,
Heinrich-Heine-Universität Düsseldorf, Germany
{marc.neveling,rothe}@hhu.de

ABSTRACT

We initiate the study of cloning in multiwinner elections, focusing on single-transferable vote (STV), single-nontransferable vote (SNTV), bloc, k-Borda, t-approval-CC, and Borda-CC. Transferring the model of cloning due to Elkind et al. [15] 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. 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

ACM Reference Format:
Marc Neveling and Jörg Rothe. 2020. The Complexity of Cloning Candidates in Multiwinner Elections. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

1 INTRODUCTION

A common thread in computational social choice—see, e.g., the books edited by Brandt et al. [6] and Rothe [32]—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., Conitzer and Walsh [12] and Baumeister and Rothe [4, Section 4.3.3]), electoral control (see, e.g., Faliszewski and Rothe [17] and Baumeister and Rothe [4, Section 4.3.4]), and bribery (see, e.g., Faliszewski and Rothe [17] and Baumeister and Rothe [4, 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 chapter by Faliszewski et al. [18]). 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 Lu and Boutilier [26], Elkind et al. [14], and Skowron et al. [34] for more detailed descriptions of the mentioned settings). In each of those settings, cloning candidates might be beneficial for a candidate to be voted into the resulting committee. For instance, in a parliament election the campaign manager of a party, whose candidates may look like clones to ignorant voters, might be inclined to nominate a strategically chosen number of candidates for the party. Another application of cloning in multiwinner elections are movie recommender systems [21] 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 [35] 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 [36] 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 [33]. 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 [10] to introduce false-name manipulation as cloning of *voters* instead of candidates.[1] Recently, the independence of clones property was studied for the single-winner variant of STV with top-truncated votes by Ayadi et al. [1].

The paper by far most closely related to our work is due to Elkind et al. [15] (see also their follow-up paper [16]). 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.

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 and Section 5 our conclusions and some open problems.

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

[1] False-name manipulation [2, 30] has also been studied in cooperative game theory and the related property of duplication monotonicity [3, 24] in fair division.

2 PRELIMINARIES

A multiwinner election $E = (C, V, k)$ is defined by a set of candidates $C = \{c_1, \ldots, c_m\}$, a list of votes $V = (v_1, \ldots, v_n)$, and a committee size k. Votes are 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 (e.g., 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* $a \in C$ is the number of votes in which a is ranked in the first t positions, and *a's Borda score* is the total number of points a scores in all votes of V, where a is rewarded with $m - i$ points whenever a is ranked in the ith position of a vote.

We consider the following multiwinner voting rules (with n voters and committee size k):

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. To break ties between candidates we will use a predefined lexicographic tie-breaking order and 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 arbitrarily.[2]

Single nontransferable vote (SNTV): Choose k candidates with highest 1-approval score.

Bloc: Choose k candidates with highest k-approval score.

***k*-Borda:** Choose k candidates with highest Borda score.

***t*-approval-CC** (where CC stands for "Chamberlin–Courant" [9]): 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).

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 [26, 29], though they are in FPT if parameterized by the number of candidates or voters [5].

For (C, V, k) a multiwinner election and candidates $c, d \in C$, let $score_{(C,V,k)}(c)$ denote the number of points c scores (according to t-approval or Borda which is always clear 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, $c\ \overrightarrow{S}\ b$ denotes the vote $c\ a\ d\ b$ and the vote $c\ \overleftarrow{S}\ b$ denotes $c\ d\ a\ b$.

[2]We cannot use "parallel-universe tie-breaking" [11] for STV since winner determination would then already be NP-hard.

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 $C = \{c_1, \ldots, c_m\}$ and $V = (v_1, \ldots, v_n)$. Let $K = (K_1, \ldots, K_m)$ with $K_i \geq 0$ be 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. [15] 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 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 $V' = (v'_1, \ldots, v'_n)$ with each $v'_i \in V'$ being a total order over C' that results from v_i by replacing 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 her 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 the distinguished candidate (or one of her 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 the distinguished candidate (or one of her 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 the distinguished candidate is part of *at least one* winning committee instead of *all* winning committees, which would be required in the *unique-winner model*. Furthermore, we consider the cost of cloning candidates. In the *general-cost (GK) model*, 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 $\rho_i(j) \leq \rho_i(j')$. $\rho_i(j)$ is the cost of cloning the ith candidate j times and replacing her in all votes with her 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 (UC) model* in which $\rho_i(j) = j - 1$ for all i and $j \geq 1$ (i.e., every additional clone costs one and there is a maximum number of additional clones), and a special case of the unit-cost model, the *zero-cost (ZK) model*, 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 in the first place.

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

Table 1: Overview of complexity results for various multiwinner voting rules

		Possible-Cloning			Necessary-Cloning		
voting rule	parameter	ZC	UC	GC	ZC	UC	GC
STV		NP-hard			coNP-hard		
SNTV		P			–		
Bloc		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

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 (see, e.g., the books by Papadimitriou [28] and Rothe [31]) and the parameterized branch (see, e.g., the books by Downey and Fellows [13] and Niedermeier [27]). 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 RESULTS

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.

4.1 STV

We show 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, even if* $k = 2$.

Proof. To prove this theorem, we will need the following observation and lemmas (proofs are omitted due to space constraints).

Observation 1. *Cloning a candidate does not change the plurality score of any other candidates or their clones.*

Lemma 4.2. *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.*

Lemma 4.3. *In an STV multiwinner election, a candidate in a winning committee that is always added last to its winning committees can be cloned without changing the outcome of the multiwinner election.*

To show NP-hardness of STV-Possible-Cloning-ZC, we now reduce from the well-known NP-hard problem X3C (see, e.g., Garey and Johnson [20]) to STV-Possible-Cloning-ZC. Let $(X, \mathcal{S})$ with $X = \{x_1, \dots, x_{3s}\}$ and $\mathcal{S} = \{S_1, \dots, S_{3s}\}$ be a given X3C instance and assume that every $x_i \in X$ appears in exactly three elements of $\mathcal{S}$ (that this restriction of X3C is still NP-complete was shown by Gonzalez [22]). We also assume that $s \geq 3$ is even, which can be achieved by duplicating the 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.

Table 2: List of votes V for the proof of Theorem 4.1

number	vote	for
$9\frac{s^2}{2} + 49\frac{s}{2} + 13$	$d\ e\ f\ p$	
1	$d\ S_i\ c\ p$	$1 \leq i \leq 3s$
$\frac{s}{2} + 1$	$S_i\ e\ p$	$1 \leq i \leq 3s$
$\frac{s}{2} + 2$	$S_i\ f\ p$	$1 \leq i \leq 3s$
$s + 5$	$x_i\ \mathcal{S}_{x_i}\ c\ p$	$1 \leq i \leq 3s$
$\frac{s}{2} + 2$	$b_i\ S_i\ e\ p$	$1 \leq i \leq 3s$
$\frac{s}{2} + 2$	$b_i\ S_i\ f\ p$	$1 \leq i \leq 3s$
$4s + 8$	$p\ c$	
$4s + 7$	$c\ p\ e$	
$4s + 4$	$e\ p\ c$	
$4s + 4$	$f\ p\ c$	

We will break ties according to the linear order $\overrightarrow{B}\ \overrightarrow{C \setminus B}$.

To complete the proof of Theorem 4.1, we will now show that $(X, \mathcal{S})$ is a yes-instance of X3C if and only if p can be made a 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. Due to space constraints, however, we will only sketch the proof of the implication from left to right (and will then prove the converse direction in full detail): From left to right, suppose there is an exact cover $\mathcal{S}'$. Clone d twice, and order them in such a way that the s votes of the form $d\ S_i\ p\ c$ for every $S_i \in \mathcal{S}'$ are not removed from the election when one clone of d is added to the winning committee. Then, p is later added to the winning committee.

For the converse direction, assume there is no exact cover. From Lemmas 4.2 and 4.3 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. Notice 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, 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. ❑

Note that, by Lemma 4.2 and Lemma 4.3, 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 [35].

The reduction above can be modified to show that constructive control by adding candidates—see [4, 17] 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 (the proof is omitted due to space constraints). Notice that in contrast to the Possible-Cloning variant we cannot fix k here.

Theorem 4.4. *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 $(X, \mathcal{S})$ be a given X3C instance, where $X = \{x_1, \ldots, x_{3s}\}$ and $\mathcal{S} = \{S_1, \ldots, S_{3s}\}$. Again, assume that every $x_i \in X$ appears in exactly three elements of $\mathcal{S}$ (recall the result by Gonzalez [22]). 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.

Table 3: List of votes V for the proof of Theorem 4.4

number	vote	for
$25s + 2$	$d_j\ r_1\ r_2\ S_1\ p$	$1 \leq j \leq s$
1	$d_j\ r_1\ r_2\ S_i\ p$	$1 \leq j \leq s$ and $1 \leq i \leq 3s$
2	$S_i\ e_i\ p$	$1 \leq i \leq 3s$
3	$b_i\ S_i\ f_i\ p$	$1 \leq i \leq 3s$
4	$x_i\ \mathcal{S}_{x_i}\ r_1\ p$	$1 \leq i \leq 3s$
2	p	
1	$r_1\ p$	
1	$r_2\ r_1\ p$	
5	$e_i\ p$	$1 \leq i \leq 3s$
4	$f_i\ p$	$1 \leq i \leq 3s$

We will use the linear order $\overrightarrow{X}\ p\ r_1\ r_2\ \overrightarrow{B}\ \overrightarrow{\mathcal{S}}\ \overrightarrow{D}\ \overrightarrow{F}\ \overrightarrow{E}$ to break ties. 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 4.4, 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. We now show that there is a cloning vector such that for every possible ordering of clones p is part of a winning committee. Consider the cloning vector in which every candidate from D is cloned twice and consider three cases on 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 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. Notice 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. From the then not removed vote of the form $d_j\ \cdots$ either some $S_i \in \mathcal{S}$ or p gains a point. 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 $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| \leq s$ 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	≥ 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	≥ 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 score 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. Due to space constraints, we omit the proof of the direction from right to left. ❑

4.2 SNTV

By modifying a proof of Elkind et al. [15], we obtain:

Theorem 4.5. *SNTV-Possible-Cloning-GC is in* P.

Necessary cloning for SNTV (in any cost model) is impossible if p is not already in a winning committee, since we can order the clones such that one of the clones is in front of all other clones of her original candidate in all votes. Therefore, all but one clone of a candidate have zero points and the one clone has the same number of points as its original candidate in the initial multiwinner election. Therefore, if a candidate was part of a winning committee, then one of her clones is in a winning committee as well.

4.3 Bloc Voting

For bloc voting, we have NP-hardness results both for possible and necessary cloning in the zero-cost model. We omit the proof for possible cloning due to space constraints and present that for necessary cloning in detail here.

Theorem 4.6. *Bloc-Possible-Cloning-ZC is* NP*-hard, even if* $k = 2$.

Theorem 4.7. *Bloc-Necessary-Cloning-ZC is* NP*-hard, even if* $k = 2$.

Proof. For a fixed $t \geq 2$, t-approval-Necessary-Cloning-ZC was shown to be NP-hard by Elkind et al. [15]. We will reduce 2-approval-Necessary-Cloning-ZC to Bloc-Necessary-Cloning-ZC. Let $((C, V), p)$ be an instance of 2-approval-Necessary-Cloning-ZC. Set the committee size to $k = 2$. Therefore, 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 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 $(C \cup D \cup \{w\}, V \cup V', 2)$, since the candidates from C do not gain additional points from V', all additional dummy candidates score only one point and w scores $|V| + 1$ points which is a higher score than p has. 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 is a candidate with a higher score than p. Therefore, p is always behind one candidate of C in $(C \cup D \cup \{w\}, V \cup V', 2)$ as well, since cloning w or 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 her, 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. [15] 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. When restricted to unit costs, we can show that it is easy to solve (the proof is omitted due to space constraints).

Theorem 4.8. *k-Borda-Possible-Cloning-UC is in* P.

On the other hand, in the zero-cost model the problem becomes NP-hard for k-Borda, even for size-1 committees.

Theorem 4.9. *k-Borda-Necessary-Cloning-ZC is* NP*-hard, 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 $\mathcal{S} = \{S_1, \dots, S_{3s}\}$ (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 votes:

(1) $7s + 1$ times a vote $a\ p\ \overrightarrow{X}\ \mathcal{S}\ d$ and a vote $\overleftarrow{X}\ a\ p\ \mathcal{S}\ d$.
(2) A vote $\overrightarrow{X}\ p\ \mathcal{S}\ a\ d$ and a 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 a vote $x_j\ S_i\ a\ p\ \overrightarrow{X \setminus \{x_j\}}\ \mathcal{S} \setminus \{S_i\}\ d$ and a 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 times a vote $a\ p\ \overrightarrow{X}\ \mathcal{S}\ d$ and a vote $p\ a\ \overleftarrow{X}\ \mathcal{S}\ d$.
(5) $9s$ times a vote $\overrightarrow{X}\ a\ p\ \mathcal{S}\ d$ and a vote $\overleftarrow{X}\ p\ a\ \mathcal{S}\ d$.
(6) For every $x_j \in X$, there are $2s + 4$ times a vote $\overleftarrow{X \setminus \{x_j\}}\ a\ p\ x_i\ d\ \mathcal{S}$ and a vote $x_i\ d\ p\ a\ \overrightarrow{X \setminus \{x_j\}}\ \mathcal{S}$.
(7) 8 times a vote $\overrightarrow{X}\ p\ a\ \mathcal{S}\ d$ and a vote $p\ \overleftarrow{X}\ a\ \mathcal{S}\ d$.
(8) 16 times a vote $\overrightarrow{X}\ p\ a\ \mathcal{S}\ d$ and a vote $a\ d\ p\ \overleftarrow{X}\ \mathcal{S}$.

We have the following point balances between p and the others:

$$
\begin{aligned}
dist_{(C,V,1)}(p, a) &= -(14s + 2) + (6s + 2) - 18s + 24s \\
&= -26s + 24s = -2s, \\
dist_{(C,V,1)}(p, x_i) &= -(7s + 1) - (3s + 1) - 18 + 3s(9s - 3) \\
&\quad -(9s - 13)(3s + 2) - (2s + 4) = 2, \\
dist_{(C,V,1)}(p, S_i) &> 6, \text{ and } dist_{(C,V,1)}(p, d) > 0.
\end{aligned}
$$

Lemma 4.10. *In the constructed election, if a candidate from $C \setminus \mathcal{S}$ is cloned more than once, p or all clones of p lose the election.*

Lemma 4.11. *In the constructed election, 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 $\mathcal{S} \setminus \{S_i\}$.

The proofs of Lemmas 4.10 and 4.11 are omitted due to space constraints. Equipped with these two lemmas, we now show 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 4.11 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 4.10 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. [15].

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 (the proof of which is omitted due to space constraints) will be helpful in the proofs of Theorems 4.13 and 4.14, the former of which is presented here while the latter is again omitted due to space constraints.

Lemma 4.12. *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.*

Theorem 4.13. *For a 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. [8] for t-approval-CC-Shift Bribery and using Lemma 4.12 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, iterate over all possible cloning vectors $(K_1, \ldots, K_m)$ 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 ith and jth 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 $\sum_{j \in [m'!]} S_{i,j} = n_i$ for every $i \in [m!]$ to ensure that the number of votes stays the same in the cloned election. Next, we introduce a constraint $\sum_{i \in [m!], j \in [m'!]} S_{i,j} \cdot \mathrm{feas}(i, j) = 0$ 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 $j \in [m'!]$ which describe the number of votes of type j in the cloned multiwinner election: $\sum_{i \in [m!]} S_{i,j} = N_j$. 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: $\sum_{i \in [m'!]} \omega(i, W) \cdot N_i > \sum_{i \in [m'!]} \omega(i, W') \cdot N_i$.

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 all cloning vectors have been iterated over 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 4.12 we only need to check cloning vectors in which every component is at most t. Additionally, $\mathrm{feas}(i, j)$ 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 Jr. [25] (which was improved by Kannan [23] and by Fredman and Tarjan [19]) that ILPs can be solved in FPT with respect to the number of integer variables as the parameter. ❑

Theorem 4.14. *For a fixed $t \geq 2$, t-approval-CC-Necessary-Cloning-GC is in* FPT *when parameterized by the number of voters.*

Next, we turn to t-approval-CC-Possible-Cloning-GC. We cannot use Lemma 4.12 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 4.15. 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. The proof of Theorem 4.16 is omitted here due to space constraints.

Theorem 4.16. *For a fixed $t \geq 2$, t-approval-CC-Possible-Cloning-GC is in* FPT *when parameterized by the number of voters.*

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 4.17. *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 $W_1, \ldots, W_f$, 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, $1 \le i \le f$? Multicolored-Independent-Set is $W[1]$-hard when parameterized by the number of colors [13].

Let $(G, f, (W_1, \ldots, W_f))$ 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 $\ell \ge 1$) 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, $1 \le i \le f$, denote by $\delta(i)$ the sum of degrees of vertices with color i, and let $\Delta = \sum_{1 \le i \le f} \delta(i)$.

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. For a color i, $1 \le i \le f$, let $W_i = \{v_1^{(i)}, \ldots, v_\ell^{(i)}\}$ and for a subset $X \subseteq V(G)$, let $G \setminus X$ be the graph G without vertices (and incident edges) of X. Define V to consist of these votes:

(1) For every color i, with $1 \le i \le f$, 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) 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:

$$\begin{aligned} dist(p, h_i) &= 2 + (f - 1)(2(E(G) + V(G)) + f + 2), \\ dist(p, e_j) &= 4 + 2(\ell - 1) + (f - 2)(2\ell + E(G) + 3) + \Delta. \end{aligned}$$

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 $B = f$. Regarding the price functions, for every $v \in V(G)$ let the cost of cloning v twice be one and cloning her more than twice be $B + 1$. Then let the cost of cloning any other candidate more than once be $B + 1$. Finally, let $k = 1$. It is easy to see that we will only need to worry about the scores of p, of candidates from H, and of candidates $E(G)$, 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 the candidate 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.

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. ❑

Since in the reduction above the ordering of clones did not matter, the following holds as well.

Corollary 4.18. *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 (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. Specifically, the parameterized complexity of possible cloning in t-approval-CC for #*candidates* (rather than #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 #voters and in all cost models for #candidates. Of course, there are many more multiwinner voting rules than those studied here (see the book chapter by Faliszewski et al. [18] for an overview), 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. [7] for Shift-Bribery, and to further explore the parameterized complexity for problems that are NP-hard.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their helpful comments. This work was supported in part by DFG grant RO-1202/14-2.

REFERENCES

[1] M. Ayadi, N. Ben Amor, J. Lang, and D. Peters. 2019. Single Transferable Vote: Incomplete Knowledge and Communication Issues. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 1288–1296.

[2] H. Aziz, Y. Bachrach, E. Elkind, and M. Paterson. 2011. False-Name Manipulations in Weighted Voting Games. *Journal of Artificial Intelligence Research* 40 (2011), 57–93.

[3] D. Baumeister, S. Bouveret, J. Lang, N. Nguyen, T. Nguyen, J. Rothe, and A. Saffidine. 2017. Positional Scoring-Based Allocation of Indivisible Goods. *Journal of Autonomous Agents and Multi-Agent Systems* 31, 3 (2017), 628–655.

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

[5] N. Betzler, A. Slinko, and J. Uhlmann. 2013. On the Computation of Fully Proportional Representation. *Journal of Artificial Intelligence Research* 47 (2013), 475–519.

[6] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice*. Cambridge University Press.

[7] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. 2016. Prices Matter for the Parameterized Complexity of Shift Bribery. *Information and Computation* 251 (2016), 140–164.

[8] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. 2016. Complexity of Shift Bribery in Committee Elections. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*. AAAI Press, 2452–2458.

[9] J. Chamberlin and P. Courant. 1983. Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule. *The American Political Science Review* 77, 3 (1983), 718–733.

[10] V. Conitzer. 2008. Anonymity-proof Voting Rules. In *Proceedings of the 4th International Workshop on Internet & Network Economics*. Springer-Verlag *Lecture Notes in Computer Science #5385*, 295–306.

[11] V. Conitzer, M. Rognlie, and L. Xia. 2009. Preference Functions That Score Rankings and Maximum Likelihood Estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*. IJCAI, 109–115.

[12] V. Conitzer and T. Walsh. 2016. Barriers to Manipulation in Voting. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia (Eds.). Cambridge University Press, 127–145.

[13] R. Downey and M. Fellows. 2013. *Parameterized Complexity* (2nd ed.). Springer-Verlag.

[14] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. 2017. Properties of Multiwinner Voting Rules. *Social Choice and Welfare* 48, 3 (2017), 599–632.

[15] E. Elkind, P. Faliszewski, and A. Slinko. 2011. Cloning in Elections: Finding the Possible Winners. *Journal of Artificial Intelligence Research* 42 (2011), 529–573.

[16] E. Elkind, P. Faliszewski, and A. Slinko. 2012. Clone Structures in Voters' Preferences. In *Proceedings of the 13th ACM Conference on Electronic Commerce*. ACM Press, 496–513.

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

[18] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. 2017. Multiwinner Voting: A New Challenge for Social Choice Theory. In *Trends in Computational Social Choice*, U. Endriss (Ed.). AI Access Foundation, 27–47.

[19] M. Fredman and R. Tarjan. 1987. Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms. *J. ACM* 34, 3 (1987), 596–615.

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

[21] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. 1999. Voting for Movies: The Anatomy of Recommender Systems. In *Proceedings of the 3rd Annual Conference on Autonomous Agents*. ACM Press, 434–435.

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

[23] R. Kannan. 1987. Minkowski's Convex Body Theorem and Integer Programming. *Mathematics of Operations Research* 12, 3 (1987), 415–440.

[24] B. Kuckuck and J. Rothe. 2019. Duplication Monotonicity in the Allocation of Indivisible Goods. *AI Communications* 32, 4 (2019), 253–270.

[25] H. Lenstra Jr. 1983. Integer Programming with a Fixed Number of Variables. *Mathematics of Operations Research* 8, 4 (1983), 538–548.

[26] T. Lu and C. Boutilier. 2011. Budgeted Social Choice: From Consensus to Personalized Decision Making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. AAAI Press/IJCAI, 280–286.

[27] R. Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.

[28] C. Papadimitriou. 1995. *Computational Complexity* (second ed.). Addison-Wesley.

[29] A. Procaccia, J. Rosenschein, and A. Zohar. 2008. On the Complexity of Achieving Proportional Representation. *Social Choice and Welfare* 30, 3 (2008), 353–362.

[30] A. Rey and J. Rothe. 2014. False-Name Manipulation in Weighted Voting Games is Hard for Probabilistic Polynomial Time. *Journal of Artificial Intelligence Research* 50 (2014), 573–601.

[31] J. Rothe. 2005. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. Springer-Verlag.

[32] J. Rothe (Ed.). 2015. *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer-Verlag.

[33] M. Schulze. 2011. A New Monotonic, Clone-Independent, Reversal Symmetric, and Condorcet-Consistent Single-Winner Election Method. *Social Choice and Welfare* 36, 2 (2011), 267–303.

[34] P. Skowron, P. Faliszewski, and J. Lang. 2016. Finding a Collective Set of Items: From Proportional Multirepresentation to Group Recommendation. *Artificial Intelligence* 241 (2016), 191–216.

[35] N. Tideman. 1987. Independence of Clones as a Criterion for Voting Rules. *Social Choice and Welfare* 4, 3 (1987), 185–206.

[36] T. Zavist and N. Tideman. 1989. Complete Independence of Clones in the Ranked Pairs Rule. *Social Choice and Welfare* 6, 2 (1989), 167–173.

CHAPTER 6

COMPLEXITY OF SHIFT BRIBERY FOR ITERATIVE VOTING RULES

6.1 Summary

We extend the study of shift bribery, introduced by Faliszewski et al. [66] and formally defined and studied by Elkind, Faliszewski, and Slinko [52], to iterative voting rules that elect the winner(s) of an election in multiple rounds.

For this chapter we will give an alternative but equivalent definition of shift bribery as it was defined in Chapter 2 that is more convenient to handle.

$\mathscr{E}$-CONSTRUCTIVE-SHIFT-BRIBERY	
Input:	An election (C,V) with n voters, a designated candidate $p \in C$, a budget B, and a list of price functions $\rho = (\rho_1, \dots, \rho_n)$.
Question:	Is it possible to make p the unique $\mathscr{E}$-winner of the election by shifting p in the votes such that the total price does not exceed B?

For the destructive variant we are trying to prevent p from being the unique $\mathscr{E}$-winner. The price functions $\rho = (\rho_1, \dots, \rho_n)$ with $\rho_i : \mathbb{N} \to \mathbb{N}$ describe how much it costs the briber to move p in a voter's vote forward (in the constructive case) or backward (in the destructive case). In particular, $\rho_i(k)$ is the cost of moving p in the ith voter's vote k positions forwards or backwards (for the constructive or destructive variant, respectively). To capture this behavior we require, for the constructive variant, that ρ_i is *nondecreasing* (i.e., $\rho_i(\ell) \leq \rho_i(\ell+1)$), $\rho_i(0) = 0$, and $\rho_i(\ell) = \rho_i(\ell-1)$ for each $\ell \geq r$ with r being the position of p in the (not-bribed) vote of voter i. Analogously, for the destructive variant we require that ρ_i is non*in*creasing, $\rho_i(0) = 0$, and $\rho_i(\ell) = \rho_i(\ell-1)$ for each $\ell \geq |C| - r + 1$ with r being the position of p in the (not-bribed) vote of voter i. For both variants the last condition is a technical requirement so that we cannot move p beyond the first or last position in a vote.

For all iterated variants of scoring rules defined in Chapter 2 we found that they are NP-hard for constructive and destructive shift bribery in both winner models. Furthermore, the price function as defined above only allows moving the designated candidate forwards in the constructive case (respectively, backwards in the destructive case) which makes sense for voting rules for which a candidate's final result in an election can only be improved if she is moved forward in the preferences of the voters. This so-called *monotonicity property* holds for scoring rules but does not hold for the iterative version except for iterated plurality and iterated veto. Therefore, we investigate whether the complexity changes if we drop the requirement that the designated candidate can only be shifted in one direction and give two examples of iterative scoring rules for which the problem still remains NP-hard.

We conjecture that the complexity also remains the same for all of our other nonmonotonic iterative scoring rules.

6.2 Publication – Maushagen, Neveling, Rothe, and Selker [106]

C. Maushagen, M. Neveling, J. Rothe, and A.-K. Selker. Complexity of shift bribery for iterative voting rules. *Journal of Artificial Intelligence Research*. Submitted.

A preliminary version of this paper was published in the proceedings of the *17th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS'18, see [107]).

6.3 Personal Contribution

The writing was done jointly with my co-authors. Theorems 2, 5, 6, 7, 8, 13, and 14 are to be attributed to my contribution.

Complexity of Shift Bribery for Iterative Voting Rules*

Cynthia Maushagen CYNTHIA.MAUSHAGEN@HHU.DE
Marc Neveling MARC.NEVELING@HHU.DE
Jörg Rothe ROTHE@HHU.DE
Ann-Kathrin Selker ANN-KATHRIN.SELKER@HHU.DE
Heinrich-Heine-Universität Düsseldorf
Universitätsstr. 1
40225 Düsseldorf, Germany

Abstract

In iterative voting systems, candidates are eliminated in consecutive rounds until either a fixed number of rounds is reached or the set of remaining candidates does not change anymore. We focus on iterative voting systems based on the positional scoring rules plurality, veto, and Borda and study their resistance against shift bribery attacks due to Elkind, Faliszewski, and Slinko (2009) and Kaczmarczyk and Faliszewski (2016). In constructive shift bribery (Elkind et al., 2009), an attacker seeks to make a designated candidate win the election by bribing voters to shift this candidate in their preferences; in destructive shift bribery (Kaczmarczyk & Faliszewski, 2016), the briber's goal is to prevent this candidate's victory. We show that many iterative voting systems are resistant to these types of attack, i.e., the corresponding decision problems are NP-hard. These iterative voting systems include iterated plurality as well as the voting rules due to Hare (see, e.g., the book by Taylor, 2005), Coombs (see, e.g., the article by Levin & Nalebuff, 1995), Baldwin (1926), and Nanson (1882); variants of Hare voting are also known as single transferable vote, instant-runoff voting, and alternative vote.

1. Introduction

One of the main themes in computational social choice (Brandt, Conitzer, Endriss, Lang, & Procaccia, 2016; Rothe, 2015) is to study the complexity of manipulative attacks on voting systems, in the hope that proving computational hardness of such attacks may provide some sort of protection against them. Besides manipulation (Bartholdi, Tovey, & Trick, 1989; Conitzer, Sandholm, & Lang, 2007)—also referred to as strategic voting—and electoral control (Bartholdi, Tovey, & Trick, 1992; Hemaspaandra, Hemaspaandra, & Rothe, 2007), much work has been done to study bribery attacks. For a comprehensive overview of the formal models and the related complexity results, we refer to the book chapters by Conitzer and Walsh (2016) for manipulation, by Faliszewski and Rothe (2016) for control and bribery, and by Baumeister and Rothe (2015) for all three topics.

Bribery in voting was introduced by Faliszewski, Hemaspaandra, and Hemaspaandra (2009a, see also the article by Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009b). In their model, a briber intends to change the outcome of an election to his or her own advantage by bribing certain voters without

*. This paper extends the preliminary conference versions that appear in the proceedings of the *17th International Conference on Autonomous Agents and Multiagent Systems* (AAMAS'18, see Maushagen, Neveling, Rothe, & Selker, 2018) and in the nonarchival website proceedings of the *International Symposium on Artificial Intelligence and Mathematics* (ISAIM'18) by presenting all proofs (some of which were omitted in the conference versions due to space limitations) in full detail, by adding new results on iterated veto and veto with runoff in Theorems 11 and 12, and by adding more illustrating examples and discussion (such as the discussion in Section 7 with new Theorems 13 and 14).

exceeding a given budget. Bribery shares some features with manipulation, as the briber (just like a strategic voter) has to find the right preference orders that the bribed voters are then requested to change their votes to. Bribery also shares some features with electoral control, as the briber (just like an election chair) has to pick the right voters to bribe so as to make the cost of bribing them as inexpensive as possible and to stay within the allowed budget.

We will focus on *shift bribery*, which was introduced by Faliszewski et al. (2009b) in the context of so-called irrational voters for Copeland elections and was then studied in detail by Elkind et al. (2009) for the constructive variant (where the briber's goal is to make a favorite candidate win the election) and was later studied by Kaczmarczyk and Faliszewski (2016) in the destructive variant (where the briber's goal is to make sure that a despised candidate does not win the election). In *swap bribery*, which generalizes shift bribery, the briber has to pay for each swap of any two candidates in the votes. Shift bribery additionally requires that swaps always involve the designated candidate that the briber wants to see win (in the constructive case) or not win (in the destructive case).

A natural interpretation of swap bribery—and thus in particular of shift bribery—regards *campaign management*: A campaign manager organizing a political campaign for some candidate seeks to influence the public opinion about this candidate by legal activities such as, e.g., running targeted television ads. Those ads might influence voters to change their opinion (and consequently their vote) of the targeted candidate positively or negatively. Campaign managers are restricted by a budget and need to choose the right ads to run in order to increase their candidates' chances of winning. Shift bribery can be seen to model campaign management in a more ethical way than general swap bribery, as campaign managers then always target their own candidates only and thus cannot change the voters' opinions over pairs of other candidates.

Another natural interpretation of swap bribery regards election fraud detection: If the winner of an election can be dethroned by only a few changes (by swapping candidates) to the votes then the election might have been tampered with or, from a more optimistic viewpoint, small errors in the counting of the votes might have influenced the election result. In that situation, a recounting would be required since for a close election result only few errors in the counting are needed to elect a candidate that is not the "true" winner of the election. This has been studied as the *margin of victory* (Xia, 2012; Reisch, Rothe, & Schend, 2014), which is closely related to destructive bribery. In this context, shift bribery models a more fine-grained search for election fraud which targets only a specific candidate.

Swap bribery generalizes the possible winner problem (Konczak & Lang, 2005; Xia & Conitzer, 2011), which itself is a generalization of unweighted coalitional manipulation. Therefore, each of the many hardness results known for the possible winner problem is directly inherited by the swap bribery problem. This was the motivation for Elkind et al. (2009) to look at restricted variants of swap bribery such as shift bribery.

Even though shift bribery possesses a number of hardness results (Elkind et al., 2009), it has also been shown to allow exact and approximate polynomial-time algorithms in a number of cases (Elkind et al., 2009; Elkind & Faliszewski, 2010; Schlotter, Faliszewski, & Elkind, 2017). For example, Elkind et al. (2009) provided a 2-approximation algorithm for shift bribery when using Borda voting.[1] This result was

1. In *Borda* with m candidates, each vote is a linear order of the candidates, the ith candidate in a vote scores $m-i$ points, and whoever has the most points wins. Borda is a very prominent positional scoring rule and can be described by the scoring vector $(m-1, m-2, \dots, 0)$. Other prominent positional scoring rules are *plurality*, where only the top candidates in the votes score a point and no one else (i.e., plurality has the scoring vector $(1, 0, \dots, 0)$), and *veto* (a.k.a. *antiplurality*), where all except the bottom candidates in the votes score a point (i.e., veto has the scoring vector $(1, \dots, 1, 0)$); again, whoever has the most points wins in these rules.

extended by Elkind and Faliszewski (2010) to all positional scoring rules; they also obtained somewhat weaker approximations for Copeland and maximin voting. Very recently Faliszewski, Manurangsi, and Sornat (2019) further extended this result to a polynomial-time approximation scheme. For Bucklin and fallback voting, the shift bribery problem is even exactly solvable in polynomial time (Schlotter et al., 2017).[2] In addition, Bredereck, Chen, Faliszewski, Nichterlein, and Niedermeier (2014b) were the first to analyze shift bribery in terms of *parameterized* complexity, and only recently a long-standing open problem regarding the parameterized complexity of bribery (including shift bribery) with the number of candidates as the parameter (see the survey by Bredereck, Chen, Faliszewski, Guo, Niedermeier, and Woeginger (2014a) for a deeper discussion on this problem) was solved by Knop, Koutecký, and Mnich (2017) for a multitude of voting rules. Furthermore, Bredereck, Faliszewski, Niedermeier, and Talmon (2016b) introduced combinatorial shift bribery in which a single shift bribery action affects multiple voters and Bredereck, Faliszewski, Niedermeier, and Talmon (2016a) studied shift bribery in the context of multiwinner elections for various committee selection rules.

While the complexity of shift bribery has been comprehensively investigated for many standard voting rules, it has not been considered yet for *iterative* voting systems. To close this glaring gap, we study shift bribery for eight iterative voting systems that are based on any one of the Borda, plurality, and veto rules (see Footnote 1 for their definitions) and that each proceed in rounds, eliminating after each except the last round the candidates performing worst in a certain sense:

- The *system of Baldwin (1926)* eliminates the candidates with the lowest Borda score and
- the *system of Nanson (1882)* eliminates the candidates whose scores are lower than the average Borda score, while
- the *system of Hare* (see, e.g., the book by Taylor, 2005) eliminates the candidates with the lowest plurality score,
- the system called *iterated plurality* (again see, e.g., the book by Taylor, 2005) eliminates the candidates that do not have the highest plurality score,
- the system called *iterated veto* is defined analogously to iterated plurality, except based on the veto rather than the plurality score, and
- the *system of Coombs* (defined, e.g., in the paper by Levin & Nalebuff, 1995) eliminates the candidates with the lowest veto score.

The last two systems that we consider differ from the above iterative voting systems because they always use exactly two rounds:

- *Plurality with runoff* (as defined, e.g., in the book by Taylor, 2005) eliminates the candidates that do not have the highest plurality score, except in the case where there is a unique plurality winner—it then eliminates all candidates that do not have the highest or second-highest plurality score; in the second round, all remaining candidates with the highest plurality score then win.

2. Faliszewski, Reisch, Rothe, and Schend (2015) have complemented these results on Bucklin and fallback voting. In particular, they studied a number of bribery problems for these rules, including a variant called "extension bribery," which was previously introduced by Baumeister, Faliszewski, Lang, and Rothe (2012) in the context of campaign management when the voters' ballots are truncated.

	Hare	Coombs	Baldwin	Nanson
Constructive	NP-c (Thm. 1)	NP-c (Thm. 3)	NP-c (Thm. 5)	NP-c (Thm. 7)
Destructive	NP-c (Thm. 2)	NP-c (Thm. 4)	NP-c (Thm. 6)	NP-c (Thm. 8)
	Iterated Plurality	Plurality with Runoff	Iterated Veto	Veto with Runoff
Constructive	NP-c (Thm. 9)	NP-c (Thm. 9)	NP-c (Thm. 11)	NP-c (Thm. 11)
Destructive	NP-c (Thm. 10)	NP-c (Thm. 10)	NP-c (Thm. 12)	NP-c (Thm. 12)

Table 1: Summary of complexity results for shift bribery problems

- *Veto with runoff* is defined analogously, except that veto scores instead of plurality scores and veto winners instead of plurality winners are considered.

These voting systems have been thoroughly studied and are also used in the real world. Among the systems we consider, Hare voting and variants thereof (some of which are called single transferable vote, instant-runoff voting, or alternative vote) are most widely used, for example in Australia, India, Ireland, New Zealand, Pakistan, the UK, and the USA.

Table 1 gives an overview of our complexity results for constructive and destructive shift bribery in our eight voting systems,[3] where the shorthand NP-c stands for "NP-complete." Our results complement results by Davies, Katsirelos, Narodytska, Walsh, and Xia (2014) who have shown unweighted coalitional manipulation to be NP-complete for Baldwin and Nanson voting (even with just a single manipulator)—and also for the underlying Borda system (with two manipulators; for the latter result, see also the paper by Betzler, Niedermeier, and Woeginger (2011)). Davies et al. (2014) also list various appealing features of the systems by Baldwin and Nanson, including that they have been applied in practice (namely, in the State of Michigan in the 1920s, in the University of Melbourne from 1926 through 1982, and in the University of Adelaide since 1968) and that (unlike Borda itself) they both are Condorcet-consistent.[4] Axiomatic properties of iterative voting systems were also studied by Freeman, Brill, and Conitzer (2014) who showed, in particular, that Hare is the only iterative voting system based on scoring rules that satisfies the independence-of-clones property. Further, it was shown by Bartholdi and Orlin (1991) that Hare (which is called STV in their work) is NP-hard to manipulate even with only one manipulator. This result was complemented by Davies, Narodytska, and Walsh (2012) who showed the same result for Coombs and a general class of iterative versions of scoring rules. For plurality with runoff, it was shown by Conitzer et al. (2007) that unweighted coalitional manipulation is NP-hard. Finally, plurality with runoff and veto with runoff were also studied by Erdélyi, Neveling, Reger, Rothe, Yang, and Zorn (2021) with respect to electoral control.

This paper is organized as follows. In Section 2, we will provide the needed definitions regarding elections and voting systems (in particular, iterative voting systems), define the shift bribery problem, and give some background on computational complexity. We will then study the complexity of shift bribery for Hare and Coombs elections in Section 3, for Baldwin and Nanson elections in Section 4, for iterated plurality and plurality with runoff in Section 5, and for iterated veto and veto with runoff in Section 6.

3. As shown by Xia (2012), destructive bribery is closely related to the *margin of victory*, a critical robustness measure for voting systems. Reisch et al. (2014) add to this connection by showing that the former problem can be easy while the latter is hard.
4. A *Condorcet winner* is a candidate who defeats every other candidate in a pairwise comparison. Such a candidate does not always exist. A voting rule is *Condorcet-consistent* if it chooses only the Condorcet winner whenever there exists one.

Further, in Section 7 we will discuss how the nonmonotonicity property of our iterative voting systems can be exploited in our reductions showing NP-hardness, exemplified for Hare voting and plurality with runoff. Finally, we will conclude in Section 8 by presenting some open problems related to our work.

2. Preliminaries

Below, we provide the needed notions and notation.

Elections and voting systems. An *election* is specified as a pair (C,V) with C being a set of candidates and V a profile of the voters' preferences over C, typically given by a list of linear orders of the candidates. A *voting system* is a function that maps each election (C,V) to a subset of C, the *winner(s) of the election*. An important class of voting systems is the family of positional scoring rules whose most prominent members are plurality, veto, and Borda count, see, e.g., the book chapters by Zwicker (2016) and Baumeister and Rothe (2015) and also the survey by Rothe (2019) on using Borda in collective decision making.

Recall from Footnote 1 in Section 1 that, in *plurality*, each voter gives her top-ranked candidate one point; in *veto* (a.k.a. *antiplurality*), each voter gives all except the bottom-ranked candidate one point; in *Borda* with m candidates, each candidate in position i of the voters' rankings scores $m-i$ points; and the winners in each case are those candidates scoring the most points.

Iterative voting systems. The iterative voting systems we will study are based on plurality, veto, and Borda but, unlike those, their election winner(s) are determined in consecutive rounds. For all iterative voting systems considered here except for plurality with runoff and veto with runoff (which will be defined shortly afterwards), if in some round all remaining candidates have the same score (for instance, there may be only one candidate left), then all those candidates are proclaimed winners of the election. In each preceding round, however, all candidates with the lowest score are eliminated.[5]

Recall from Section 1 that the eight scoring methods we will use work as follows: The iterative voting systems due to *Hare*, *Coombs*, and *Baldwin* use, respectively, plurality, veto, and Borda scores in order to decide which candidates are the weakest and thus to be removed. The *Nanson* system eliminates in every (except the last) round all candidates that have less than the average Borda score. *Iterated plurality* eliminates all candidates that do not have the highest plurality score, and *iterated veto* eliminates all candidates that do not have the highest veto score.

Unlike the above multiple-round iterative voting systems, *plurality with runoff* (respectively, *veto with runoff*) always proceeds in two rounds: In the first round, it eliminates all candidates that do not have the highest plurality score (respectively, veto score), unless there is a unique plurality winner (respectively, veto winner) in which case all candidates are eliminated except those with the highest or second-highest plurality score (respectively, veto score); in the second round, all candidates with the highest plurality score (respectively, veto score) win.

Shift bribery. For any given voting system $\mathscr{E}$, we now define the problem $\mathscr{E}$-SHIFT-BRIBERY, which is a special case of $\mathscr{E}$-SWAP-BRIBERY, introduced by Faliszewski et al. (2009b) in the context of so-called irrational voters for Copeland and then comprehensively studied by Elkind et al. (2009). Informally stated, given a profile of votes, a swap-bribery price function exacts a price for each swap of any two candidates in the votes, and in shift bribery only swaps involving the designated candidate are allowed.

5. In the original sources defining these iterative voting systems as stated in the Introduction, certain tie-breaking schemes are used whenever more than one candidate has the lowest score in some round. For the sake of convenience and uniformity, however, we prefer eliminating them all and will therefore disregard tie-breaking issues in such a case.

$\mathscr{E}$-CONSTRUCTIVE-SHIFT-BRIBERY	
Given:	An election (C,V) with n votes, a designated candidate $p \in C$, a budget B, and a list of price functions $\rho = (\rho_1, \dots, \rho_n)$.
Question:	Is it possible to make p the unique $\mathscr{E}$ winner of the election by shifting p in the votes such that the total price does not exceed B?

In the corresponding problem $\mathscr{E}$-DESTRUCTIVE-SHIFT-BRIBERY, given the same input, we ask whether it is possible to prevent p from being a unique winner.

These problems are here defined in the unique-winner model where a constructive (respectively, destructive) bribery action is considered successful only if the designated candidate can be made (respectively, can be prevented from being) the only winner of the election. We also consider these problems in the nonunique-winner model where for a constructive (respectively, destructive) bribery action to be considered successful it is required that the designated candidate is merely one among possibly several winners (respectively, does not win at all). Note that a yes-instance of $\mathscr{E}$-CONSTRUCTIVE-SHIFT-BRIBERY in the unique-winner model is also a yes-instance of the same problem in the nonunique-winner model, whereas a yes-instance of $\mathscr{E}$-DESTRUCTIVE-SHIFT-BRIBERY in the nonunique-winner model is also a yes-instance of the same problem in the unique-winner model; analogous statements apply to the no-instances of these problems by swapping the unique-winner model with the nonunique-winner model. We will make use of these facts in our proofs that all work in both winner models.

Membership in NP is obvious for all considered problems, so it will be enough to show only NP-hardness so as to prove in fact NP-completeness.

Regarding the list of price functions $\rho = (\rho_1, \dots, \rho_n)$ with $\rho_i : \mathbb{N} \to \mathbb{N}$, in the constructive case $\rho_i(k)$ indicates the price the briber has to pay in order to move p in vote i by k positions to the top (respectively, to the bottom in the destructive case). For all i, we require that ρ_i is nondecreasing ($\rho_i(\ell) \leq \rho_i(\ell+1)$), $\rho_i(0) = 0$, and if p is at position r in vote i then $\rho_i(\ell) = \rho_i(\ell-1)$ whenever $\ell \geq r$ in the constructive case (respectively, whenever $\ell \geq |C| - r + 1$ in the destructive case). The latter condition ensures that p can be shifted upward no farther than to the top (respectively, the bottom).[6] When the voter i in ρ_i is clear from the context, we omit the subscript and simply write ρ.

Our proofs use the following notation: A vote of the form $a\ b\ c$ indicates that the voter ranks candidate a on top position, then candidate b, and last candidate c. If a set $S \subseteq C$ of candidates appears in a vote as $\overrightarrow{S}$, its candidates are placed in this position in lexicographical order. By $\overleftarrow{S}$ we mean the reverse of the lexicographical order of the candidates in S. If S occurs in a vote without an arrow on top, the order in which the candidates from S are placed here does not matter for our argument. We use $\cdots$ in a vote to indicate that the remaining candidates may occur in any order.

Computational complexity. We assume familiarity with the standard concepts of complexity theory, including the classes P and NP, polynomial-time many-one reducibility, and NP-hardness and -completeness. We will use the following NP-complete problem:

EXACT-COVER-BY-3-SETS (X3C)	
Given:	A set $X = \{x_1, \dots, x_{3m}\}$ and a family of sets $\mathscr{S} = \{S_1, \dots, S_n\}$ such that $S_i \subseteq X$ and $|S_i| = 3$ for all $S_i \in \mathscr{S}$.
Question:	Does there exist an exact cover of X, i.e., a subset $\mathscr{S}' \subseteq \mathscr{S}$ such that $|\mathscr{S}'| = m$ and $\bigcup_{S_i \in \mathscr{S}'} S_i = X$?

6. If p is in the first (respectively, the last) position of a vote, this voter cannot be bribed and we tacitly assume a price function of $\rho(t) = 0$ for each $t \geq 0$. We will disregard these voters when setting price functions for the other voters in our proofs.

In instances of X3C, we assume that each $x_j \in X$ is contained in exactly three sets $S_i \in \mathscr{S}$; thus $|X| = |\mathscr{S}|$. Gonzalez (1985) shows that X3C under this restriction remains NP-hard. Note that if not stated otherwise, we will use $(X, \mathscr{S})$ to denote an X3C instance, where $X = \{x_1, \ldots, x_{3m}\}$, $\mathscr{S} = \{S_1, \ldots, S_{3m}\}$, and $S_i = \{x_{i,1}, x_{i,2}, x_{i,3}\}$. Also note that we assume $x_{i,1}$ to be the $x_j \in S_i$ with the smallest subscript and $x_{i,3}$ to be the $x_j \in S_i$ with the largest subscript.

ONE-IN-THREE-POSITIVE-3SAT	
Given:	A set X of boolean variables, a set S of clauses over X, each containing exactly three unnegated literals.
Question:	Does there exist a truth assignment to the variables in X such that exactly one literal is set to true for each clause in S?

In instances of ONE-IN-THREE-POSITIVE-3SAT, we assume that each $x_j \in X$ is contained in exactly three clauses. Porschen, Schmidt, Speckenmeyer, and Wotzlaw (2014) show that this restricted problems remains NP-complete.

For more background on computational complexity, the reader is referred to, for instance, the textbooks by Garey and Johnson (1979), Papadimitriou (1995), and Rothe (2005).

3. Hare and Coombs

We start by showing NP-hardness of shift bribery for Hare elections.

Theorem 1. *In both the unique-winner and the nonunique-winner model, Hare-*CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. NP-hardness follows by a reduction from X3C. Given an X3C instance $(X, \mathscr{S})$, construct an instance $((C,V), p, B, \rho)$ of Hare-CONSTRUCTIVE-SHIFT-BRIBERY with candidate set $C = X \cup \mathscr{S} \cup \{p\}$, designated candidate p, and the following list V of votes, with # denoting their number:

#	vote	for
1	$S_i\ x_{i,1}\ \overrightarrow{X \setminus \{x_{i,1}\}}\ \cdots$	$1 \le i \le 3m$
1	$S_i\ x_{i,2}\ \overrightarrow{X \setminus \{x_{i,2}\}}\ \cdots$	$1 \le i \le 3m$
1	$S_i\ x_{i,3}\ \overrightarrow{X \setminus \{x_{i,3}\}}\ \cdots$	$1 \le i \le 3m$
4	$x_i\ \overrightarrow{X \setminus \{x_i\}}\ \cdots$	$1 \le i \le 3m$
1	$S_i\ p\ \cdots$	$1 \le i \le 3m$
3	$p\ \cdots$	

For votes of the form $S_i\ p\ \cdots$, we use the price function $\rho(1) = 1$, and $\rho(t) = m+1$ for all $t \ge 2$; and for every other vote, we use the price function ρ with $\rho(t) = m+1$ for all $t \ge 1$. Finally, set the budget $B = m$. Without loss of generality, we assume that $m > 1$.

Note that p scores three points while the rest of the candidates score four points each, so p is eliminated in the first round and does not win the election without bribing voters.

We claim that $(X,\mathscr{S})$ is in X3C if and only if $((C,V),p,B,\rho)$ is in Hare-CONSTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

$(\Rightarrow)$ Suppose that $(X,\mathscr{S})$ is a yes-instance of X3C. Then there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of size m. We now show that it is possible for p to become a unique Hare winner of an election obtained by shifting p in the votes without exceeding the budget B. For every $S_i \in \mathscr{S}'$, we bribe the voter with the vote of the form $S_i\ p\ \cdots$ by shifting p once, so her new vote is of the form $p\ S_i\ \cdots$; each such bribe action costs us only 1 from our budget, so the budget will not be exceeded. In the first round, p now has $m+3$ points, every candidate from $\mathscr{S}'$ has 3 points, and every other candidate has 4 points. Therefore, all candidates in $\mathscr{S}'$ are eliminated. In the second round, all candidates in X now gain one point from the elimination of $\mathscr{S}'$, since it is an exact cover. Therefore, p and all candidates in X proceed to the next round and the remaining candidates $\mathscr{S} \setminus \mathscr{S}'$ are eliminated. In the next round with only p and the candidates from X remaining, p has $3m+3$ points, while every candidate in X scores 7 points (recall that every $x_i \in X$ is contained in exactly three members of $\mathscr{S}$). Since all candidates from X have been eliminated now, p is the only remaining candidate and thus the unique Hare winner.

$(\Leftarrow)$ Suppose that $(X,\mathscr{S})$ is a no-instance of X3C. Then no subset $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq m$ covers X. We now show that we cannot make p become a Hare winner of an election obtained by bribing voters without exceeding budget B. Note that we can only bribe at most m voters with votes of the form $S_i\ p\ \cdots$ without exceeding the budget. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that $S_i \in \mathscr{S}'$ exactly if the voter with the vote $S_i\ p\ \cdots$ has been bribed. Clearly, $|\mathscr{S}'| \leq m$ and in all those votes p has been shifted once to the left, so p is now ranked first in these votes. Therefore, p now has $3+|\mathscr{S}'|$ points and every $S_i \in \mathscr{S}'$ scores 3 points. Since every other candidate scores as many points as before the bribery (namely, 4 points), the candidates in $\mathscr{S}'$ are eliminated in the first round. Let $X' = \{x_i \in X \mid x_i \notin \bigcup_{S_j \in \mathscr{S}'} S_j\}$ be the subset of candidates $x_i \in X$ that are not covered by $\mathscr{S}'$. We have $X' \neq \emptyset$ (otherwise, $\mathscr{S}'$ would have been an exact cover of X). In the second round, unlike the candidates from $X \setminus X'$, the candidates in X' will not gain additional points from eliminating the candidates in $\mathscr{S}'$. Thus, in the current situation, the candidates from X' and $\mathscr{S} \setminus \mathscr{S}'$ are trailing behind with 4 points each and are eliminated in this round.[7] Therefore, in the next round, only p and the candidates from $X \setminus X'$ are remaining in the election. Let $x_\ell \in X \setminus X'$ be the candidate from $X \setminus X'$ with the smallest subscript. Since all candidates from $\mathscr{S}$ are eliminated, p has $3m+3$ points and every candidate from $X \setminus X'$ except x_ℓ has 7 points. On the other hand, x_ℓ gains additional points from eliminating the candidates from X'; therefore, x_ℓ survives this round by scoring more than 7 points. In the final round with only p and x_ℓ remaining, p is eliminated, since $3m \cdot 7 > 3m+3$. $\square$

Example 1. *Let $(X,\mathscr{S})$ be a yes-instance of* X3C *defined by*

$$\begin{aligned} X &= \{x_1,\ldots,x_6\} \text{ and} \\ \mathscr{S} &= \{\{1,2,3\},\{4,5,6\},\{2,3,6\},\{2,4,5\},\{1,3,4\},\{1,5,6\}\}. \end{aligned}$$

Construct $((C,V),p,B,\rho)$ from $(X,\mathscr{S})$ as in the proof of Theorem 1; in particular, the budget is $B=2$. If we bribe the voters with $S_1\ p\ \cdots$ and $S_2\ p\ \cdots$ so as to shift p to the top of their votes, p will be the unique winner of the election that proceeds as follows (where the numbers in the columns below candidates give their scores):

7. Note that in the case that $|\mathscr{S}'| = 1$, i.e., only one voter was bribed, p also gets eliminated in this round and is consequently not a Hare winner, which is what we want to show. Therefore, we will now assume that at least two voters were bribed.

Round	p	$x \in X$	S_1, S_2	S_3, S_4, S_5, S_6
1	5	4	3	4
2	5	5	*out*	4
3	9	7	*out*	*out*

Now consider a no-instance $(X, \mathscr{S})$ *of* X3C *with*

$$
\begin{aligned}
X &= \{x_1, \ldots, x_6\} \text{ and} \\
\mathscr{S} &= \{\{1,2,4\}, \{4,5,6\}, \{2,3,6\}, \{2,3,5\}, \{1,3,4\}, \{1,5,6\}\}.
\end{aligned}
$$

If we bribe no voter, p gets eliminated in the first round and so does not win. If we bribe one voter, say the one with vote $S_1\ p\ \cdots$, then p gets eliminated in the second round:

Round	p	x_1	x_2, x_4	x_3, x_5, x_6	S_1	$S_i \in \mathscr{S} \setminus \{S_1\}$
1	4	4	4	4	3	4
2	4	5	5	4	*out*	4
3	*out*	≥ 28	≥ 7	*out*	*out*	*out*

Since $(X, \mathscr{S})$ is a no-instance of X3C*, no matter which two subsets $S_i, S_j \in \mathscr{S}$ we choose, at least one x_k is in both subsets, so p loses the direct comparison in the last round. For example, if we bribe the voters with $S_1\ p\ \cdots$ and $S_2\ p\ \cdots$, the election proceeds as follows:*

Round	p	x_1	x_3	x_4	x_2, x_5, x_6	S_1, S_2	S_3, S_4, S_5, S_6
1	5	4	4	4	4	3	4
2	5	5	4	6	5	*out*	4
3	9	14	*out*	7	7	*out*	*out*
4	9	42	*out*	*out*	*out*	*out*	*out*

This completes Example 1.

Next, we show that shift bribery is NP-hard for Hare also in the destructive case.

Theorem 2. *In both the unique-winner and the nonunique-winner model, Hare-*DESTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. Again, we use a reduction from X3C. Construct from a given X3C instance $(X, \mathscr{S})$ a Hare-DESTRUCTIVE-SHIFT-BRIBERY instance $((C,V), p, B, \rho)$ as follows. Let $D = \{d_1, \ldots, d_{3m}\}$ be a set of $3m$ dummy candidates. The candidate set is $C = X \cup \mathscr{S} \cup D \cup \{p, w\}$ with designated candidate p. The list V of votes is constructed as follows:

#	vote	for
2	$S_i\ x_{i,1}\ \overrightarrow{X \setminus \{x_{i,1}\}}\ w\ p\ \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,2}\ \overrightarrow{X \setminus \{x_{i,2}\}}\ w\ p\ \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,3}\ \overrightarrow{X \setminus \{x_{i,3}\}}\ w\ p\ \cdots$	$1 \leq i \leq 3m$
7	$x_i\ \overrightarrow{X \setminus \{x_i\}}\ w\ p\ \cdots$	$1 \leq i \leq 3m$
1	$p\ S_i\ \cdots$	$1 \leq i \leq 3m$
12	$w\ p\ \cdots$	
$18m$	$p\ \cdots$	
6	$d_i\ S_i\ p\ \cdots$	$1 \leq i \leq 3m$

For votes of the form $p\ S_i\ \cdots$, we use the price function $\rho(1) = 1$, and $\rho(t) = m+1$ for all $t \geq 2$; and for every other vote, we use the price function ρ with $\rho(t) = m+1$ for all $t \geq 1$. Finally, set the budget $B = m$.

Without bribing, the election (C,V) proceeds as follows:

Round	p	w	$x_i \in X$	$S_i \in \mathscr{S}$	$d_i \in D$
1	$21m$	12	7	6	6
2	$39m$	12	13	out	out
3	$39m+12$	out	13	out	out

It follows that p has won the election after three rounds.

We claim that $(X,\mathscr{S})$ is in X3C if and only if $((C,V),p,B,\rho)$ is in Hare-DESTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

$(\Rightarrow)$ Suppose that $(X,\mathscr{S})$ is a yes-instance of X3C. Then there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of size m. We now show that it is possible to eliminate p from an election obtained by shifting p in the votes without exceeding the budget B. For every $S_i \in \mathscr{S}'$, we bribe the voter with the vote of the form $p\ S_i\ \cdots$ by shifting p once, so her new vote is of the form $S_i\ p\ \ldots$; each such bribe action costs us only 1 from our budget, so the budget will not be exceeded. Now the election proceeds as follows:

Round	p	w	$x_i \in X$	$S_i \in \mathscr{S}'$	$S_i \in \mathscr{S} \setminus \mathscr{S}'$	$d_i \in D$
1	$20m$	12	7	7	6	6
2	$32m$	12	11	13	out	out
3	$32m$	$33m+12$	out	13	out	out
4	$39m$	$39m+12$	out	out	out	out

We see that p is eliminated in the fourth round and w wins.

$(\Leftarrow)$ Suppose that $(X,\mathscr{S})$ is a no-instance of X3C. Then no subset $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq m$ covers X. We now show that p will not be eliminated in any election obtained by bribing voters without exceeding budget B but will in fact become the only winner. Note that we can only bribe at most m voters with votes of the form $p\ S_i\ \cdots$ without exceeding the budget. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that for every $S_i \in \mathscr{S}'$ we have bribed the voter whose vote is $p\ S_i\ \cdots$. We can assume that $|\mathscr{S}'| > 0$. Every candidate in $\mathscr{S}'$ will gain an additional point and therefore survives the first round. All candidates from D and $\mathscr{S} \setminus \mathscr{S}'$ will be eliminated, since p only loses at most m points.

In the second round, the remaining candidates from $\mathscr{S}$ will additionally gain six points from the elimination of candidates in D and will score 13 points in this round (and in all subsequent rounds with p still standing). If a candidate $S_i \in \mathscr{S}$ was eliminated in the previous round, every $x_i \in S_i$ gains two additional points in this round. Partition X into sets X_0, X_1, X_2, and X_3 so that $x_i \in X_k \Leftrightarrow |\{S_j \in \mathscr{S}' \mid x_i \in S_j\}| = k$ for $k \in \{0,1,2,3\}$. Note that X_0, X_1, X_2, and X_3 are disjoint and $|X_0| > 0$, but one or two of X_1, X_2, and X_3 may be empty. Then $x_i \in X_j$ scores $7 + (6-2j) \in \{7,9,11,13\}$ points depending on how many times x_i is covered by $\mathscr{S}'$. Therefore, every $x_i \in X_0$ scores more points than w who has 12 points. Thus there are candidates from X that survive this round and other candidates from X (more precisely, candidates from X_1, X_2, or X_3) who are eliminated.

In the third round, the candidate $x_\ell \in X$ with the smallest subscript who is still standing gains at least seven points from the eliminated candidates, so that x_ℓ scores at least 16 points.[8] All other candidates still score the same number of points as in the last round. Therefore, p scores at least $20m$ points, w scores still 12 points, every $S_i \in \mathscr{S}'$ scores 13 points, and every still standing candidate from X except x_ℓ scores at most 13 points. Since w can only gain additional points when all candidates from X are eliminated and only x_ℓ gains points from the elimination of candidates from $X \setminus \{x_\ell\}$ in the subsequent rounds, all candidates $X \setminus (\{x_\ell\} \cup X_0)$ and w are eliminated. Then all still standing candidates from $X_0 \setminus \{x_\ell\}$ and candidates from $\mathscr{S}'$ who each score 13 points are eliminated, which leaves p and x_ℓ in the last round. In this round, p scores $39m + 12$ points and x_ℓ scores $39m$ points, so p solely wins the election, no matter how we bribe voters within the budget, i.e., we have a no-instance of Hare-DESTRUCTIVE-SHIFT-BRIBERY in both winner models. □

Next, we turn to shift bribery for Coombs elections. While the idea of the reduction is similar, and perhaps even simpler than in the previous two proofs, the proof of correctness is way more involved.

Theorem 3. *In both the unique-winner and the nonunique-winner model, Coombs-*CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we now describe a reduction from X3C to Coombs-CONSTRUCTIVE-SHIFT-BRIBERY. Given an X3C instance $(X, \mathscr{S})$, construct an election (C, V) with the set $C = \{p, w, d_1, d_2, d_3\} \cup X \cup Y$ of candidates, where p is the designated candidate and $Y = \{y_i \mid x_i \in X\}$. Construct the following list V of votes:

#	vote	for
1	$\cdots\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p$	$1 \le i \le 3m$
$2m$	$\cdots\ p\ \overrightarrow{Y \setminus \{y_i\}}\ y_i\ x_i$	$1 \le i \le 3m$
$2m$	$\cdots\ p\ \overrightarrow{Y}\ w\ d_1\ d_2\ d_3$	
1	$\cdots\ p\ \overrightarrow{Y}\ w\ X\ d_1\ d_2\ d_3$	
m	$\cdots\ p\ \overrightarrow{Y}\ w$	

For votes of the form $\cdots\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p$, we use the price function $\rho(1) = \rho(2) = \rho(3) = 1$, and $\rho(t) = m+1$ for all $t \ge 4$; and for all the remaining votes, we use the price function $\rho(t) = m+1$ for all $t \ge 1$. Furthermore, our budget is $B = m$.

8. Since this candidate x_ℓ is still in the election, x_ℓ cannot have been in X_3 and thus must have had at least nine points.

The candidates have the following veto counts: p has $3m$ vetoes, each $x_i \in X$ has $2m$ vetoes, w has m vetoes, d_3 has $2m+1$ vetoes, and the remaining candidates each have 0 vetoes. Therefore, p will be eliminated in the first round and thus does not win the election.

We claim that $(X, \mathscr{S})$ is in X3C if and only if $((C,V), p, B, \rho)$ is in Coombs-CONSTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

($\Rightarrow$) Assume that $(X, \mathscr{S})$ is in X3C. This means that there exists a subset $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = m$ and $\bigcup_{S_i \in \mathscr{S}'} S_i = X$. So we have a partition of X into three sets, $X = X_1 \cup X_2 \cup X_3$, such that:

$$
\begin{aligned}
X_1 &= \{x_i \in S_i \mid x_i \text{ has the lowest subscript in } S_i \in \mathscr{S}'\}, \\
X_3 &= \{x_i \in S_i \mid x_i \text{ has the highest subscript in } S_i \in \mathscr{S}'\}, \text{ and} \\
X_2 &= X \setminus (X_1 \cup X_3).
\end{aligned}
$$

Let $Y = Y_1 \cup Y_2 \cup Y_3$ be the corresponding partition of Y.

We bribe the voters with votes of the form $\cdots\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p$ for $S_i \in \mathscr{S}'$ so that they change their votes to $\cdots\ p\ x_{i,1}\ x_{i,2}\ x_{i,3}$. Since $\mathscr{S}'$ is an exact cover of X, it follows that p now has a total of $2m$ vetoes, whereas each $x \in X_3$ receives an additional veto for a total of $2m+1$. The number of vetoes for the remaining candidates remain unchanged. If a candidate has the highest number of vetoes then she has the fewest number of points and cannot proceed to the next round (unless all candidates have the same score). Here, the candidates in X_3 and d_3 have the fewest number of points (and fewer than the other candidates) and therefore are eliminated in the first round.

Without the candidates in X_3, each candidate in X_2 gets an additional veto and the candidates in Y_3 each take all but one of the vetoes of the eliminated candidates in X_3. Furthermore, d_2 receives the vetoes of d_3. As a consequence, in the second round the candidates in X_2 and d_2 have the fewest number of points (and fewer than the remaining candidates) and are eliminated.

Similarly to the first round, vetoes from candidates in X_2 and d_2 are passed on to candidates in X_1 and Y_1 and to d_1. Thus the candidates have the following veto counts in the third round: p and each $y \in Y_2 \cup Y_3$ receive $2m$ vetoes, w receives m vetoes, each $y \in Y_1$ receives zero vetoes, and d_1 and each $x_i \in X_1$ receive $2m+1$ vetoes. Consequently, all the candidates $x_i \in X_1$ and d_1 are eliminated in the third round, so in the next round there are no candidates from X and no d_i, $1 \leq i \leq 3$.

It follows that w receives $2m+1$ additional vetoes in the fourth round, so w has the most vetoes in this round and is eliminated. We need $3m$ further rounds until p ends up as the last remaining candidate and sole winner of the election. In each of these rounds, the candidate in Y that is still alive and has the highest subscript has at least $2m+2m+1+m = 5m+1$ vetoes, while p always has only $3m$ vetoes.

($\Leftarrow$) Suppose that $(X, \mathscr{S})$ is a no-instance of X3C. We will show that $((C,V), p, B, \rho)$ then is a no-instance of Coombs-CONSTRUCTIVE-SHIFT-BRIBERY in the nonunique-winner (and thus also in the unique-winner) model. Observe that if we were going to make p a winner of the election, we would have to bribe at least m voters with a vote of the form $\cdots\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p$; otherwise, p would have at least $2m+1$ vetoes and would be eliminated right away in the first round. Due to our budget, on the other hand, we can bribe no more than m (and thus would have to bribe exactly m) such voters and cannot bribe any further voters. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that $S_i \in \mathscr{S}'$ exactly if the voter with the vote of the form $\cdots\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p$ has been bribed. Note that $|\mathscr{S}'| = m$ and $\mathscr{S}'$ does not cover X because we have a no-instance of X3C. Now p has only $2m$ vetoes and will not be eliminated in the first round.

Let X_1 be the set of candidates $x_i \in S_i$ for $S_i \in \mathscr{S}'$ with the smallest subscript in S_i, let X_2 be the set of candidates $x_i \in S_i$ for $S_i \in \mathscr{S}'$ with the second-smallest subscript in S_i, and let X_3 be the set of candidates $x_i \in S_i$ for $S_i \in \mathscr{S}'$ with the highest subscript in S_i. Note that $X_1 \cup X_2 \cup X_3 \neq X$, since $\mathscr{S}'$ does not cover X.

For w to have more vetoes than p, the candidates d_1, d_2, and d_3 need to be eliminated. For that to happen, there must be three rounds in which no other candidate has more than $2m+1$ vetoes. In the round where d_i, $1 \leq i \leq 3$, is eliminated, all still standing candidates in X_i are eliminated as well. Assume there were three rounds in which $2m+1$ was the maximal number of vetoes for a candidate. Then d_1, d_2, d_3, and all candidates in $X_1 \cup X_2 \cup X_3$ are eliminated. Note that those candidates that are not covered by $\mathscr{S}'$ always have only $2m$ vetoes and are still participating in the election. Therefore, in the next round, p and w have $3m$ vetoes each, the remaining candidates from X have at most $2m+1$ vetoes, and the candidates from Y have at most $2m$ vetoes. So even if p survives the first rounds with the candidates d_1, d_2, and d_3 still present, p will then surely be eliminated in the following round. If there is at least one voter who shifts p only one or two positions upward, then p has to drop out with d_1 or even before d_1 drops out, because at the latest after two rounds (with $2m+1$ being the maximal number of vetoes for a candidate) p receives another veto and thus has at least the same number of vetoes as d_1. $\square$

Example 2. *Let $(X, \mathscr{S})$ be a yes-instance of* X3C *defined by*

$$X = \{x_1, \ldots, x_6\} \text{ and}$$
$$\mathscr{S} = \{\{1,2,3\},\{4,5,6\},\{2,3,6\},\{2,4,5\},\{1,3,4\},\{1,5,6\}\}.$$

Construct $((C,V),p,B,\rho)$ from $(X,\mathscr{S})$ as in the proof of Theorem 3; in particular, the budget is $B = 2$. If we bribe the voters that correspond to the sets in the exact cover, S_1 and S_2, to change their votes from $\cdots\ x_1\ x_2\ x_3\ p$ and $\cdots\ x_4\ x_5\ x_6\ p$ to $\cdots p\ x_1\ x_2\ x_3$ and $\cdots p\ x_4\ x_5\ x_6$, then p alone wins the election that proceeds as follows, where in order to make this example easier to follow, the numbers in the table count the candidates' vetoes, not their points, i.e., the candidates with the highest *number in a round (row) get eliminated:*

Round	p	w	x_1,x_4	x_2,x_5	x_3,x_6	y_1	y_2	y_3	y_4	y_5	y_6	d_1	d_2	d_3
1	4	2	4	4	5	0	0	0	0	0	0	0	0	5
2	4	2	4	5	*out*	0	0	4	0	0	4	0	5	*out*
3	4	2	5	*out*	*out*	0	4	4	0	4	4	5	*out*	*out*
4	6	7	*out*	*out*	*out*	4	4	4	4	4	4	*out*	*out*	*out*
5	6	*out*	*out*	*out*	*out*	4	4	4	4	4	11	*out*	*out*	*out*
6	6	*out*	*out*	*out*	*out*	4	4	4	4	15	*out*	*out*	*out*	*out*
7	6	*out*	*out*	*out*	*out*	4	4	4	19	*out*	*out*	*out*	*out*	*out*
8	6	*out*	*out*	*out*	*out*	4	4	23	*out*	*out*	*out*	*out*	*out*	*out*
9	6	*out*	*out*	*out*	*out*	4	27	*out*	*out*	*out*	*out*	*out*	*out*	*out*
10	6	*out*	*out*	*out*	*out*	31	*out*	*out*	*out*	*out*	*out*	*out*	*out*	*out*

It follows that p is the sole winner of the election.

Now consider a no-instance $(X, \mathscr{S})$ with

$$X = \{x_1, \ldots, x_6\} \text{ and}$$
$$\mathscr{S} = \{\{1,2,4\},\{4,5,6\},\{2,3,6\},\{2,3,5\},\{1,3,4\},\{1,5,6\}\}.$$

Recall that we can bribe at most two voters. If we bribe fewer than two voters, however, p will be eliminated in the first round. Since $(X, \mathscr{S})$ is a no-instance of X3C, *no matter which two subsets $S_i, S_j \in \mathscr{S}$ we choose, at least one x_k is in both S_i and S_j. For example, if we bribe the voters that correspond to the sets S_1 and S_2, changing their votes from $\cdots\ x_1\ x_2\ x_4\ p$ and $\cdots\ x_4\ x_5\ x_6\ p$ to $\cdots p\ x_1\ x_2\ x_4$ and $\cdots p\ x_4\ x_5\ x_6$, then the election proceeds as follows:*

Round	p	w	x_1	x_2, x_5	x_3	x_4, x_6	y_1	y_2, y_5	y_3	y_4, y_6	d_1	d_2	d_3
1	4	2	4	4	4	5	0	0	0	0	0	0	5
2	4	2	4	5	4	*out*	0	0	0	4	0	5	*out*
3	5	2	5	*out*	4	*out*	0	4	0	4	5	*out*	*out*
4	*out*	2	*out*	*out*	4	*out*	4	4	0	4	*out*	*out*	*out*
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$

Since x_4 is in both S_1 and S_2, p gets an additional veto in round 3 and is subsequently eliminated. The same will happen for similar reasons in every other case.

This completes Example 2.

We now modify the previous reduction so as to work for the destructive case in Coombs elections.

Theorem 4. *In both the unique-winner and the nonunique-winner model, Coombs-*DESTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we again reduce from the NP-complete problem X3C to Coombs-DESTRUCTIVE-SHIFT-BRIBERY. Given an X3C instance $(X, \mathscr{S})$ where we may assume that $m > 2$ for $|X| = 3m$, we construct a DESTRUCTIVE-SHIFT-BRIBERY instance $((C,V), p, B, \rho)$ as follows. Let $C = X \cup \mathscr{S} \cup D \cup \{p, w, y\}$ be the candidate set with designated candidate p and a set $D = \{d_{i,j} \mid 1 \leq i \leq m-1, 1 \leq j \leq 4\}$ of dummy candidates. Let $D = D_1 \cup D_2 \cup D_3 \cup D_4$ be a partition of D with $D_j = \{d_{i,j} \mid 1 \leq i \leq m-1\}$ for $1 \leq j \leq 4$. The list V of votes is then constructed as follows:

#	vote	for
1	$\cdots\ p\ S_i$	$1 \leq i \leq 3m$
$4m$	$p\ \cdots\ w\ x_{i,1}\ x_{i,2}\ x_{i,3}\ S_i$	$1 \leq i \leq 3m$
$4m+1$	$\cdots\ p\ X\ d_{i,1}\ d_{i,2}\ d_{i,3}\ d_{i,4}$	$1 \leq i \leq m-1$
1	$p\ \cdots\ y\ x_i$	$1 \leq i \leq 3m$
3	$\cdots\ p$	
2	$p\ \cdots\ w$	

Unlike in the previous proofs, it is here necessary that the candidates that are represented by "$\cdots$" are placed in lexicographical order. For votes of the form $\cdots\ p\ S_i$, we use the price function $\rho(1) = 1$, and $\rho(t) = 2m+1$ for all $t \geq 2$; and for all the remaining voters, we use the price function $\rho(t) = 2m+1$ for all $t \geq 1$. Finally, we set the budget $B = 2m$.

Analyzing the constructed election without bribing voters, the candidates have the following veto counts: p has three vetoes, w has two vetoes, each $x \in X$ has one veto, each $S_i \in \mathscr{S}$ and each $d \in D_4$ has $4m+1$ vetoes, and the remaining candidates each have zero vetoes. It follows that all candidates from $\mathscr{S}$ and D_4 are eliminated. The candidates from D_4 transfer their vetoes to candidates in D_3 who each have $4m+1$ vetoes now; p gets $3m$ additional vetoes from the eliminated candidates in $\mathscr{S}$; and the remaining $12m^2$ vetoes (from the second group of voters) are shared among candidates from X. Since they are ordered lexicographically in those votes, there must be one candidate from X (now and in the following rounds) that obtains more than $4m+1$ vetoes leading to the elimination of all candidates from X in the following rounds. In each of these following rounds, the candidate who receives some of those $12m^2$ vetoes from a previously eliminated candidate (starting with w) will now be eliminated, eventually leaving p as the last standing candidate and sole winner.

We claim that $(X,\mathscr{S})$ is in X3C if and only if $((C,V),p,B,\rho)$ is in Coombs-DESTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

$(\Rightarrow)$ Assume that $(X,\mathscr{S})$ is in X3C. This means that there exists a subset $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = m$ and $\bigcup_{S_i \in \mathscr{S}'} S_i = X$. So we have a partition of X into three sets, $X = X_1 \cup X_2 \cup X_3$, such that:

$$\begin{aligned} X_1 &= \{x_i \in S_i \mid x_i \text{ has the lowest subscript in } S_i \in \mathscr{S}'\}, \\ X_3 &= \{x_i \in S_i \mid x_i \text{ has the highest subscript in } S_i \in \mathscr{S}'\}, \text{ and} \\ X_2 &= X \setminus (X_1 \cup X_3). \end{aligned}$$

We bribe the voters with a vote of the form $\cdots\ p\ S_i$ with $S_i \in \mathscr{S} \setminus \mathscr{S}'$ such that they change their vote to $\cdots\ S_i\ p$. Now the election proceeds as follows, where we again count the vetoes and not the points:

Round	p	w	y	$\mathscr{S}'$	$\mathscr{S}\setminus\mathscr{S}'$	X_1	X_2	X_3	D_1	D_2	D_3	D_4
1	$2m+3$	2	0	$4m+1$	$4m$	1	1	1	0	0	0	$4m+1$
2	$3m+3$	2	0	out	$4m$	1	1	$4m+1$	0	0	$4m+1$	out
3	$3m+3$	2	m	out	$4m$	1	$4m+1$	out	0	$4m+1$	out	out
4	$3m+3$	2	$2m$	out	$4m$	$4m+1$	out	out	$4m+1$	out	out	out
5	$4m^2+2$	$4m^2+2$	$3m$	out	$4m$	out	out	out	out	out	out	out

We see that p is eliminated in the fifth round, whereas y and some other candidates from $\mathscr{S} \setminus \mathscr{S}'$ are still in the election. Hence, p does not win.

$(\Leftarrow)$ Suppose that $(X,\mathscr{S})$ is a no-instance of X3C. Then no subset $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq m$ covers X. We now show that p will not be eliminated in an election obtained by bribing voters without exceeding budget B but will in fact become the only winner. Note that we can only bribe at most $2m$ voters with votes of the form $\cdots\ p\ S_i$ without exceeding the budget. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that for every $S_i \in \mathscr{S} \setminus \mathscr{S}'$ we have bribed the voter whose vote was $\cdots\ p\ S_i$ and now is $\cdots\ S_i\ p$. We can assume that $|\mathscr{S} \setminus \mathscr{S}'| > 0$.

Every candidate in $\mathscr{S} \setminus \mathscr{S}'$ will gain an additional point and therefore survives the first round. All candidates in D_4 and $\mathscr{S}'$ will be eliminated in the first round. It follows that p has $3m+3$ vetoes in the second round. At this point, p is in each voter group other than the third voter group (with votes of the form $\cdots\ p\ X\ d_{i,1}\ d_{i,2}\ d_{i,3}\ d_{i,4}$) either the most (groups 2, 4, and 6) or the least preferred (groups 1 and 5) candidate; therefore, p does not receive any further vetoes before some candidate $d \in D_1$ is eliminated.

We note that $|\mathscr{S}'| \geq m$. Since $\mathscr{S}'$ is not an exact cover of X, we have at least one $x \in X$ which is in two sets $S, S' \in \mathscr{S}'$. Let $X' = \{x \in X \mid \exists\, S, S' \in \mathscr{S}',\ S \neq S',\ x \in S \cap S'\}$. After two further rounds in which $4m+1$ is the maximum number of vetoes, the candidates $d \in D \setminus D_1$ are eliminated. If each $x \in X'$ is still in the election, it follows that each $x \in X'$ has at least $4m+2$ vetoes such that some candidates $x \in X'$ will be eliminated. It follows that in the next round w receives at least $4m+2$ vetoes such that w has the most vetoes while the candidates $d \in D_1$ still have $4m+1$ vetoes. Otherwise, if at least one candidate $x \in X'$ is eliminated, it follows that w receives at least $4m+2$ vetoes at the latest in the fourth round, while each $d \in D_1$ still has $4m+1$ vetoes. After w is eliminated, in each following round the candidate x with the highest subscript and later the candidate S with the highest subscript and y will be eliminated. It follows that only p and the candidates $d \in D_1$ are still in the election. In each following round, p has at most $4m^2 - 4m + 1$ vetoes while the still standing candidate $d \in D_1$ with the highest subscript receives at least $12m^2 + 7m + 3$ vetoes. Hence, eventually p alone wins the election. $\square$

4. Baldwin and Nanson

We now show NP-hardness of shift bribery for Baldwin and Nanson elections. Note that our reductions are inspired by and similar to those used by Davies et al. (2014) to show NP-hardness of the unweighted coalitional manipulation problem for these voting systems.

For a preference profile V over a set of candidates C, let $avg(V)$ be the average Borda score of the candidates in V (i.e., $avg(V) = (|C|-1)|V|/2$). To conveniently construct votes, for a set of candidates C and $c_1, c_2 \in C$, let

$$W_{(c_1,c_2)} = (c_1\ c_2\ \overrightarrow{C \setminus \{c_1,c_2\}},\ \overleftarrow{C \setminus \{c_1,c_2\}}\ c_1\ c_2).$$

Under Borda, from the two votes in $W_{(c_1,c_2)}$ candidate c_1 scores $|C|$ points, c_2 scores $|C|-2$ points, and all other candidates score $|C|-1$ points. Also, observe that if a candidate $c^* \in C$ is eliminated in some round and $c^* \notin \{c_1,c_2\}$ then all other candidates lose one point due to the votes in $W_{(c_1,c_2)}$; if $c^* = c_1$ then c_2 loses no points but all other candidates lose one point; and if $c^* = c_2$ then c_1 loses two points and all other candidates lose one point. Therefore, if c^* is eliminated, the point difference caused by this elimination with respect to the votes in $W_{(c_1,c_2)}$ remains the same for all candidates, with two exceptions: (a) If $c^* = c_1$ then c_2 gains a point with respect to every other candidate, and (b) if $c^* = c_2$ then c_1 loses a point with respect to every other candidate. Furthermore, let $score_{(C,V)}(x)$ denote the number of points candidate x obtains in a Borda election (C,V), and let $dist_{(C,V)}(x,y) = score_{(C,V)}(x) - score_{(C,V)}(y)$.

We start with the complexity of shift bribery in Baldwin elections for the constructive case.

Theorem 5. *In both the unique-winner and the nonunique-winner model, Baldwin-*CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we reduce the NP-complete problem X3C to Baldwin-CONSTRUCTIVE-SHIFT-BRIBERY. From a given X3C instance $(X, \mathscr{S})$, we construct an election (C,V) with the set of candidates $C = \{p,w,d\} \cup X \cup \mathscr{S}$ and designated candidate p and with V consisting of two lists of votes, V_1 and V_2, where V_1 contains the following votes:

#	votes	for	#	votes	for
1	$W_{(S_j,p)}$	$1 \le j \le 3m$	2	$W_{(x_{j,3},S_j)}$	$1 \le j \le 3m$
2	$W_{(x_{j,1},S_j)}$	$1 \le j \le 3m$	2	$W_{(w,x_i)}$	$1 \le i \le 3m$
2	$W_{(x_{j,2},S_j)}$	$1 \le j \le 3m$	7	$W_{(w,p)}$	

The votes in V_1 give the following scores to the candidates in C:

$$\begin{aligned}
score_{(C,V_1)}(x_i) &= avg(V_1) + 4 \text{ for every } x_i \in X,\\
score_{(C,V_1)}(S_j) &= avg(V_1) - 5 \text{ for every } S_j \in \mathscr{S},\\
score_{(C,V_1)}(p) &= avg(V_1) - 3m - 7,\\
score_{(C,V_1)}(w) &= avg(V_1) + 6m + 7,\\
score_{(C,V_1)}(d) &= avg(V_1).
\end{aligned}$$

Furthermore, V_2 contains the following votes:

#	votes	for	#	votes
$2m+1$	$W_{(d,S_j)}$	$1 \le j \le 3m$	1	$W_{(p,d)}$
$2m+9$	$W_{(d,x_i)}$	$1 \le i \le 3m$	$2m+14$	$W_{(d,w)}$

The votes in V_2 give the following scores to the candidates in C:

$$\begin{aligned}
score_{(C,V_2)}(x_i) &= avg(V_2) - (2m+9) \text{ for every } x_i \in X,\\
score_{(C,V_2)}(S_j) &= avg(V_2) - (2m+1) \text{ for every } S_j \in \mathscr{S},\\
score_{(C,V_2)}(p) &= avg(V_2) + 1,\\
score_{(C,V_2)}(w) &= avg(V_2) - (2m+14),\\
score_{(C,V_2)}(d) &= avg(V_2) + 12m^2 + 32m + 13.
\end{aligned}$$

Let $V = V_1 \cup V_2$ and $avg(V) = avg(V_1) + avg(V_2)$. Then we have the following Borda scores for the complete preference profile V over C:

$$\begin{aligned}
score_{(C,V)}(x_i) &= avg(V) - 2m - 5 \text{ for every } x_i \in X,\\
score_{(C,V)}(S_j) &= avg(V) - 2m - 6 \text{ for every } S_j \in \mathscr{S},\\
score_{(C,V)}(p) &= avg(V) - 3m - 6,\\
score_{(C,V)}(w) &= avg(V) + 4m - 7,\\
score_{(C,V)}(d) &= avg(V) + 12m^2 + 32m + 13.
\end{aligned}$$

Regarding the price function, for every first vote of $W_{(S_j,p)}$ (i.e., a vote of the form $S_j\ p\ \overrightarrow{C \setminus \{S_j,p\}}$), let $\rho(1) = 1$ and $\rho(t) = m+1$ for every $t \geq 2$. For every other vote, let $\rho(t) = m+1$ for every $t \geq 1$. Finally, we set the budget $B = m$.

It is easy to see that p is eliminated in the first round in the election (C,V) and thus does not win.

We claim that $(X,\mathscr{S})$ is in X3C if and only if $((C,V),p,B,\rho)$ is in Baldwin-CONSTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

($\Rightarrow$) Suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Then we bribe the first votes of $W_{(S_j,p)}$ for every $S_j \in \mathscr{S}'$ by shifting p to the left once. Note that we won't exceed our budget, since shifting once costs 1 in those votes and $|\mathscr{S}'| = m$. After this bribery, for every $S_j \in \mathscr{S}'$, the two votes from $W_{(S_j,p)}$ result in two votes that are symmetric to each other (i.e., $p\ S_j\ \overrightarrow{C \setminus \{S_j,p\}}$ equals the vote $\overleftarrow{C \setminus \{S_j,p\}}\ S_j\ p$ in reverse order) and can thus be disregarded from now on, as all candidates gain the same number of points from those votes and all candidates lose the same number of points if a candidate is eliminated from the election. After those m votes have been bribed, only the scores of p and every $S_j \in \mathscr{S}'$ change. With $score_{(C,V)}(p) = avg(V) - 2m - 6$ and $score_{(C,V)}(S_j) = avg(V) - 2m - 7$, all candidates in $\mathscr{S}'$ are tied for the last place. If any $S_j \in \mathscr{S}'$ is eliminated in a round, the three candidates $x_{j,1}$, $x_{j,2}$, and $x_{j,3}$ will lose two points more than the candidates from $\mathscr{S}' \setminus \{S_j\}$ that were in the last position before S_j was eliminated. Therefore, those three candidates from X will then be in the last position in the next round. This means that all candidates $\mathscr{S}'$ and every $x_i \in X$ that is covered by $\mathscr{S}'$ will be eliminated in the subsequent rounds. Since $\mathscr{S}'$ is an exact cover, now there is no candidate from X left. Thus the point difference between p and w is 1 and between p and the remaining $S_j \in (\mathscr{S} \setminus \mathscr{S}')$ is -6. Note that p can beat d only if no candidate of $C \setminus \{p,d\}$ is still participating. So in the next round, w is eliminated. From this p gains seven points on all $S_j \in (\mathscr{S} \setminus \mathscr{S}')$, so these are tied for the last place. Therefore, the remaining candidates from $\mathscr{S}$ are eliminated, which leaves p and d for the next and final round, where d is eliminated and p wins the election alone.

($\Leftarrow$) Suppose there is no exact cover. It is obvious that at most m of the first votes of $W_{(S_j,p)}$ can be bribed without exceeding the budget. Without bribing, p is in the last place and the point difference to the second-to-last candidate(s) is $dist_{(C,V)}(p,S_j) = m$, $1 \leq j \leq 3m$. By bribing, as explained above, p gains $m+1$ points on m candidates from $\mathscr{S}$, which then will be eliminated from the election. This leads to the elimination of all $x_i \in X$ that are covered by the set $\mathscr{S}' \subseteq \mathscr{S}$ of candidates that were eliminated. Since there is no exact cover, $\mathscr{S}'$ doesn't cover X. So there are candidates $X' \subseteq X$, $|X'| \geq 1$, who were not eliminated before, as for every candidate $x_i \in X'$ all three candidates $S_j \in (\mathscr{S} \setminus \mathscr{S}')$ with $x_i \in S_j$ are still in the election. With the candidates $C_1 = \{p,w,d\} \cup (\mathscr{S} \setminus \mathscr{S}') \cup X'$ still standing, the point differences of p to the other remaining candidates are as follows:

$$\begin{aligned}
dist_{(C_1,V)}(p,d) &= -2m-5-2m(2m+1)-|X'|(2m+9)-(2m+14) < 0,\\
dist_{(C_1,V)}(p,w) &= 1-2|X'| < 0,\\
dist_{(C_1,V)}(p,x_i) &= -1 \text{ for every } x_i \in X', \text{ and}\\
dist_{(C_1,V)}(p,S_j) &\leq 0 \text{ for every } S_j \in \mathscr{S} \setminus \mathscr{S}'.
\end{aligned}$$

Therefore, p is in the last place and is eliminated and thus does not win. □

The proof of the following theorem, which handles the destructive variant for Baldwin, uses a similar idea as the proof of Theorem 5. That is why we refrain from presenting all proof details in full; a proof sketch will suffice.

Theorem 6. *In both the unique-winner and the nonunique-winner model, Baldwin-*DESTRUCTIVE-SHIFT-BRIBERY *is* NP-*hard.*

Proof Sketch. To prove NP-hardness, we reduce the NP-complete problem X3C to Baldwin-DESTRUCTIVE-SHIFT-BRIBERY. From a given X3C instance $(X,\mathscr{S})$, we construct an election (C,V), where $C = \{p,w,b,d\} \cup X \cup \mathscr{S}$ is the set of candidates, p is the designated candidate, and V consists of two lists of votes, V_1 and V_2, where V_1 contains the following votes:

#	votes	for	#	votes	for
1	$W_{(p,S_j)}$	$1 \leq j \leq 3m$	2	$W_{(w,x_i)}$	$1 \leq i \leq 3m$
2	$W_{(S_j,x_{j,1})}$	$1 \leq j \leq 3m$	$3m+7$	$W_{(w,d)}$	
2	$W_{(S_j,x_{j,2})}$	$1 \leq j \leq 3m$	$m+10$	$W_{(b,S_j)}$	$1 \leq j \leq 3m$
2	$W_{(S_j,x_{j,3})}$	$1 \leq j \leq 3m$			

Furthermore, V_2 contains the following votes:

#	votes	for	#	votes
1	$W_{(d,p)}$		$6m+14$	$W_{(p,w)}$
$2m+7$	$W_{(p,S_j)}$	$1 \leq j \leq 3m$	$3m^2+33m+12$	$W_{(p,b)}$
$3m+3$	$W_{(p,x_i)}$	$1 \leq i \leq 3m$		

Let $V = V_1 \cup V_2$. Then we have the following Borda scores for the complete profile V:

$$\begin{aligned}
score_{(C,V)}(x_i) &= avg(V) - 3m - 11 \text{ for every } x_i \in X,\\
score_{(C,V)}(S_j) &= avg(V) - 3m - 12 \text{ for every } S_j \in \mathscr{S},\\
score_{(C,V)}(d) &= avg(V) - 3m - 6,\\
score_{(C,V)}(w) &= avg(V) + 3m - 7,\\
score_{(C,V)}(b) &= avg(V) - 3m - 12,\\
score_{(C,V)}(p) &= avg(V) + 18m^2 + 72m + 25.
\end{aligned}$$

Regarding the price function, for every first vote of $W_{(p,S_j)}$ (i.e., a vote of the form $p\ S_j\ C \setminus \{S_j, p\}$), let $\rho(1) = 1$ and $\rho(t) = m+1$ for every $t \geq 2$. For every other vote, let $\rho(t) = m+1$ for every $t \geq 1$. Finally, we set the budget $B = m$.

It is easy to see that p wins the election (C,V).

We claim that $(X, \mathscr{S})$ is in X3C if and only if $((C,V), p, B, \rho)$ is in Baldwin-DESTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

($\Rightarrow$) Suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Then we bribe the first votes of $W_{(p,S_j)}$ for every $S_j \in \mathscr{S}'$ by shifting p to the right once. With a similar argument as in the proof of Theorem 5, d alone wins the election, i.e., p is not among the winners.

($\Leftarrow$) Suppose there is no exact cover. Then, for every $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq m$, there is at least one $x_i \in X$ that is not covered by $\mathscr{S}'$. It is obvious that at most m of the first votes of $W_{(p,S_j)}$ can be bribed without exceeding the budget. We can then show, similarly as in the proof of Theorem 5, that d will always be eliminated before w and therefore p cannot be prevented from winning the election alone. $\square$

Finally, we turn to Nanson elections for which we again will show that shift bribery is NP-hard. The reduction below will only use pairs of votes of the form $W_{(c_1,c_2)}$. The average Borda score for those two votes is $|C|-1$. The candidate c_1 scores one point more than the average Borda score and c_2 scores one point fewer than the average Borda score. The other candidates score exactly the average Borda score. If a candidate is eliminated in a round, the average Borda score required to survive the next round decreases by one. Regardless of which candidate is eliminated, all remaining candidates that are not c_1 or c_2 lose one point and still have exactly the average Borda score. If c_2 is eliminated, c_1 loses its advantage with respect to the average Borda score and now scores exactly the average Borda score as well. If one of the other candidates is eliminated, c_1 continues to have one point more than the average Borda score. By symmetry, this holds analogously for c_2: If c_1 is eliminated, c_2 scores the average Borda score, and if one of the other candidates is eliminated, c_2 still has one point fewer than the average Borda score.

Theorem 7. *In both the unique-winner and the nonunique-winner model, Nanson-*CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we reduce the NP-complete problem X3C to Nanson-CONSTRUCTIVE-SHIFT-BRIBERY. Again, starting from a given X3C instance $(X, \mathscr{S})$, we construct an election (C,V) with the set of candidates $C = \{p, w_1, w_2, d\} \cup X \cup \mathscr{S}$, where p is the designated candidate. Then we construct two sets of votes, V_1 and V_2, where V_1 contains the following votes:

#	votes	for	#	votes	for
1	$W_{(S_j,p)}$	$1 \leq j \leq 3m$	1	$W_{(x_{j,3},S_j)}$	$1 \leq j \leq 3m$
1	$W_{(x_i,p)}$	$1 \leq i \leq 3m$	4	$W_{(S_j,w_1)}$	$1 \leq j \leq 3m$
1	$W_{(x_{j,1},S_j)}$	$1 \leq j \leq 3m$	$15m$	$W_{(w_1,w_2)}$	
1	$W_{(x_{j,2},S_j)}$	$1 \leq j \leq 3m$	$3m$	$W_{(p,w_1)}$	

Furthermore, V_2 contains the following votes:

#	votes	for
$2m$	$W_{(p,d)}$	
2	$W_{(d,S_j)}$	$1 \leq j \leq 3m$
4	$W_{(d,x_i)}$	$1 \leq i \leq 3m$

Let $V = V_1 \cup V_2$. Then we have the following Borda scores for the complete profile V:

$$\begin{aligned}
score_{(C,V)}(x_i) &= avg(V) \text{ for every } x_i \in X, \\
score_{(C,V)}(S_j) &= avg(V) \text{ for every } S_j \in \mathscr{S}, \\
score_{(C,V)}(p) &= avg(V) - m, \\
score_{(C,V)}(w_1) &= avg(V), \\
score_{(C,V)}(w_2) &= avg(V) - 15m, \\
score_{(C,V)}(d) &= avg(V) + 16m.
\end{aligned}$$

The price function is again defined as follows. For every first vote of $W_{(S_j,p)}$ (i.e., a vote of the form $S_j\ p\ C \setminus \{S_j, p\}$), let $\rho(1) = 1$ and $\rho(t) = m+1$ for every $t \geq 2$. For every other vote, let $\rho(t) = m+1$ for every $t \geq 1$. Finally, we set the budget $B = m$.

It is easy to see that p is eliminated in the first round of the election (C,V) and so does not win.

We claim that $(X, \mathscr{S})$ is in X3C if and only if $((C,V), p, B, \rho)$ is in Nanson-CONSTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

($\Rightarrow$) Suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Then, for every $S_j \in \mathscr{S}'$, we bribe the first vote of $W_{(S_j,p)}$ by shifting p to the left once in all those votes. Note that we won't exceed our budget, since this bribe action costs 1 per vote and $|\mathscr{S}'| = m$. With the additional m points, p reaches the average Borda score and is not eliminated in the first round. However, all candidates in $\mathscr{S}'$ lose one point and are eliminated. Additionally, w_2 will be eliminated as well.

In the next round, w_1 will be eliminated, since she has $11m$ points fewer than the average Borda score required to survive this round. Since the candidates in $\mathscr{S}'$ were eliminated in the last round and $\mathscr{S}'$ is an exact cover, every candidate in X now has fewer points than the average Borda score and is eliminated.

In the third round, only p, d, and the candidates in $\mathscr{S} \setminus \mathscr{S}'$ are still standing. Therefore, the only pairs of votes that are not symmetric are $W_{(S_j,p)}$, twice $W_{(d,S_j)}$ for every $S_j \in (\mathscr{S} \setminus \mathscr{S}')$, and $2m$ pairs of $W_{(p,d)}$. Since $|\mathscr{S} \setminus \mathscr{S}'| = 2m$, we have that p scores exactly the average Borda score and survives this round, just as d. Every $S_j \in (\mathscr{S} \setminus \mathscr{S}')$ has one point fewer than the average Borda score and is eliminated. This leaves only p and d in the last round, which p alone wins.

($\Leftarrow$) Suppose there is no exact cover. Then, for every $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = m$, there is at least one $x_i \in X$ that is not covered by $\mathscr{S}'$. Note that we can only bribe the first votes of any $W_{(S_j,p)}$ without exceeding the budget. For p to survive the first round, we need to bribe m of those votes by shifting p to the left once. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that $\mathscr{S}'$ contains S_j exactly if the first vote of $W_{(S_j,p)}$ has been bribed. Then every $S_j \in \mathscr{S}'$ has a score of $avg(V) - 1$ and p has a score of $avg(V)$. Therefore, in the first round, every candidate from $\mathscr{S}'$ and w_2 are eliminated from the election.

In the second round, w_1 will be eliminated because of the $15m$ pairs of votes $W_{(w_1,w_2)}$ and the elimination of w_2. Furthermore, a candidate $x_i \in X$ reaches the average Borda score with p and d still standing only if all three $S_j \in \mathscr{S}$ with $x_i \in S_j$ are also not yet eliminated. Since the candidates in $\mathscr{S}'$ were eliminated in the previous round, for every $S_j \in \mathscr{S}'$, all three $x_i \in S_j$ will be eliminated in this round. Since $\mathscr{S}'$ is not an exact cover, there are candidates $X' \subseteq X$ that survive this round. d also reaches the average Borda score, as there are $2m$ candidates $\mathscr{S} \setminus \mathscr{S}'$ and those candidates $\mathscr{S} \setminus \mathscr{S}'$ survive due to w_1.

In the next round, the candidates still standing are p, d, X', and $\mathscr{S} \setminus \mathscr{S}'$. Because $|X'| \geq 1$, candidate p has $|X'|$ points fewer than the average Borda score and is eliminated in this round. Thus p does not win. $\square$

Our last result in this section shows that the destructive variant of shift bribery in Nanson elections is intractable as well.

Theorem 8. *In both the unique-winner and the nonunique-winner model, Nanson-*DESTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we reduce the NP-complete problem X3C to Nanson-DESTRUCTIVE-SHIFT-BRIBERY. Once more, given an X3C instance $(X, \mathscr{S})$, we construct an election (C,V) with the set of candidates $C = \{p, w_1, w_2, w_3, d\} \cup X \cup \mathscr{S}$, where p is the designated candidate and $(X, \mathscr{S})$ is the given X3C instance. Then we construct two sets of votes, V_1 and V_2, where V_1 contains the following votes:

#	votes	for	#	votes	for
1	$W_{(p,S_j)}$	$1 \leq j \leq 3m$	6	$W_{(S_j,w_3)}$	$1 \leq j \leq 3m$
1	$W_{(d,x_i)}$	$1 \leq i \leq 3m$	$20m$	$W_{(w_1,w_2)}$	
2	$W_{(x_{j,1},S_j)}$	$1 \leq j \leq 3m$	$19m$	$W_{(w_3,w_1)}$	
2	$W_{(x_{j,2},S_j)}$	$1 \leq j \leq 3m$	$3m+1$	$W_{(w_3,d)}$	
2	$W_{(x_{j,3},S_j)}$	$1 \leq j \leq 3m$			

Furthermore, V_2 contains the following votes:

#	votes	for	#	votes
1	$W_{(d,p)}$		$3m+1$	$W_{(p,w_3)}$
1	$W_{(p,x_i)}$	$1 \leq i \leq 3m$		

Let $V = V_1 \cup V_2$. Then we have the following Borda scores for the complete profile V:

$$\begin{aligned}
score_{(C,V)}(x_i) &= avg(V) + 4 \text{ for every } x_i \in X,\\
score_{(C,V)}(S_j) &= avg(V) - 1 \text{ for every } S_j \in \mathscr{S},\\
score_{(C,V)}(d) &= avg(V),\\
score_{(C,V)}(w_1) &= avg(V) + m,\\
score_{(C,V)}(w_2) &= avg(V) - 20m,\\
score_{(C,V)}(w_3) &= avg(V) + m,\\
score_{(C,V)}(p) &= avg(V) + 9m.
\end{aligned}$$

The price function is again defined as follows. For every first vote of $W_{(p,S_j)}$ (i.e., a vote of the form $p\ S_j\ C \setminus \{S_j, p\}$), let $\rho(1) = 1$ and $\rho(t) = m+1$ for every $t \geq 2$. For every other vote, let $\rho(t) = m+1$ for every $t \geq 1$. Finally, we set the budget $B = m$.

It is easy to see that p will only have fewer points than the average Borda score if all candidates from $\mathscr{S}$, X, and the candidate w_3 are eliminated while d is still standing. Without bribing, d is eliminated in the third round while w_3 is still standing, and eventually p wins the election (C,V).

We claim that $(X, \mathscr{S})$ is in X3C if and only if $((C,V), p, B, \rho)$ is in Nanson-DESTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model.

($\Rightarrow$) Suppose there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$. Then, for every $S_j \in \mathscr{S}'$, we bribe the first vote of $W_{(p,S_j)}$ by shifting p to the right once in all those votes. Note that we won't exceed our budget, since this bribe action costs 1 per vote and $|S'| = m$. After those m votes have been bribed, every $S_j \in \mathscr{S}'$ gains a point and therefore survives the first round. All other candidates $\mathscr{S} \setminus \mathscr{S}'$ and w_2 are eliminated.

Let $C_1 = \{p, d, w_1, w_3\} \cup X \cup \mathscr{S}'$ be the set of candidates present in the second round. w_1 loses $19m$ points on the average Borda score from the elimination of w_2 and is eliminated. Additionally, all candidates of X lose four points on the average Borda score but still survive this round, as they now have exactly the average Borda score.

Let $C_2 = \{p, d, w_3\} \cup X \cup \mathscr{S}'$ be the candidates in the third round. In this round, only w_3 is eliminated because w_3 lost $19m$ points on the average Borda score from the elimination of w_1.

Let $C_3 = \{p, d\} \cup X \cup \mathscr{S}'$ be the candidates in the fourth round. The scores are as follows:

$$\begin{aligned}
score_{(C_3,V)}(x_i) &= avg(V) \text{ for every } x_i \in X, \\
score_{(C_3,V)}(S_j) &= avg(V) - 6 \text{ for every } S_j \in \mathscr{S}', \\
score_{(C_3,V)}(d) &= avg(V) + 3m + 1, \\
score_{(C_3,V)}(p) &= avg(V) + 3m - 1.
\end{aligned}$$

Therefore all candidates in $\mathscr{S}'$ are eliminated. In the following round, all candidates in X are eliminated. This leaves only p and d in the final round in which p is eliminated and thus cannot win.

($\Leftarrow$) Suppose there is no exact cover. Then, for every $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq m$, there is at least one $x_i \in X$ that is not covered by $\mathscr{S}'$. Note that we can only bribe the first votes of any $W_{(p,S_j)}$ without exceeding the budget.

We now show that, even with optimal bribing, d will be eliminated in the third round and, therefore, p alone wins the election. Within our budget, we can prevent at most m candidates from $\mathscr{S}$, say $\mathscr{S}'$, of being eliminated in the first round by bribing the corresponding vote of $W_{(p,S_j)}$. Since $\mathscr{S}'$ cannot be an exact cover of X, there is at least one $x_i \in X$ for which all $S_j \in \mathscr{S}$ with $x_i \in S_j$ are eliminated. This x_i is eliminated in the second round, as it has lost six points on the average Borda score from the eliminations of candidates in the previous round. In the third round, w_3 is still participating since w_2 and w_1 were only eliminated in the first and second round, respectively. Therefore, the score of d minus the average Borda score of this round is at most -1, which means that d is eliminated in this round. Thus, there is no candidate left that can prevent p from winning the election. $\square$

5. Iterated Plurality and Plurality with Runoff

In this section, we show hardness of shift bribery for iterated plurality and plurality with runoff, handling both voting systems simultaneously and starting with the constructive case.

Theorem 9. *In both the unique-winner and the nonunique-winner model, for iterated plurality and plurality with runoff,* CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we reduce X3C to CONSTRUCTIVE-SHIFT-BRIBERY for these two voting systems. Let $(X, \mathscr{S})$ be a given X3C instance. We construct the CONSTRUCTIVE-SHIFT-BRIBERY instance $((C,V), p, B, \rho)$ as follows. Let $C = \{p, w\} \cup X \cup \mathscr{S} \cup D$ be the set of candidates, where p is the designated candidate and $D = \{d_{i,j} \mid 1 \leq i \leq 3m \text{ and } 1 \leq j \leq m-7\}$ is a set of dummy candidates. The list V of votes is constructed as follows:

#	vote	for
1	$S_i\ p\ \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,1}\ \overrightarrow{X \setminus \{x_{i,1}\}} \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,2}\ \overrightarrow{X \setminus \{x_{i,2}\}} \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,3}\ \overrightarrow{X \setminus \{x_{i,3}\}}\ \cdots$	$1 \leq i \leq 3m$
1	$S_i\ d_{i,j}\ \overrightarrow{X \setminus \{x_i\}}\ \cdots$	$1 \leq i \leq 3m$, $1 \leq j \leq m-7$
m	$x_i\ \overrightarrow{X \setminus \{x_i\}}\ \cdots$	$1 \leq i \leq 3m$
m	$d_{i,j}\ \overrightarrow{X}\ \cdots$	$1 \leq i \leq 3m$, $1 \leq j \leq m-7$
3	$w\ p\ \cdots$	

For voters with votes of the form $S_i\ p\ \cdots$, we use the price function $\rho(1) = 1$, and $\rho(t) = m+1$ for all $t \geq 2$; and for every other voter, we use the price function $\rho(t) = m+1$ for $t \geq 1$. Finally, set the budget $B = m$.

Without bribing, p has a score of zero and is eliminated immediately in both voting systems.

We claim that $(X, \mathscr{S})$ is in X3C if and only if $((C,V), p, B, \rho)$ is in CONSTRUCTIVE-SHIFT-BRIBERY for either of the two voting systems, regardless of the winner model.

$(\Rightarrow)$ Suppose that $(X, \mathscr{S})$ is a yes-instance of X3C. Then there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of size m. We now show that it is possible for p to become a unique iterated-plurality (respectively, plurality-with-runoff) winner of an election obtained by shifting p in the votes without exceeding the budget. For every $S_i \in \mathscr{S}'$, we bribe the voter with the vote of the form $S_i\ p\ \cdots$, so her new vote is of the form $p\ S_i\ \cdots$. In the first round p, every $x_i \in X$, every $d_{i,j} \in D$, and every $S_i \in \mathscr{S} \setminus \mathscr{S}'$ is a plurality winner, so only these candidates participate in the next round. In the second round, p receives three further points from the three voters whose vote is $w\ p\ \cdots$. Every candidate $x_j \in X$ receives two further points from the votes of the form $S_i\ x_j\ \cdots$ with $x_j \in S_i$ and $S_i \in \mathscr{S}'$. Every $d_{i,j}$ with $S_i \in \mathscr{S}'$ and $1 \leq j \leq m-7$ receives one additional point from the voters with vote $S_i\ d_{i,j}\ \cdots$. It follows that p has the most points and therefore p is the unique iterated-plurality (respectively, plurality-with-runoff) winner.

$(\Leftarrow)$ Suppose that $(X, \mathscr{S})$ is a no-instance of X3C. Then, for every $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = m$, there is at least one candidate in X that is not covered and, therefore, at least one candidate in X occurring in at least two sets from $\mathscr{S}'$. We show that it is not possible for p to become a winner of the election obtained from the original election by bribing without exceeding the budget.

To become a winner of such a bribed election, it is necessary for p to get at least m points in the first round. Due to the budget, it is also necessary to bribe m voters with a vote of the form $S_i\ p\ \cdots$ with $S_i \in \mathscr{S}'$. It follows that p, each $x \in X$, each $S_i \in \mathscr{S} \setminus \mathscr{S}'$, and each $d_{i,j} \in D$ participate in the second round. As mentioned above, at least one candidate in X receives at least four further points due to the fact that $\mathscr{S}'$ is not a cover of X. Thus p does not win. That means that $((C,V), p, B, \rho)$ is a no-instance of CONSTRUCTIVE-SHIFT-BRIBERY for either of iterated plurality and plurality with runoff regardless of the winner model. □

We have the same result in the destructive case. This is the first proof where we use an NP-complete problem other than X3C to show NP-hardness, namely ONE-IN-THREE-POSITIVE-3SAT, which was also defined in Section 2.

Theorem 10. *In both the unique-winner and the nonunique-winner model, for iterated plurality and plurality with runoff,* DESTRUCTIVE-SHIFT-BRIBERY *is* NP-*hard.*

Proof. To prove NP-hardness, we reduce the NP-complete problem ONE-IN-THREE-POSITIVE-3SAT to DESTRUCTIVE-SHIFT-BRIBERY for both voting systems. Let (X,S) be a given ONE-IN-THREE-POSITIVE-3SAT instance, where $X = \{x_1,\dots,x_{3m}\}$ and $S = \{S_1,\dots S_{3m}\}$ with $S_i = \{x_{i,1},x_{i,2},x_{i,3}\} \subseteq X$ for each $1 \le i \le 3m$. Without loss of generality, we can assume that $m > 6$. We construct the DESTRUCTIVE-SHIFT-BRIBERY instance for both voting systems as follows. Let $C = \{p,w,e,f\} \cup D \cup Y \cup X$ with $D = \{d_{i,j} \mid 1 \le i \le 3m \text{ and } 1 \le j \le 2m-1\}$ and $Y = \{y_{i,j} \mid 1 \le i \le 3m \text{ and } 1 \le j \le 4\}$ and where p is the designated candidate. The list V of votes is constructed as follows:

#	votes	for
1	$p\ x_i\ \cdots$	$1 \le i \le 3m$
1	$y_{i,1}\ x_{i,1}\ x_{i,2}\ w\ p\ \cdots$	$1 \le i \le 3m$
1	$y_{i,2}\ x_{i,2}\ x_{i,3}\ w\ p\ \cdots$	$1 \le i \le 3m$
1	$y_{i,3}\ x_{i,1}\ x_{i,3}\ w\ p\ \cdots$	$1 \le i \le 3m$
4	$y_{i,4}\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p\ \cdots$	$1 \le i \le 3m$
1	$x_i\ d_{i,j}\ p\ \cdots$	$1 \le i \le 3m$, $1 \le j \le 2m-1$
$2m$	$d_{i,j}\ p\ \cdots$	$1 \le i \le 3m$, $1 \le j \le 2m-1$
$2m$	$w\ p\ \cdots$	
$2m-1$	$e\ p\ \cdots$	
m	$f\ p\ \cdots$	

For votes of the form $p\ x_i\ \cdots$ we use the price function $\rho(1) = 1$ and $p(t) = m+1$ for all $t \ge 2$. For every other vote, we use the price function $\rho(t) = m+1$ for $t \ge 1$. Finally, set the budget $B = m$.

Without bribing, the election proceeds as follows. In the first round, p scores $3m$ points, w and every $d_{i,j} \in D$ scores $2m$ points, and each of the remaining candidates scores fewer than $2m$ points. In the second round, p scores $18m-1$ points, w scores $11m$ points, and every $d_{i,j}$ scores $2m+1$ points. It follows that p is the unique winner for either of iterated plurality and plurality with runoff.

We claim that (X,S) is in ONE-IN-THREE-POSITIVE-3SAT if and only if $((C,V),p,B,\rho)$ is in DESTRUCTIVE-SHIFT-BRIBERY for either of the two voting systems, regardless of the winner model.

($\Rightarrow$) Suppose that (X,S) is a yes-instance of ONE-IN-THREE-POSITIVE-3SAT. Then there exists a subset $U \subseteq X$ such that for each clause S_j we have $|U \cap S_j| = 1$. We bribe the voters with the vote of the form $p\ x_i\ \cdots$ with $x_i \in U$ so that the new vote has the form $x_i\ p\ \cdots$. It follows that p, w, every $x_i \in U$, and every $d_{i,j} \in D$ reach the second round with $2m$ points each. In the second round, p gains $3m-1$ additional points while w gains $3m$ additional points. It follows that p is not a winner of the election, so $((C,V),p,B,\rho)$ is a yes-instance of DESTRUCTIVE-SHIFT-BRIBERY for both voting systems, regardless of the winner model.

($\Leftarrow$) Suppose that (X,S) is a no-instance of ONE-IN-THREE-POSITIVE-3SAT. We show that $((C,V),$ $p,B,\rho)$ is also a no-instance of DESTRUCTIVE-SHIFT-BRIBERY for both voting systems. To ensure that p is not the only plurality winner in the first round, it is necessary to bribe m voters with votes of the form $p\ x_i\ \cdots$ to now vote $x_i\ p\ \cdots$. Note that we can only bribe at most m such voters without exceeding the budget. Let $U \subseteq X$ be the set of candidates that benefit from the bribery action. It follows that p, every $d_{i,j} \in D$, every $x_i \in U$, and w can move forward to the next round with $2m$ points each. In this round, the designated candidate p gains $3m-1$ additional points from the votes of the form $e\ p\ \cdots$ and $f\ p\ \cdots$; every candidate $d_{i,j}$ with $x_i \notin U$ gains one additional point; every candidate $x_i \in U$ can receive at most 18 additional points; and w is discussed separately in the following paragraph.

To prevent the victory of p, it is necessary that w gains at least $3m$ points (since if w gains only $3m-1$ points, it follows that w and p move forward to the final round, where p would achieve a clear victory). For w to gain at least one point from any one of the three votes of the form $y_{i,1}\ x_{i,1}\ x_{i,2}\ w\ p\ \cdots$, $y_{i,2}\ x_{i,2}\ x_{i,3}\ w\ p\ \cdots$, and $y_{i,3}\ x_{i,1}\ x_{i,3}\ w\ p\ \cdots$, it is necessary that at most one candidate $x_{i,j}$ participates in the second round. On the other hand, if no candidate $x_{i,j}$ participates in the second round, p gains four points from the voters of the fifth line, whose vote is $y_{i,4}\ x_{i,1}\ x_{i,2}\ x_{i,3}\ p\ \cdots$, i.e., this clause harms w. Only a clause S_i with $|S_i \cap U| = 1$ helps w to reduce the point difference to p. Since (X,S) is a no-instance of ONE-IN-THREE-POSITIVE-3SAT, there are at most $3m-2$ clauses with this property.

With these clauses w can reduce the point difference to two. With the two remaining clauses the point difference is growing. This implies that p is always a unique winner of the election, i.e., $((C,V),p,B,\rho)$ is a no-instance of DESTRUCTIVE-SHIFT-BRIBERY for both voting systems, regardless of the winner model. □

6. Iterated Veto and Veto with Runoff

In this section, we show hardness of shift bribery for iterated veto and veto with runoff, again handling both voting systems simultaneously and starting with the constructive case.

Theorem 11. *In both the unique-winner and the nonunique-winner model, for veto with runoff and iterated veto,* CONSTRUCTIVE-SHIFT-BRIBERY *is* NP-*hard.*

Proof. To prove NP-hardness, we reduce X3C to CONSTRUCTIVE-SHIFT-BRIBERY for veto with runoff and iterated veto at the same time. Let $(X,\mathscr{S})$ be a given X3C instance and construct the CONSTRUCTIVE-SHIFT-BRIBERY instance $((C,V),p,B,\rho)$ as follows. Let $C = \{p,d_1,d_2\} \cup X \cup \mathscr{S}$ be the set of candidates, where p is the designated candidate, and construct the voter preferences in V as follows:

#	votes	for
1	$\cdots\ S_i\ p$	$1 \le i \le 3m$
2	$\cdots\ x_{i,1}\ S_i$	$1 \le i \le 3m$
2	$\cdots\ x_{i,2}\ S_i$	$1 \le i \le 3m$
2	$\cdots\ x_{i,3}\ S_i$	$1 \le i \le 3m$
$2m-6$	$\cdots\ d_2\ S_i$	$1 \le i \le 3m$
$2m$	$\cdots\ x_i$	$1 \le i \le 3m$
m	$\cdots\ d_2\ x_i\ d_1$	$1 \le i \le 3m$
$m+2$	$\cdots\ d_2\ S_i\ d_1$	$1 \le i \le 3m$
$2m$	$\cdots\ d_2$	
1	$\cdots\ p\ d_1$	

For votes of the form $\cdots\ S_i\ p$, we use the price function $\rho(1) = 1$, and $\rho(t) = m+1$ for all $t \ge 2$; and for every other voter, we use the price function $\rho(t) = m+1$ for $t \ge 1$. Finally, set the budget $B = m$.

Note that for both voting rules, p is eliminated in the first round with $3m$ vetoes and therefore cannot be the winner without bribing voters.

We claim that $(X,\mathscr{S})$ is in X3C if and only if $((C,V),p,B,\rho)$ is in CONSTRUCTIVE-SHIFT-BRIBERY for either of iterated veto and veto with runoff, regardless of the winner model.

($\Rightarrow$) Suppose that $(X,\mathscr{S})$ is a yes-instance of X3C. Then there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of size m. Shift p one position forward in the votes of the form $\cdots\ S_i\ p$ for each $S_i \in \mathscr{S}'$, so that the new vote has the form $\cdots\ p\ S_i$. It follows that p, each $S \in \mathscr{S} \setminus \mathscr{S}'$, each x_i for $1 \leq i \leq 3m$, and d_2 are veto winners with $2m$ vetoes each and thus proceed to the second round. Since $\mathscr{S}'$ is an exact cover, each x_i receives two additional vetoes from the voters in lines 2–4 corresponding to the sets in the exact cover and m vetoes from the voters in line 7. Furthermore, each $S \in \mathscr{S} \setminus \mathscr{S}'$ receives $m+2$ vetoes from the voters in line 8, whereas p receives m vetoes from the voters in line 1 and only one additional veto from the voter in the last line. Since d_2 gains far more than $m+1$ vetoes in this round, it follows that p is the unique veto winner of the bribed election. Thus $((C,V),p,B,\rho)$ is a yes-instance of CONSTRUCTIVE-SHIFT-BRIBERY for either of iterated veto and veto with runoff, regardless of the winner model.

($\Leftarrow$) Suppose that $(X,\mathscr{S})$ is a no-instance of X3C. This means that for every $\mathscr{S}' \subseteq \mathscr{S}$, $|\mathscr{S}'| \leq m$, there is an $x' \in X$ that is not covered by any $S \in \mathscr{S}'$.

To not be eliminated in the first round and to not exceed the budget of m, p has to lose exactly m vetoes so as to tie with the $2m$ vetoes of the x_i. This is only possible by bribing the voters in the first line. Let $\mathscr{S}' \subseteq \mathscr{S}$, $|\mathscr{S}'| = m$, be the set that corresponds to the S_i of the bribed voters. Candidates p and d_2 as well as each $S \in \mathscr{S} \setminus \mathscr{S}'$ and each x_i, $1 \leq i \leq 3m$, reach the second round with $2m$ vetoes. However, in the second round, the $x' \in X$ that was not covered by $\mathscr{S}'$ receives only m additional vetoes in contrast to p who receives $m+1$ additional vetoes. It follows that p is not winning the election for either of the two voting rules. That means that $((C,V),p,B,\rho)$ is a no-instance of CONSTRUCTIVE-SHIFT-BRIBERY for either of iterated veto and veto with runoff, regardless of the winner model. □

We now turn to the destructive variant of shift bribery for iterated veto and veto with runoff.

Theorem 12. *In both the unique-winner and the nonunique-winner model, for veto with runoff and iterated veto,* DESTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard.*

Proof. To prove NP-hardness, we reduce the NP-complete problem ONE-IN-THREE-POSITIVE-3SAT to DESTRUCTIVE-SHIFT-BRIBERY for veto with runoff and iterated veto simultaneously. Given an instance (X,S) of ONE-IN-THREE-POSITIVE-3SAT, where $X = \{x_1,\dots,x_{3m}\}$ and $S = \{S_1,\dots,S_{3m}\}$, with $S_i = \{x_{i,1},x_{i,2},x_{i,3}\} \subseteq X$ for each $1 \leq i \leq 3m$, we construct the election (C,V) with candidate set $C = \{p,w,d_1,d_2\} \cup X$, designated candidate p, and the following list V of votes:

#	votes	for
1	$\cdots\ p\ x_i$	$1 \leq i \leq 3m$
2	$\cdots\ p\ x_{i,1}\ x_{i,2}\ d_1$	$1 \leq i \leq 3m$
2	$\cdots\ p\ x_{i,2}\ x_{i,3}\ d_1$	$1 \leq i \leq 3m$
2	$\cdots\ p\ x_{i,1}\ x_{i,3}\ d_1$	$1 \leq i \leq 3m$
7	$\cdots\ w\ x_{i,1}\ x_{i,2}\ x_{i,3}\ d_1$	$1 \leq i \leq 3m$
$2m$	$\cdots\ d_2\ x_i$	$1 \leq i \leq 3m$
$22m$	$\cdots\ d_2\ x_i\ d_1$	$1 \leq i \leq 3m$
$2m$	$\cdots\ d_2$	
m	$\cdots\ p$	
$2m$	$\cdots\ w$	
$8m-1$	$\cdots\ w\ d_1$	

For every vote of the form $\cdots\ p\ x_i$, let the price function be $\rho(1) = 1$ and $\rho(t) = m+1$ for every $t \geq 2$. For every other vote, define $\rho(t) = m+1$ for every $t \geq 1$. Finally, we set the budget $B = m$.

It is easy to see that p is the winner of this election for both voting rules.

We claim that (X,S) is in ONE-IN-THREE-POSITIVE-3SAT if and only if $((C,V),p,B,\rho)$ is in DESTRUCTIVE-SHIFT-BRIBERY for either of veto with runoff and iterated veto, regardless of the winner model.

($\Rightarrow$) Assume that (X,S) is in ONE-IN-THREE-POSITIVE-3SAT. Then there is a subset $X' \subseteq X$ such that for each clause S_i we have $|X' \cap S_i| = 1$. Bribe the voters with votes of the form $\cdots\ p\ x_i$ with $x_i \in X'$ so that the new vote has the form $\cdots\ x_i\ p$. It follows that p, w, d_2, and each $x_i \in X'$ have the fewest vetoes (namely, $2m$) and therefore proceed to the second round. In the second round, p receives $2m$ vetoes from the votes in line 1 and for each of the $3m$ clauses two vetoes from the voters in lines 2–4 for a total of $8m$ additional vetoes, whereas w only receives a total of $8m-1$ vetoes. It follows that p is not a winner of the election for either of the two voting rules.

($\Leftarrow$) Let (X,S) be a yes-instance of DESTRUCTIVE-SHIFT-BRIBERY for veto with runoff (respectively, iterated veto), i.e., it is possible to bribe voters so that p does not win the election. Recall that it is only possible to bribe voters in line 1 without ecxeeding the budget. In the first round, p receives m vetoes, i.e., the fewest vetoes of all candidates. Due to the votes in line 7, the only candidate capable of receiving fewer vetoes than p or the same number of vetoes as p in the second round is w.[9] However, this is only possible if p receives at least $9m-1$ additional vetoes since w has $10m-1$ vetoes in the second round from the last two lines alone. p receives $3m$ of these additional vetoes from line 1—after bribing voters so that p is in the last position, or eliminating the x_i in the first round—leaving a gap of $6m-1$ vetoes. For each clause S_j such that no $x_i \in S_j$ is present in the second round, p receives six additional vetoes (lines 2–4), whereas w receives in this case seven additional vetoes from the voters in line 5, i.e., this widens the gap between p and w instead of closing it. That means that for each clause S_j, there has to be at least one $x_i \in S_j$ present in the second round, i.e., for each clause S_j, a voter with a vote of the form $\cdots\ p\ x_i$ with $x_i \in S_j$ needs to be bribed to cast a vote of the form $\cdots\ x_i\ p$ to bring the respective vetoes down to $2m$, the same as, e.g., d_2. However, if at least two literals, say x_i and x_k, in a clause S_j are present in the second round, p receives no additional veto, which does not help to close the gap between p and w. The only possibility remaining for p not to be a winner of the bribed election is that the bribed voters correspond to the variables set to true in an assignment where in each clause there is exactly one literal true, i.e., we have a yes-instance of ONE-IN-THREE-POSITIVE-3SAT. □

7. Using the Nonmonotonicity Property

Informally stated, a voting rule is said to be *monotonic* if winners can never be turned into nonwinners by improving their position in some votes, everything else remaining the same.[10] Intuitively, that is to say that only shifting a candidate forward (closer to the top) is beneficial, whereas shifting a candidate backward (closer to the bottom) is not. In shift bribery under some monotonic voting rule, it thus makes only sense for the briber to shift the designated candidate forward in the constructive case (respectively, backward in the destructive case). However, all voting rules considered here except iterated plurality and iterated veto are *not* monotonic, and in nonmonotonic voting rules, shifting the designated candidate backward in the constructive case (respectively, forward in the destructive case) could also be beneficial for the briber.

It would therefore be interesting to find out whether the complexity of our problems changes when the nonmonotonicity of voting rules is specifically allowed, or even required, to be exploited in shift bribery

9. Note that d_1 will definitely be eliminated in the first round.

10. This definition captures just one common notion of monotonicity, the one we will be using here; but note that there are also other notions of monotonicity for voting rules known in social choice theory.

actions. Indeed, with slight modifications to the proofs, we can show that Hare-CONSTRUCTIVE-SHIFT-BRIBERY and plurality-with-runoff-CONSTRUCTIVE-SHIFT-BRIBERY are still NP-hard if the designated candidate can *only* be shifted *backward*. We conjecture that all other proofs (except the proofs for the monotonic voting rules iterated plurality and iterated veto) can be adapted in such a way as well.

We start with constructive shift bribery in Hare elections where the only allowed bribery action is to shift the designated candidate *backward*.

Theorem 13. *In both the unique-winner and the nonunique-winner model, Hare-*CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard even if the designated candidate can only be shifted backward.*

Proof. NP-hardness again follows by a reduction from X3C. Construct from a given X3C instance $(X,\mathscr{S})$ an instance $((C,V),p,B,\rho)$ of Hare-CONSTRUCTIVE-SHIFT-BRIBERY with candidate set $C = X \cup \mathscr{S} \cup D \cup \{p,w\}$, where $D = \{d_1,\dots,d_{3m}\}$ is a set of dummy candidates and p the designated candidate, and the following list V of votes:

#	vote	for
1	$S_i\ x_{i,1}\ \overrightarrow{X \setminus \{x_{i,1}\}}\ w\ p\ \cdots$	$1 \le i \le 3m$
1	$S_i\ x_{i,2}\ \overrightarrow{X \setminus \{x_{i,2}\}}\ w\ p\ \cdots$	$1 \le i \le 3m$
1	$S_i\ x_{i,3}\ \overrightarrow{X \setminus \{x_{i,3}\}}\ w\ p\ \cdots$	$1 \le i \le 3m$
4	$x_i\ \overrightarrow{X \setminus \{x_i\}}\ w\ p\ \cdots$	$1 \le i \le 3m$
6	$w\ \overrightarrow{X}\ p\ \cdots$	
1	$p\ S_i\ \cdots$	$1 \le i \le 3m$
6	$p\ \cdots$	
3	$d_i\ S_i\ p\ w \cdots$	$1 \le i \le 3m$

For votes of the form $p\ S_i\ \cdots$, we use the price function $\rho(1) = 1$, and $\rho(t) = m+1$ for all $t \ge 2$; and for every other vote, we use the price function ρ with $\rho(t) = m+1$ for all $t \ge 1$. Finally, set the budget $B = m$.

Without bribing the voters the election proceeds as follows:

Round	p	w	x_1	$x_i \in X \setminus \{x_1\}$	$S_i \in \mathscr{S}$	$d_i \in D$
1	$3m+6$	6	4	4	3	3
2	$12m+6$	6	7	7	out	out
3	$12m+6$	out	13	7	out	out
4	$12m+6$	out	$21m+6$	out	out	out

It follows that p is eliminated in the last round and does not win the election.

We claim that $(X,\mathscr{S})$ is in X3C if and only if $((C,V),p,B,\rho)$ is in Hare-CONSTRUCTIVE-SHIFT-BRIBERY, regardless of the winner model, even if the designated candidate can only be shifted backward.

$(\Rightarrow)$ Suppose that $(X,\mathscr{S})$ is a yes-instance of X3C. Then there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of size m. We now show that it is possible for p to become a unique Hare winner of an election obtained by shifting p in the votes without exceeding the budget B. For every $S_i \in \mathscr{S}'$, we bribe the voter with the vote of the form $p\ S_i\ \cdots$ by shifting p once, so her new vote is of the form $S_i\ p\ \cdots$; each such bribe action costs us only 1 from our budget, so the budget will not be exceeded. Now the election proceeds as follows:

Round	p	w	$x_i \in X$	$S_i \in \mathscr{S}'$	$S_i \in \mathscr{S} \setminus \mathscr{S}'$	$d_i \in D$
1	$2m+6$	6	4	4	3	3
2	$8m+6$	6	6	7	out	out
3	$26m+12$	out	out	7	out	out

We see that p is the only candidate still standing in the fourth round and thus the only Hare winner of the bribed election.

($\Leftarrow$) Suppose that $(X,\mathscr{S})$ is a no-instance of X3C. Then no subset $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq m$ covers X. We now show that p will be eliminated in all elections obtained by bribing voters without exceeding budget B. Note that we can only bribe at most m voters with votes of the form $p\ S_i\ \cdots$ without exceeding the budget. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that for every $S_i \in \mathscr{S}'$ we have bribed the voter whose vote is $p\ S_i\ \cdots$. We can assume that $|\mathscr{S}'| > 0$.

Every candidate in $\mathscr{S}'$ will gain an additional point and therefore survives the first round. All candidates from D and $\mathscr{S} \setminus \mathscr{S}'$ will be eliminated, since p only loses at most m points.

In the second round, the remaining candidates from $\mathscr{S}$ will gain three additional points from the elimination of candidates in D and score seven points in this round (and in all subsequent rounds with p still standing). If a candidate $S_i \in \mathscr{S}$ was eliminated in the previous round, every $x_j \in S_i$ gains one additional point in this round. Partition X into sets X_0, X_1, X_2, and X_3 so that $x_i \in X_k \Leftrightarrow |\{S_j \in \mathscr{S}' \mid x_i \in S_j\}| = k$ for $k \in \{0,1,2,3\}$. Note that X_0, X_1, X_2, and X_3 are disjoint and $|X_0| > 0$, but one or two of X_1, X_2, and X_3 may be empty. Then $x_i \in X_j$ scores $4+(3-j) \in \{4,5,6,7\}$ points depending on how many times x_i is covered by $\mathscr{S}'$. Therefore, every $x_i \in X_0$ scores more points than w who has six points. So, there are candidates from X that survive this round and other candidates from X (i.e., candidates from X_1, X_2, or X_3), who are eliminated.

In the third round, the candidate $x_\ell \in X$ with the smallest subscript who is still standing gains at least four points from the eliminated candidates, so that she scores at least nine points now (since no candidates from X_3 are left in the election). All other candidates still score the same number of points as in the previous round. Therefore, p scores $4|\mathscr{S} \setminus \mathscr{S}'|+6$ points, w scores six points (if w was not already eliminated along with the candidates from X_1), every $S_i \in \mathscr{S}'$ scores seven points, and every still standing candidate from X except x_ℓ scores at most seven points. Since w can only gain additional points when all candidates from X are eliminated and only x_ℓ gains points from the elimination of w or candidates from $X \setminus \{x_\ell\}$ in the subsequent rounds, all candidates $X \setminus (\{x_\ell\} \cup X_0)$ and w are eliminated. Then all still standing candidates from $X_0 \setminus \{x_\ell\}$ and candidates from $\mathscr{S}'$ who score seven points each are eliminated, which leaves p and x_ℓ in the last round. In this round, p scores $12m+6$ points and x_ℓ scores $21m+6$ points, so p is eliminated from the election and does not win. ☐

Next, we show the corresponding result for plurality with runoff.

Theorem 14. *In both the unique-winner and the nonunique-winner model, plurality-with-runoff-*CONSTRUCTIVE-SHIFT-BRIBERY *is* NP*-hard even if the designated candidate can only be shifted backward.*

Proof. To prove NP-hardness, we reduce X3C to CONSTRUCTIVE-SHIFT-BRIBERY for plurality with runoff. Let $(X,\mathscr{S})$ be a given X3C instance, where $X = \{x_1,\ldots,x_{3m}\}$ and $\mathscr{S} = \{S_1,\ldots,S_{3m}\}$. Also, we require that $m > 3$. We construct the CONSTRUCTIVE-SHIFT-BRIBERY instance $((C,V),p,B,\rho)$ as follows. Let $C = \{p\} \cup X \cup \mathscr{S} \cup D \cup Y$ with sets of dummy candidates $D = \{d_{i,j} \mid 1 \leq i \leq 3m$ and $1 \leq j \leq 2m^2-5m-4\}$ and $Y = \{y_i \mid 1 \leq i \leq 3m+1\}$ and designated candidate p. The list V of votes is constructed as follows:

#	vote	for
1	$p\ S_i\ \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,1}\ w\ \overrightarrow{X \setminus \{x_{i,1}\}}\ \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,2}\ w\ \overrightarrow{X \setminus \{x_{i,2}\}}\ \cdots$	$1 \leq i \leq 3m$
2	$S_i\ x_{i,3}\ w\ \overrightarrow{X \setminus \{x_{i,3}\}}\ \cdots$	$1 \leq i \leq 3m$
$3m$	$w\ p\ \cdots$	
1	$y_i\ p$	$1 \leq i \leq 3m+1$
$m-3$	$S_i\ w\ p$	$1 \leq i \leq 3m$
$m-4$	$S_i\ p\ w$	$1 \leq i \leq 3m$
$2m$	$x_i\ w\ p$	$1 \leq i \leq 3m$
1	$d_{i,j}\ x_i\ w\ p\ \cdots$	$1 \leq i \leq 3m$, $1 \leq j \leq 2m^2-5m-4$

For votes of the form $p\ S_i\ \cdots$, we use the price function $\rho(1) = 1$, and $\rho(t) = m+1$ for all $t \geq 2$; and for every other vote, we use the price function $\rho(t) = m+1$ for $t \geq 1$. Finally, set the budget $B = m$.

Without bribing, only p and w reach the second and final round with $3m$ points each. Clearly, w alone wins the election with only p and w present.

We claim that $(X, \mathscr{S})$ is in X3C if and only if $((C,V), p, B, \rho)$ is in CONSTRUCTIVE-SHIFT-BRIBERY for plurality with runoff, regardless of the winner model.

($\Rightarrow$) Suppose that $(X, \mathscr{S})$ is a yes-instance of X3C. Then there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ of size m. We now show that it is possible for p to become a unique plurality-with-runoff winner of an election obtained by shifting p in the votes without exceeding the budget. For every $S_i \in \mathscr{S}'$, we bribe the voter with the vote of the form $p\ S_i\ \cdots$ once, so her new vote is of the form $S_i\ p\ \cdots$.

In the first round, w scores $3m$ points; p, every $x_i \in X$, and every $S_i \in \mathscr{S}'$ score $2m$ points each; every $S_i \in \mathscr{S} \setminus \mathscr{S}'$ scores $2m-1$ points; and every candidate from D and Y scores only one point. Since w is the only plurality winner, all second-place candidates (namely, p, every $x_i \in X$, and every $S_i \in \mathscr{S}'$) proceed to the second round.

In the second round, every $S_i \in \mathscr{S}'$ still scores the same number of points as in the first round, w gains $2m(m-3)$ additional points, p gains $(3m+1)+2m(m-4)$ additional points, and every $x_i \in X$ gains $(2m^2-5m-4)+4$ additional points. Therefore, p alone wins the election with $2m^2-3m+1$ points, ahead of w and every $x_i \in X$ with $2m^2-3m$ points each, and every $S_i \in \mathscr{S}'$ with $2m$ points each.

($\Leftarrow$) Suppose that $((C,V), p, B, \rho)$ is a yes-instance of Plurality-with-runoff-CONSTRUCTIVE-SHIFT-BRIBERY. Notice that if no voters are bribed, p and w are leading in the election with $3m$ points each, so they both proceed to the final round. It is easy to see that w wins against p in a one-on-one election. To prevent w and p from being the only candidates in the second round, m voters with votes of the form $p\ S_i\ \cdots$ have to be bribed. Let $\mathscr{S}' \subseteq \mathscr{S}$ be such that $S_i \in \mathscr{S}'$ if the voter with vote $p\ S_i\ \cdots$ has been bribed. Then w, p, every $x_i \in X$, and every $S_i \in \mathscr{S}'$ survive the first round. Since every other candidate is deleted in the first round, p now scores $2m^2-5m+1$ points and beats w by a margin of one point. Moreover, p beats every $S_i \in \mathscr{S}'$ since the candidates from $\mathscr{S}'$ did not gain any additional points in this round. Regarding the candidates from X, every $x_i \in X$ gains $2m^2-5m-4$ points and two additional points for every $S_j \in \mathscr{S} \setminus \mathscr{S}'$ with $x_i \in S_j$ that was eliminated in the first round. Since there are exactly three $S_j \in \mathscr{S}$ with $x_i \in S_j$, every $x_i \in X$ can gain six points if all those candidates were eliminated in the last round, which would let x_i overtake p by one point. In order for p to beat all $x_i \in X$, at least one $S_j \in \mathscr{S}$ with $x_i \in S_j$ needs to be in $\mathscr{S}'$ and is therefore still standing in the second round. Since $|\mathscr{S}'| = m$ and there are $3m$ candidates in X, p can beat every $x_i \in X$ (and subsequently win the election) only if $\mathscr{S}'$ is an exact cover of X. $\square$

8. Conclusions and Open Questions

We have shown that shift bribery is NP-complete for each of the iterative voting systems of Hare, Coombs, Baldwin, Nanson, iterated plurality, plurality with runoff, iterated veto, and veto with runoff, each for both the constructive and the destructive case and in both the unique-winner and the nonunique-winner model. This contrasts previous results due to Elkind et al. (2009), Elkind and Faliszewski (2010), and Schlotter et al. (2017) showing that shift bribery can be solved efficiently by exact or approximation algorithms for many natural voting rules that do not proceed iteratively. Indeed, the iterative nature of the voting rules we have studied seems to be responsible for the hardness of shift bribery.

While these are interesting theoretical results complementing earlier work both on shift bribery and on these voting systems, NP-hardness of course has its limitations in terms of providing protection against shift bribery attacks in practice. Therefore, it would be interesting to also study shift bribery for these voting systems in terms of approximation and parameterized complexity and to do a typical-case analysis. Based on our results in this article, Zhou and Guo (2020) already obtained first results regarding the parameterized complexity of iterative voting systems with respect to a fixed number of shifts, votes, or candidates. Further, they have shown that the hardness of shift bribery for the Hare, Coombs, Baldwin, and Nanson rules also holds for unit price cost functions. It would be particularly interesting to determine the role of the cost function for the hardness of shift bribery. Furthermore, it would be interesting future work to study in detail the effect that specific tie-breaking models (such as the "parallel universes" model (Conitzer, Rognlie, & Xia, 2009) and other models) may have on the complexity of shift bribery problems for iterative voting rules.

A feature shared by most of the iterative voting rules we have studied is that many of them are not monotonic. This has the somewhat counterintuitive effect that shifting the designated candidate forward in some votes can actually hurt this candidate's chances to win, and shifting the designated candidate backward can increase these chances. We have discussed this feature in Section 7, showing that constructive shift bribery remains NP-hard even if we are allowed to only shift the designated candidate backward in some votes for two iterative voting systems: Hare voting and plurality with runoff. We leave the analogous question open for the remaining iterative voting systems studied here (except, of course, for the monotonic rules iterated plurality and iterated veto), and conjecture that they share this property. Even more interestingly, we pose as an open question whether there is a nonmonotonic voting system—a natural one or an artificially constructed one—for which unrestricted shift bribery is NP-hard but becomes efficiently solvable when restricted to shift bribery actions specifically exploiting their nonmonotonicity (i.e., allowing to shift the designated candidate only backward in the constructive case, or forward in the destructive case).

Acknowledgments

We thank the anonymous AAMAS'18 and ISAIM'18 reviewers for helpful comments. This work was supported in part by DFG grants RO 1202/21-1, RO 1202/15-1, RO 1202/14-2, and BA 6270/1-1.

References

Baldwin, J. (1926). The technique of the Nanson preferential majority system of election. *Transactions and Proceedings of the Royal Society of Victoria*, *39*, 42–52.

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

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

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

Baumeister, D., Faliszewski, P., Lang, J., & Rothe, J. (2012). Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pp. 577–584. IFAAMAS.

Baumeister, D., & Rothe, J. (2015). Preference aggregation by voting. In Rothe, J. (Ed.), *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer Texts in Business and Economics, chap. 4, pp. 197–325. Springer-Verlag.

Betzler, N., Niedermeier, R., & Woeginger, G. (2011). Unweighted coalitional manipulation under the Borda rule is NP-hard. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pp. 55–60. AAAI Press/IJCAI.

Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. (Eds.). (2016). *Handbook of Computational Social Choice*. Cambridge University Press.

Bredereck, R., Chen, J., Faliszewski, P., Guo, J., Niedermeier, R., & Woeginger, G. (2014a). Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, *19*(4), 358–373.

Bredereck, R., Chen, J., Faliszewski, P., Nichterlein, A., & Niedermeier, R. (2014b). Prices matter for the parameterized complexity of shift bribery. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 1398–1404. AAAI Press.

Bredereck, R., Faliszewski, P., Niedermeier, R., & Talmon, N. (2016a). Complexity of shift bribery in committee elections. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pp. 2452–2458. AAAI Press.

Bredereck, R., Faliszewski, P., Niedermeier, R., & Talmon, N. (2016b). Large-scale election campaigns: Combinatorial shift bribery. *Journal of Artificial Intelligence Research*, *55*, 603–652.

Conitzer, V., Rognlie, M., & Xia, L. (2009). Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 109–115. IJCAI.

Conitzer, V., Sandholm, T., & Lang, J. (2007). When are elections with few candidates hard to manipulate? *Journal of the ACM*, *54*(3), Article 14.

Conitzer, V., & Walsh, T. (2016). Barriers to manipulation in voting. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. (Eds.), *Handbook of Computational Social Choice*, chap. 6, pp. 127–145. Cambridge University Press.

Davies, J., Katsirelos, G., Narodytska, N., Walsh, T., & Xia, L. (2014). Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artificial Intelligence*, *217*, 20–42.

Davies, J., Narodytska, N., & Walsh, T. (2012). Eliminating the weakest link: Making manipulation intractable? In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 1333–1339. AAAI Press.

Elkind, E., & Faliszewski, P. (2010). Approximation algorithms for campaign management. In *Proceedings of the 6th International Workshop on Internet & Network Economics*, pp. 473–482. Springer-Verlag *Lecture Notes in Computer Science #6484*.

Elkind, E., Faliszewski, P., & Slinko, A. (2009). Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pp. 299–310. Springer-Verlag *Lecture Notes in Computer Science #5814*.

Erdélyi, G., Neveling, M., Reger, C., Rothe, J., Yang, Y., & Zorn, R. (2021). Towards completing the puzzle: Complexity of control by replacing, adding, and deleting candidates or voters. *Journal of Autonomous Agents and Multi-Agent Systems*, *forthcoming*. Conference versions due to Erdélyi, Reger, and Yang (2019) and Neveling, Rothe, and Zorn (2020).

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. IFAAMAS.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2009a). How hard is bribery in elections? *Journal of Artificial Intelligence Research*, *35*, 485–532.

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

Faliszewski, P., Manurangsi, P., & Sornat, K. (2019). Approximation and hardness of shift-bribery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 1901–1908. AAAI Press.

Faliszewski, P., Reisch, Y., Rothe, J., & Schend, L. (2015). Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting. *Journal of Autonomous Agents and Multi-Agent Systems*, *29*(6), 1091–1124.

Faliszewski, P., & Rothe, J. (2016). Control and bribery in voting. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. (Eds.), *Handbook of Computational Social Choice*, chap. 7, pp. 146–168. Cambridge University Press.

Freeman, R., Brill, M., & Conitzer, V. (2014). On the axiomatic characterization of runoff voting rules. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pp. 675–681. AAAI Press.

Garey, M., & Johnson, D. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

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

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

Kaczmarczyk, A., & Faliszewski, P. (2016). Algorithms for destructive shift bribery. In *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*, pp. 305–313. IFAAMAS.

Knop, D., Koutecký, M., & Mnich, M. (2017). Voting and bribing in single-exponential time. In *Proceedings of the 34th Annual Symposium on Theoretical Aspects of Computer Science*, Vol. 66 of *LIPIcs*, pp. 46:1–46:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

Konczak, K., & Lang, J. (2005). Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pp. 124–129.

Levin, J., & Nalebuff, B. (1995). An introduction to vote-counting schemes. *The Journal of Economic Perspectives*, *9*(1), 3–26.

Maushagen, C., Neveling, M., Rothe, J., & Selker, A. (2018). Complexity of shift bribery in iterative elections. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1567–1575. IFAAMAS.

Nanson, E. (1882). Methods of election. *Transactions and Proceedings of the Royal Society of Victoria*, *19*, 197–240.

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. Springer-Verlag *Lecture Notes in Computer Science #12159*.

Papadimitriou, C. (1995). *Computational Complexity* (Second edition). Addison-Wesley.

Porschen, S., Schmidt, T., Speckenmeyer, E., & Wotzlaw, A. (2014). XSAT and NAE-SAT of linear CNF classes. *Discrete Applied Mathematics*, *167*, 1–14.

Reisch, Y., Rothe, J., & Schend, L. (2014). The margin of victory in Schulze, cup, and Copeland elections: Complexity of the regular and exact variants. In *Proceedings of the 7th European Starting AI Researcher Symposium*, pp. 250–259. IOS Press.

Rothe, J. (2005). *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag.

Rothe, J. (Ed.). (2015). *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer Texts in Business and Economics. Springer-Verlag.

Rothe, J. (2019). Borda count in collective decision making: A summary of recent results. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pp. 9830–9836. AAAI Press.

Schlotter, I., Faliszewski, P., & Elkind, E. (2017). Campaign management under approval-driven voting rules. *Algorithmica*, *77*, 84–115.

Taylor, A. (2005). *Social Choice and the Mathematics of Manipulation*. Cambridge University Press.

Xia, L. (2012). Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pp. 982–999. ACM Press.

Xia, L., & Conitzer, V. (2011). Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research*, *41*, 25–67.

Zhou, A., & Guo, J. (2020). Parameterized complexity of shift bribery in iterative elections. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1665–1673. IFAAMAS.

Zwicker, W. (2016). Introduction to the theory of voting. In Brandt, F., Conitzer, V., Endriss, U., Lang, J., & Procaccia, A. (Eds.), *Handbook of Computational Social Choice*, chap. 2, pp. 23–56. Cambridge University Press.

CHAPTER 7

CONCLUSIONS

We have studied how efficiently elections can be tampered with depending on which voting rule we choose to evaluate the election.

In Section 3 we have studied electoral control for the Borda Count and solved all open cases of standard electoral control and some cases of online electoral control. Borda turns out to be very resistant to constructive electoral control being resistant to all (standard) constructive types and in contrast vulnerable to most types of destructive electoral control. For future work we propose to solve the open problems of online candidate control, in particular involving partitioning the set of candidates. Furthermore, for the NP-hard cases parameterized complexity can be studied with, e.g., the number of candidates or voters as the parameter and for the vulnerable cases, the complexity of the more general cases with weighted elections can be studied. Lastly, the study of structured domains[1] (i.e., single-peaked and single-crossing elections) has been given attention lately [17, 22, 67, 159] and we propose to study whether the complexity of control for Borda changes if elections are structured.

In Section 4 we have studied electoral control focusing on control by replacing candidates or voters for various voting rules thus taking a step to complete the picture of complexity results regarding electoral control. One important case is still open which is constructive control by replacing candidates for 2-approval. The problem is seemingly related to the corresponding problem with 3-veto which is shown to be in P but the same approach cannot be used here. Since the problems for constructive control by *adding* candidates and by *deleting* candidates are in P for 2-approval, constructive control by replacing candidates is likely to be in P for 2-approval as well. On the contrary, showing that the problem is NP-hard would be interesting as we found that the complexity of replacement control usually follows the complexity of the corresponding problems of control by addition and deletion. Next, the problems for control by partitioning of candidates or voters are still open for plurality/veto with run-off. Lastly, showing dichotomy results for pure scoring rules similar to Hemaspaandra and Schnoor [90] is a challenging and interesting task.

In Section 5 we have devised a model for studying electoral control by cloning candidates in the setting of multiwinner elections and found a wide range of complexity results from easy cases like SNTV over cases that are easy in some ways but hard in others like k-Borda to cases like STV for which cloning is generally hard. We propose to solve the open cases regarding k-approval-CC and Borda-CC and extend our study to other multiwinner voting rules. In particular, Bredereck et al. [29] considered approximative versions of k-approval-CC and Borda-CC that only compute an approximated solution but run in polynomial time. Furthermore, it is interesting to study other classes of prize functions such

[1] Structured domains are motivated by the fact that, in practice, the voters' preferences are rarely purely random but structured in some way. For example, in political elections all candidates can be ordered on a left-right scale and voters tend to vote according to this scale. That is, a voter belonging to the left spectrum obviously prefers candidates on the left to candidates on the right.

as *all-or-nothing* prices. Lastly, our model could be extended to take on a probabilistic perspective similar to the model of cloning in singlewinner elections by Elkind, Faliszewski, and Slinko [53].

In Section 6 we have studied shift bribery for iterative scoring rule. We found that iterative scoring rules seem to be very resistant to shift bribery by showing NP-hardness of shift bribery for all iterative scoring rules that we have studied. In contrast, the standard non-iterative scoring rules are sometimes vulnerable to shift bribery as is the case for k-approval [52]. We have also investigated how nonmonotonicity affects the complexity of shift bribery by allowing the distinguished candidate to be shifted backwards in the constructive case and forwards in the destructive case and found no change in complexity. We propose to continue this study by proving or disproving our conjecture that using the nonmonotonicity of iterative scoring rules does not change the complexity of shift bribery for them. Since we have found exclusively NP-hardness results studying parameterized complexity for our problems with common parameters—the number of candidates, the number of voters, or the budget—seems natural. Recently, Zhou and Guo [160] started research in this direction by studying the parameterized complexity of shift bribery for four of our iterative voting rules finding a wide range of results including FPT and W[1]-hard cases. We propose to solve the cases that they left open and extend their study to other iterative voting rules. Moreover, we have studied iterative versions of the scoring rules plurality, veto, and Borda but there are many more scoring rules for which the iterative versions could be studied. It would be interesting to know if shift bribery is NP-hard for all of them which seems likely.

The study of the computational complexity of election tampering attempts (in particular, electoral control and bribery) has been a thriving research direction in computational social choice. In this thesis we have only covered worst-case complexity which admittedly is not the last word of wisdom. Rothe and Schend [139] argue that often times although some voting rule is resistant (i.e., NP-hard) against some form of election tampering on average the corresponding problem can be solved efficiently. Therefore, finding hardness in the average-case in addition to hardness in the worst-case is an interesting and important challenge for future work. Recently, Spielman, and Teng [148] proposed smoothed complexity theory which investigates the running time of algorithms when the input is randomly perturbed. In essence, smoothed complexity tries to answer the question of how robust or fragile worst-case instances of hard problems are. Therefore, smoothed complexity theory stands between worst-case and average-case analysis. Baumeister, Hogrebe, and Rothe [11] proposed to apply smoothed complexity to computational social choice. Furthermore, finding connections or interactions between the different subfields of computational social choice often yields interesting results as was done by Rey and Rothe [133] who, inspired by electoral control, have studied structural control in weighted voting games or Rothe, Schadrack, and Schend [138] who have used the Borda Count for FEN-hedonic games. Lastly, over the many years of research in computational social choice we have gained substantial insights into voting rules but applications of our insights besides for elections are sparse, so as a long term goal we propose to find new applications beyond computational social choice where our knowledge of voting rules becomes valuable.

BIBLIOGRAPHY

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

[2] K. Arrow. *Social Choice and Individual Values.* Wiley: New York, first edition, 1951.

[3] H. Aziz, M. Brill, v. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. *Social Choice and Welfare*, 48(2):461–485, 2017.

[4] H. Aziz, E. Elkind, P. Faliszewski, M. Lackner, and P. Skowron. The Condorcet principle for multiwinner elections: From shortlisting to proportionality. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 84–90. IJCAI, 2017.

[5] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 107–115. IFAAMAS, 2015.

[6] H. Aziz, B. Lee, and N. Talmon. Proportionally representative participatory budgeting: Axioms and algorithms. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 23–31. IFAAMAS, July 2018.

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

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

[9] J. Bartholdi III, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

[10] 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.

[11] D. Baumeister, T. Hogrebe, and J. Rothe. Towards reality: Smoothed analysis in computational social choice. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1691–1695. IFAAMAS, May 2020.

[12] D. Baumeister and J. Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, 2012.

[13] G. Benadè, S. Nath, A. Procaccia, and N. Shah. Preference elicitation for participatory budgeting. *Management Science*, 67(5):2813–2827, 2020.

[14] N. Betzler and B. Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.

[15] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 53–58. IJCAI, July 2009.

[16] N. Betzler, R. Niedermeier, and G. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 55–60. AAAI Press/IJCAI, July 2011.

[17] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

[18] D. Black. On the rationale of group decision-making. *Journal of Political Economy*, 56(1):23–34, 1948.

[19] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[20] J.-C. de Borda. Mémoire sur les élections au scrutin. *Histoire de L'Académie Royale des Sciences, Paris*, 1781. English translation appears in the paper by de Grazia [42].

[21] S. Brams, P. Edelman, and P. Fishburn. Fair division of indivisible items. *Theory and Decision*, 55(2):147–180, 2003.

[22] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. *Journal of Artificial Intelligence Research*, 53:439–496, July 2015.

[23] F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[24] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. *Information and Computation*, 251:140–164, 2016.

[25] R. Bredereck, J. Chen, S. Hartung, S. Kratsch, R. Niedermeier, O. Suchý, and G. Woeginger. A multivariate complexity analysis of lobbying in multiple referenda. *Journal of Artificial Intelligence Research*, 50:409–446, 2014.

[26] R. Bredereck, J. Chen, R. Niedermeier, and T. Walsh. Parliamentary voting procedures: Agenda control, manipulation, and uncertainty. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, pages 164–170. AAAI Press/IJCAI, July 2015.

[27] R. Bredereck, P. Faliszewski, A. Kaczmarczyk, R. Niedermeier, P. Skowron, and N. Talmon. Robustness among multiwinner voting rules. *Artificial Intelligence*, 290:Article 103403, 2021.

[28] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Large-scale election campaigns: Combinatorial shift bribery. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 67–75. IFAAMAS, May 2015.

[29] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Complexity of shift bribery in committee elections. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2452–2458. AAAI Press, February 2016.

[30] Y. Cabannes. Participatory budgeting: A significant contribution to participatory democracy. *Environment and Urbanization*, 16(1):27–46, 2004.

[31] Y. Chevaleyre, U. Endriss, J. Lang, and N. Maudet. A short introduction to computational social choice. In *Proceedings of the 33rd International Conference on Current Trends in Theory and Practice of Computer Science*, pages 51–69. Springer-Verlag *Lecture Notes in Computer Science #4362*, January 2007.

[32] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58:345–363, 1936.

[33] J.-A.-N. de Caritat, Marquis de Condorcet. Plan de constitution, presenté à la convention nationale les 15 et 16 Fevrier 1793. In A. Condorcet O'Connor and M. Arago, editors, *Oeuvres de Condorcet, V. 12*. Firmin Didot Frères, 1847.

[34] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 109–115. IJCAI, July 2009.

[35] V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proceedings of the 18th National Conference on Artificial Intelligence*, pages 392–397. AAAI Press, 2002.

[36] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[37] V. Conitzer and T. Walsh. Barriers to manipulation in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 6, pages 127–145. Cambridge University Press, 2016.

[38] 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.

[39] M. Cygan, F. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[40] Baumeister D, S. Dennisen, and L. Rey. Winner determination and manipulation in minisum and minimax committee elections. In *Proceedings of the 4th International Conference on Algorithmic Decision Theory*, pages 469–485. Springer-Verlag *Lecture Notes in Artificial Intelligence #9346*, September 2015.

[41] J. Davies, G. Katsirelos, N. Narodytska, T. Walsh, and L. Xia. Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artificial Intelligence*, 217:20–42, 2014.

[42] A. de Grazia. Mathematical deviation of an election system. *Isis*, 44(1–2):41–51, 1953.

[43] B. Debord. Prudent k-choice functions: Properties and algorithms. *Mathematical Social Sciences*, 26(1):63–77, 1993.

[44] C. Dodgson. Suggestions as to the best method of taking votes, where more than two issues are to be voted on. 1874. Reprints appear in [111, pp. 287–288] and in [19, pp. 222–224].

[45] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.

[46] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[47] R. Downey and M. Fellows. *Fundamentals of Parameterized Complexity*, volume 4. Springer, 2013.

[48] J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard–Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[49] B. Dutta, M. Jackson, and M. Le Breton. Strategic candidacy and voting procedures. *Econometrica*, 69(4):1013–1037, 2001.

[50] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International World Wide Web Conference*, pages 613–622. ACM Press, 2001.

[51] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. *Social Choice and Welfare*, 48(3):599–632, 2017.

[52] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 299–310. Springer-Verlag *Lecture Notes in Computer Science #5814*, October 2009.

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

[54] E. Elkind and J. Rothe. Cooperative game theory. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer Texts in Business and Economics, chapter 3, pages 135–193. Springer-Verlag, 2015.

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

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

[57] G. Erdélyi, E. Hemaspaandra, and L. Hemaspaandra. Bribery and voter control under voting-rule uncertainty. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pages 61–68. IFAAMAS, May 2014.

[58] 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):1–48, 2021.

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

[60] 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.

[61] R. Erikson. Malapportionment, gerrymandering, and party fortunes in congressional elections. *The American Political Science Review*, pages 1234–1245, 1972.

[62] P. Faliszewski. Nonuniform bribery. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 1569–1572. IFAAMAS, May 2008.

[63] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[64] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

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

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

[67] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011.

[68] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems*, pages 983–990. IFAAMAS, May 2008.

[69] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[70] 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.

[71] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Multiwinner voting: A new challenge for social choice theory. In U. Endriss, editor, *Trends in Computational Social Choice*, chapter 2, pages 27–47. AI Access Foundation, 2017.

[72] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Multiwinner analogues of the plurality rule: Axiomatic and algorithmic perspectives. *Social Choice and Welfare*, 51(3):513–550, 2018.

[73] P. Faliszewski, P. Skowron, and N. Talmon. Bribery as a measure of candidate success: Complexity results for approval-based multiwinner rules. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 6–14. IFAAMAS, May 2017.

[74] D. Felsenthal and Z. Maoz. Normative properties of four single-stage multi-winner electoral procedures. *Behavioral Science*, 37(2):109–127, 1992.

[75] P. Fishburn. *The Theory of Social Choice*. Princeton University Press, Princeton, N.J., 1973.

[76] J. Flum and M. Grohe. *Parameterized Complexity Theory*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2006.

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

[78] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

[79] M. Guo and V. Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 1676–1679. AAAI Press, July 2010.

[80] J. Hartmanis and R. Stearns. On the computational complexity of algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

[81] T. Haynes, S. Sen, N. Arora, and R. Nadella. An automated meeting scheduling system that utilizes user preferences. In *Proceedings of the 1st International Conference on Autonomous Agents*, pages 308–315. ACM Press, 1997.

[82] E. Hemaspaandra and L. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

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

[84] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[85] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online manipulation of sequential elections. *Journal of Computer and System Sciences*, 80(4):697–710, 2014.

[86] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of controlling candidate-sequential elections. *Theoretical Computer Science*, 678:14–21, 2017.

[87] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online voter control in sequential elections. *Journal of Autonomous Agents and Multi-Agent Systems*, 31(5):1055–1076, 2017.

[88] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. The complexity of online bribery in sequential elections. In *Proceedings of the 17th Conference on Theoretical Aspects of Rationality and Knowledge*, pages 233–251. Electronic Proceedings in Theoretical Computer Science #297, July 2019.

[89] E. Hemaspaandra, L. Hemaspaandra, and H. Schnoor. A control dichotomy for pure scoring rules. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 712–720. AAAI Press, July 2014.

[90] E. Hemaspaandra and H. Schnoor. Dichotomy for pure scoring rules under manipulative electoral actions. In *Proceedings of the 22nd European Conference on Artificial Intelligence*, pages 1071–1079. IOS Press, August/September 2016.

[91] S. Issacharoff. Gerrymandering and political cartels. *Harvard Law Review*, pages 593–648, 2002.

[92] A. Kaczmarczyk and P. Faliszewski. Algorithms for destructive shift bribery. *Journal of Autonomous Agents and Multi-Agent Systems*, 33(3):275–297, 2019.

[93] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[94] J. Kemeny. Mathematics without numbers. *Dædalus*, 88(4):571–591, 1959.

[95] D. Kilgour. Approval balloting for multi-winner elections. In J. Laslier and R. Sanver, editors, *Handbook on Approval Voting*, chapter 6, pages 105–124. Springer, 2010.

[96] M. Kilgour, S. Brams, and R. Sanver. How to elect a representative committee using approval balloting. In F. Pukelsheim and B. Simeone, editors, *Mathematics and Democracy: Voting Systems and Collective Choice*, pages 83–95. Springer, 2006.

[97] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[98] K. Lam and C. Leung. Rank aggregation for meta-search engines. In *Proceedings of the 13th International World Wide Web Conference on Alternate Track Papers & Posters*, pages 384–385, 2004.

[99] J. Lang. Collective decision making under incomplete knowledge: Possible and necessary solutions. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 4885–4891. IJCAI, July 2020.

[100] J. Lang and J. Rothe. Fair division of indivisible goods. In J. Rothe, editor, *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, Springer Texts in Business and Economics, chapter 8, pages 493–550. Springer-Verlag, 2015.

[101] H. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[102] 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.

[103] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 280–286. AAAI Press/IJCAI, July 2011.

[104] T. Magrino, R. Rivest, E. Shen, and D. Wagner. Computing the margin of victory in IRV elections. In *Website Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy Elections*, August 2011.

[105] J. Manza and C. Uggen. *Locked out: Felon disenfranchisement and American democracy*. Oxford University Press, 2008.

[106] C. Maushagen, M. Neveling, J. Rothe, and A.-K. Selker. Complexity of shift bribery for iterative voting rules. *Journal of Artificial Intelligence Research*. Submitted.

[107] C. Maushagen, M. Neveling, J. Rothe, and A.-K. Selker. Complexity of shift bribery in iterative elections. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 1567–1575. IFAAMAS, July 2018.

[108] C. Maushagen and J. Rothe. 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.

[109] C. Maushagen and J. Rothe. 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*, pages 163–170. IOS Press, August/September 2020.

[110] K. May. A set of independent necessary and sufficient conditions for simple majority decision. *Econometrica: Journal of the Econometric Society*, pages 680–684, 1952.

[111] I. McLean and A. Urken. *Classics of Social Choice*. University of Michigan Press, 1995.

[112] R. Meir, M. Polukarov, J. Rosenschein, and N. Jennings. Convergence to equilibria in plurality voting. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 823–828. AAAI Press, 2010.

[113] 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.

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

[115] A. Meyer and L. Stockmeyer. The equivalence problem for regular expressions with squaring requires exponential space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129. IEEE Computer Society Press, 1972.

[116] Nina Narodytska and Toby Walsh. Manipulating two stage voting rules. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pages 423–430. IFAAMAS, May 2013.

[117] M. Neveling and J. Rothe. Closing the gap of control complexity in Borda elections: Solving ten open cases. In *Proceedings of the 18th Italian Conference on Theoretical Computer Science*, volume 1949, pages 138–149. CEUR-WS.org, September 2017.

[118] M. Neveling and J. Rothe. Solving seven open problems of offline and online control in Borda elections. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, pages 3029–3035. AAAI Press, February 2017.

[119] 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.

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

[121] 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, June 2020.

[122] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[123] S. Obraztsova, Y. Zick, and E. Elkind. On manipulation in multi-winner elections based on scoring rules. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pages 359–366. IFAAMAS, 2013.

[124] K. Oflazer and G. Tür. Morphological disambiguation by voting constraints. In *Proceedings of the 8th Conference of the European Chapter of the Association for Computational Linguistics*, pages 222–229. ACL/Morgan Kaufmann, 1997.

[125] Y. Narahari P. Dey, N. Misra. Frugal bribery in voting. *Theoretical Computer Science*, 676:15–32, 2017.

[126] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.

[127] D. Parkes and L. Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1429–1435. AAAI Press, July 2012.

[128] D. Pennock, E. Horvitz, and C. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th National Conference on Artificial Intelligence*, pages 729–734. AAAI Press, July/August 2000.

[129] D. Peters. Proportionality and strategyproofness in multiwinner elections. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, page 1549–1557. IFAAMAS, July 2018.

[130] M. Pini, F. Rossi, B. Venable, and T. Walsh. Dealing with incomplete agents' preferences and an uncertain agenda in group decision making via sequential majority voting. In *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning*, pages 571–578. AAAI Press, September 2008.

[131] A. Procaccia, J. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control, and winner-determination. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1476–1481. IJCAI, January 2007.

[132] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[133] A. Rey and J. Rothe. Structural control in weighted voting games. *The B.E. Journal on Theoretical Economics*, 18(2), 2018.

[134] A. Roth, T. Sönmez, and M. Ünver. Kidney exchange. *The Quarterly Journal of Economics*, 119(2):457–488, 2004.

[135] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[136] J. Rothe, editor. *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer Texts in Business and Economics. Springer-Verlag, 2015.

[137] J. Rothe. Borda count in collective decision making: A summary of recent results. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 9830–9836. AAAI Press, January/February 2019.

[138] J. Rothe, H. Schadrack, and L. Schend. Borda-induced hedonic games with friends, enemies, and neutral players. *Mathematical Social Sciences*, 96:21–36, 2018.

[139] J. Rothe and L. Schend. Challenges to complexity shields that are supposed to protect elections against manipulation and control: A survey. *Annals of Mathematics and Artificial Intelligence*, 68(1–3):161–193, 2013.

[140] D. Russo. *Structural Properties of Complexity Classes*. PhD thesis, University of California at Santa Barbara, Santa Barbara, CA, 1985.

[141] D. Saari. Which is better: The Condorcet or Borda winner? *Social Choice and Welfare*, 27(1):107–129, 2006.

[142] 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.

[143] I. Schlotter, P. Faliszewski, and E. Elkind. Campaign management under approval-driven voting rules. *Algorithmica*, 77:84–115, 2017.

[144] S. Sekar, S. Sikdar, and L. Xia. Condorcet consistent bundling with social choice. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 33–41. IFAAMAS, May 2017.

[145] G. Sigletos, G. Paliouras, C. Spyropoulos, and M. Hatzopoulos. Combining information extraction systems using voting and stacked generalization. *Journal of Machine Learning Research*, 6(11):1751–1782, 2005.

[146] P. Skowron, P. Faliszewski, and J. Lang. Finding a collective set of items: From proportional multirepresentation to group recommendation. *Artificial Intelligence*, 241:191–216, 2016.

[147] P. Skowron, P. Faliszewski, and A. Slinko. Axiomatic characterization of committee scoring rules. *Journal of Economic Theory*, 180:244–273, 2019.

[148] D. Spielman and Teng S. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3):385–463, 2004.

[149] L. Stockmeyer. The polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):1–22, 1976.

[150] N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[151] C. Tovey. Tutorial on computational complexity. *Interfaces*, 32(3):30–61, 2002.

[152] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society, ser. 2*, 42:230–265, 1936. Correction, *ibid*, vol. 43, pp. 544–546, 1937.

[153] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, pages 3–8. AAAI Press, July 2007.

[154] M. Weston. One person, no vote: Staggered elections, redistributing and disenfranchisement. *Yale LJ*, 121:2013, 2011.

[155] L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 982–999. ACM Press, June 2012.

[156] L. Xia and V. Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

[157] L. Xia, V. Conitzer, and A. Procaccia. A scheduling approach to coalitional manipulation. In *Proceedings of the 11th ACM Conference on Electronic Commerce*, pages 275–284. ACM Press, June 2010.

[158] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 348–353. IJCAI, July 2009.

[159] Y. Yang. On the complexity of Borda control in single-peaked elections. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 1178–1186. IFAAMAS, May 2017.

[160] A. Zhou and J. Guo. Parameterized complexity of shift bribery in iterative elections. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1665–1673. IFAAMAS, May 2020.

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. Desweiteren 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 folgender Zeitschriftenartikel und Konferenzberichte veröffentlicht oder zur Begutachtung eingereicht und sind entsprechend gekennzeichnet: [58], [60], [107], [106], [120], [117], [118], [121], [119], [137].

Ort, Datum

Marc Neveling