



On the Parameterized Complexity of Controlling Amendment and Successive Winners

Yongjie Yang¹

Received: 5 December 2023 / Accepted: 28 February 2025 / Published online: 13 March 2025
© The Author(s) 2025, corrected publication 2025

Abstract

The amendment procedure and the successive procedure have been widely employed in parliamentary and legislative decision making and have undergone extensive study in the literature from various perspectives. However, investigating them through the lens of computational complexity theory has not been as thoroughly conducted as for many other prevalent voting procedures heretofore. To the best of our knowledge, there is only one paper which explores the complexity of several strategic voting problems under these two procedures, prior to our current work. To provide a better understanding of to what extent the two procedures resist strategic behavior, we study the parameterized complexity of constructive/destructive control by adding/deleting voters/candidates for both procedures. To enhance the generalizability of our results, we also examine a more generalized form of the amendment procedure. Our exploration yields a comprehensive (parameterized) complexity landscape of these problems with respect to numerous parameters.

Keywords Amendment procedure · Successive procedure · Election control · NP-hard · $W[1]$ -hard · $W[2]$ -hard · FPT

1 Introduction

The amendment and the successive voting procedures are two fundamental methods commonly employed in parliamentary and legislative decision-making. The amendment procedure is particularly prevalent in countries such as the United States, the United Kingdom, Finland, and Switzerland, among others. Meanwhile, the successive procedure is widely utilized across many European nations, including the Czech Republic, Denmark, Germany, Hungary, and Iceland. For a more detailed discussion,

An extended abstract summarizing the results for the amendment procedure and the successive procedure appeared in the Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022) [54].

✉ Yongjie Yang
yyongjiecs@gmail.com

¹ Chair of Economic Theory, Saarland University, 66123 Saarbrücken, Germany

refer to Rasch [46]. Since the pioneering works of Black [8] and Farquharson [28], these two procedures have been extensively and intensively investigated in the literature (see, e.g., the works of Apesteguia, Ballester, and Masatlioglu [1], Horan [34], Miller [43], Rasch [46, 47]). Each of these procedures takes as input a set of candidates, a group of voters with linear preferences over these candidates, and an agenda (also referred to as a priority), which is specified as a linear order of the candidates. The output is a single candidate selected as the winner. The agenda specifies the order in which candidates are considered in the decision making process. The amendment procedure takes m rounds to determine the winner, each determining a temporary winner, where m denotes the number of candidates. The winner of the first round is the first candidate in the agenda. For the i th round, the winner is either the i th candidate in the agenda or the winner of the $(i - 1)$ th round, determined by the head-to-head comparison between the two candidates, with the winning candidate becoming the winner of the i th round. The amendment winner is the winner of the last round. For the successive procedure, the winner is the first candidate in the agenda for whom there is a majority of voters, with each voter preferring this candidate to all successors of the candidate.

In practice, several factors can influence the outcome of an election. For instance, it is known that given the same voting profile, different agendas may result in different winners. Notably, according to the works of Black [8] and Miller [43], the successive procedure is more vulnerable to agenda control than the amendment procedure, in the sense that for the same profile, any candidate that can be made the amendment winner by some agenda can also be made the successive winner by some agenda (see also the work of Barberá and Gerber [3] for an extension of this result). To gain a more nuanced understanding of how election outcomes under the amendment and the successive procedures can be affected by different factors, Bredereck et al. [11] studied the complexity of several related combinatorial problems under these two procedures. Concretely, they studied problems such as AGENDA CONTROL, COALITION MANIPULATION, POSSIBLE WINNER, NECESSARY WINNER, and weighted variants of these problems. Their results reveal that the amendment procedure is more resistant to agenda control and manipulation than the successive procedure.

To the best of our knowledge, the work of Bredereck et al. [11] is so far the sole exploration into the complexity of strategic problems under these two procedures, leaving many other types of strategic operations for these two procedures unexplored. In an effort to fill these gaps and significantly expand our understanding of the extent to which the two procedures resist other types of strategic operations, we investigate several standard control problems. Particularly, we delve into the problems of constructive control by adding/deleting voters/candidates, initially proposed in the pioneering paper by Bartholdi, Tovey, and Trick [4]. These problems model scenarios where we are given a number of registered voters and candidates, and an election controller aims to make a distinguished candidate the winner by either adding a limited number of unregistered voters/candidates or deleting a limited number of registered voters/candidates. Additionally, we study the destructive counterparts of these problems, first proposed by Hemaspaandra, Hemaspaandra, and Rothe [33]. In the destructive control problems, the goal of the controller is to prevent the distinguished candidate from becoming the winner.

To make our results more general, in addition to the two procedures, we also study a generalization of the amendment procedure, which we term h -amendment, where h is a positive integer. Generally speaking, this procedure involves multiple rounds, where in each round, a candidate c is compared with her next h successors to determine whether c or the next h successors of c in the agenda are eliminated. The winner is the one who survives the last round. (We refer to Sect. 2 for the formal definition.) Notably, the standard amendment procedure is exactly the 1-amendment procedure. Furthermore, another intriguing special case of the h -amendment procedures arises when h equals the number of candidates minus one. We refer to this special case as the full-amendment procedure. An advantage of the full-amendment procedure is that it selects a superior winner compared to the amendment procedure and the successive procedure. Specifically, for a fixed election and a given agenda, the full-amendment winner either coincides with or is preferred by a majority of voters over the amendment/successive winner.

For these procedures, we provide a comprehensive understanding of the parameterized complexity of election control problems, including many intractability results (NP-hardness results, W[1]-hardness results, W[2]-hardness results, and paraNP-hardness results), and numerous tractability results (P-results and FPT-results). The parameters examined in the paper include the number of predecessors/successors of the distinguished candidate in the agenda, the number of added/deleted voters/candidates, the number of registered voters/candidates, and the number of voters/candidates not deleted. Table 1 summarizes our main results. We also examine the parameterized complexity with respect to two natural parameters: the number of voters and the number of candidates. Many of the results concerning these parameters are either straightforward to observe or implicitly derived from the results for the aforementioned parameters, with only a few cases remaining open. For more details, we refer the reader to Sect. 2.4. Additionally, we note that beyond the results presented in Table 1, our hardness reductions also imply many kernelization and approximation lower bounds, which shall be discussed in detail in Sect. 4.

The general message conveyed by our study is that the amendment procedure and the successive procedure exhibit distinct behaviours regarding their resistance to the eight standard control problems. We will discuss this issue at the end of the paper in greater detail.

As a side note, for readers interested in the topic, it is worth mentioning that similar control problems have been investigated in many other settings, such as multiwinner voting (see, e.g., the works of Meir et al. [42], Yang [52]), judgment aggregation (see, e.g., the works of Baumeister et al. [5], Baumeister et al. [6]), group identification (see, e.g., the works of Erdélyi, Reger, and Yang [26], Yang and Dimitrov [55]), tournament solutions (see, e.g., the work of Brill, Schmidt-Kraepelin, and Suksompong [12]).

1.1 Motivation

Our motivation for the study is twofold.

First, the control types (operations) of adding/deleting voters/candidates are highly relevant for practical applications. In the context of parliamentary decision-making,

Table 1 A summary of the parameterized complexity of election control problems under the amendment procedure, the full-amendment procedure, and the successive procedure

	CCAV- τ	CCDV- τ	CCAC- τ	CCDC- τ
Amendment	$W[1]\text{-}h^\top$ [38] $W[2]\text{-}h^\perp$ (Thm. 1) $W[1]\text{-}h^\top$ [38] $W[1]\text{-}h^\perp$ (Thm. 11)	$W[1]\text{-}h^\top$ [38, 39] $W[2]\text{-}h^\perp$ (Thms. 2, 3) $W[1]\text{-}h^\top$ [38, 39] $W[2]\text{-}h^\perp$ (Thms. 12, 13)	immune^\top [4] P (Thm. 9) immune^\top [4] FPT^\leftarrow (Thm. 18) $W[2]\text{-}h^\perp$ (Thm. 17)	P (Thm. 10) P (Thm. 19)
Successive	FPT^\leftarrow (Cor. 7) $W[1]\text{-}h^\perp$ (Thm. 21) DCAV- τ FPT^\leftarrow (Cor. 4) $W[1]\text{-}h^\perp$ (Thm. 6) P^\top (Cor. 8) $W[1]\text{-}h^\top$ (Thm. 14) P (Cor. 8)	FPT^\leftarrow (Cor. 7) $W[2]\text{-}h^\perp$ (Thms. 22, 23) DCDV- τ FPT^\leftarrow (Cor. 4) $W[2]\text{-}h^\perp$ (Thms. 7, 8) P^\top (Cor. 8) $W[2]\text{-}h^\top$ (Thms. 15, 16) P (Cor. 8)	immune^\top (Cor. 9) $W[2]\text{-}h^\top$ (Thm. 26) DCAC- τ P (Cor. 5) P (Thm. 20) $W[2]\text{-}h^\top$ (Thm. 30) FPT^\rightarrow (Thm. 31)	$W[1]\text{-}h^\top$ (Thms. 27, 28) FPT^\rightarrow (Thm. 29) DCDC- τ immune^\top [4] P (Cor. 5) immune^\top [4] P (Cor. 6) immune^\top (Cor. 10) $W[1]\text{-}h^\top$ (Thms. 32, 33)

Results with the superscripts \top and \perp respectively mean that they hold when the distinguished candidate is the first and the last candidate in the agenda. Results with the superscript \top mean that they hold as long as the distinguished candidate is not the first one in the agenda. FPT^\leftarrow results with the superscripts \leftarrow and \rightarrow are respectively with respect to the number of predecessors and the number of successors of the distinguished candidate. All $W[1]$ -hardness and $W[2]$ -hardness results for CCAV- τ and DCAV- τ are with respect to the combined parameter of the number of added votes and the number of registered votes. All $W[1]$ -hardness and $W[2]$ -hardness results for CCDV- τ and DCDV- τ are with respect to both the number of deleted votes and the number of remaining votes. The $W[1]$ -hardness of CCDC-Successive and DCDC-Successive holds with respect to both the number of deleted candidates and the number of remaining candidates. All $W[2]$ -hardness results of CCAC- τ are with respect to the number of added candidates. The $W[2]$ -hardness of CCAC-Successive holds even if we add one additional restriction that there are only two registered candidates. The $W[2]$ -hardness of DCAC-Successive holds even when the distinguished candidate is the only registered candidate

for example, members of parliament have the right to abstain from voting. Deleting voters represents the scenario where a strategic controller may persuade particular voters to abstain, though these voters initially planned not to do so. On the other hand, adding voters represents a scenario where the strategic controller persuades certain voters who initially planned to abstain to participate in the voting procedure. Similarly, adding or deleting candidates is relevant in scenarios where a strategic controller has the power to decide which motions, bills, proposals, etc., are eligible to be put on the agenda.

Second, investigating the complexity of the aforementioned election control problems for many traditional voting procedures (those that do not require an agenda to determine a winner) had partially dominated the early development of computational social choice for several years, culminating in an almost complete complexity landscape of these problems (see the book chapters [7, 27] for important progress up to 2016, and the papers [25, 44, 50, 51, 53] for recent results). The motivation behind the extensive research is rooted in the belief that complexity serves as a robust barrier against strategic behavior. Given the broad applicability of the amendment and the successive procedures, we believe it is of great importance to address the gaps in this regard for the two procedures.

It is worth noting that several experimental studies have demonstrated that many computationally hard election problems can often be solved efficiently on practical instances, providing empirical evidence that mitigates the perception of complexity as a barrier to strategic voting. However, we emphasize that establishing the theoretical complexity of a problem is a foundational step toward a comprehensive understanding of its nature and limitations. This theoretical groundwork is indispensable for systematically addressing the challenges and opportunities presented by these problems in both theoretical and practical domains.

1.2 Organization

In Sect. 2, we provide important notations used in our study. Following this, we unfold our main results in Sect. 3, which is further divided into three subsections respectively covering our results for the h -amendment procedures, the $(m - h)$ -amendment procedures where m is the total number of candidates and h is a constant, and the successive procedure. Moving on, we discuss algorithmic lower bounds implied by our reductions in Sect. 4. Finally, in Sect. 5, we briefly conclude our results, discuss their implications on the usage of these procedures, and lay out several topics for future research.

2 Preliminaries

We assume that the reader has a basic understanding of (parameterized) complexity theory and graph theory. This section provides the necessary notations for our study, while terms not explicitly defined in the paper are referenced in the monographs edited by Bang-Jensen and Gutin [2] and the work of West [49]. For a gentle introduction to complexity theory, we recommend the paper by Tovey [48].

For an integer i , $[i]$ denotes the set of all positive integers less than or equal to i .

2.1 Elections

An election is defined as a tuple $E = (C, V)$, where C is a set of candidates and V is a multiset of votes cast by the voters. In particular, each $\succ \in V$ is a linear order on C , representing the preference of a corresponding voter over the candidates in C . Specifically, $a \succ b$ for $a, b \in C$ indicates that the voter prefers a to b . We also say that the vote \succ ranks a before b if $a \succ b$. For notational brevity, we sometimes write a preference in the format of a sequence of candidates from the most preferred one to the least preferred one. For instance, stating a vote with the preference $a \ b \ c$ means that, according to the vote, a is ranked before b , and b ranked before c . For two disjoint subsets $X, Y \subseteq C$, $X \succ Y$ means that all candidates in X are ranked before all candidates in Y in the vote \succ . For $C' \subseteq C$, $V|_{C'}$ is the multiset of votes obtained from those in V by removing all candidates in $C \setminus C'$. Thus, $(C', V|_{C'})$ is the election (C, V) restricted to C' . For notational convenience, we use (C', V) to denote $(C', V|_{C'})$.

A subset $C' \subseteq C$ is called a block with respect to V if its members are ranked consecutively in all votes from V , i.e., for each $\succ \in V$ and each $c \in C \setminus C'$, it holds that either $\{c\} \succ C'$ or $C' \succ \{c\}$. An ordered block on a subset C' of candidates is a linear order on C' such that C' is a block and, moreover, all votes of V restricted to C' are consistent with the linear order. For instance, if (a, b, c) is an ordered block, all votes rank $\{a, b, c\}$ together, and they all prefer a to b to c .

Let $n_V(a, b)$ denote the number of votes in V ranking a before b , i.e., $n_V(a, b) = |\{\succ \in V : a \succ b\}|$. We say that a beats b with respect to V if $n_V(a, b) > n_V(b, a)$, and that a ties with b with respect to V if $n_V(a, b) = n_V(b, a)$. For a candidate $a \in C$ and a subset $C' \subseteq C \setminus \{a\}$, we say that a majority-dominates C' with respect to V if there exists $V' \subseteq V$ such that $|V'| > |V|/2$, and $\{a\} \succ C'$ holds for all $\succ \in V'$. For uniformity, we define that every candidate $a \in C$ majority-dominates the empty set \emptyset .

A candidate $c \in C$ is the Condorcet winner of (C, V) if c beats every candidate from $C \setminus \{c\}$ with respect to V . A candidate c is a weak Condorcet winner of (C, V) if c is not beaten by any other candidate from C with respect to V . A voting procedure is Condorcet-consistent if it always selects the Condorcet winner of an election as the winner whenever the election admits a Condorcet winner.

An oriented graph is a directed graph (digraph) in which there is at most one arc between any two vertices. The majority graph of an election (C, V) is an oriented graph with vertex set C , where there is an arc from candidate a to candidate b if and only if a beats b with respect to V . An arc from vertex a to vertex b in a digraph is denoted by (a, b) . By assigning a weight of $n_V(a, b)$ to each arc (a, b) in the majority graph, we obtain the weighted majority graph of (C, V) .

An agenda is a linear order \triangleright on C . We use $\triangleright[i]$ to denote the i th candidate in \triangleright . Let $m = |C|$. For two integers i and j , we define $\triangleright[i, j] = \{\triangleright[x] : x \in [m], i \leq x \leq j\}$. Note that if $j > m$, then $\triangleright[i, j] = \triangleright[i, m]$, and if $i > j$, then $\triangleright[i, j] = \emptyset$. For a candidate $c \in C$, we call candidates before c in the agenda the predecessors of c , and call those after c her successors. In particular, for two candidates $\triangleright[i]$ and $\triangleright[j]$ such

that $j > i$, we call $\triangleright[j]$ the $(j - i)$ th-successor of $\triangleright[i]$, or say that $\triangleright[j]$ is a t_{\leq} th-successor of $\triangleright[i]$ where t is an integer such that $t \geq j - i$. The number of predecessors of c plus one is also called the position of c in the agenda \triangleright . The successive winner and the h -amendment winner with respect to (C, V) and \triangleright are determined as follows.

- **Successive**¹ Let $i \in [m]$ be the smallest integer such that $\triangleright[i]$ majority-dominates $\triangleright[i + 1, m]$ with respect to V . The successive winner is $\triangleright[i]$.
- **h -amendment** This procedure comprises several rounds, where in each round either the first candidate in the current agenda is eliminated, or up to h candidates following the first candidate in the agenda are eliminated. Specifically, in round r where $r = 1, 2, \dots$, if the first candidate $\triangleright[1]$ in the current agenda beats every candidate in $\triangleright[2, h + 1]$, then all candidates in $\triangleright[2, h + 1]$ are eliminated from \triangleright . Otherwise, $\triangleright[1]$ is eliminated from \triangleright . The procedure continues until only one candidate remains in the agenda; this candidate is then declared the winner. We emphasize that when a set $\triangleright[x, y]$ of candidates is eliminated, each $\triangleright[j]$ for $j > y$ is automatically shifted upward by $y - x + 1$ positions in \triangleright . As a result, the first candidate in the agenda is always denoted by $\triangleright[1]$. Furthermore, if the election involves at most $m \leq h$ candidates, the h -amendment winner is determined using the $(m - 1)$ -amendment procedure.

The amendment procedure is exactly the 1-amendment procedure.² The full-amendment procedure is the $(m - 1)$ -amendment procedure. Notice that as each round in the amendment procedure eliminates exactly one candidate, the amendment procedure takes $m - 1$ rounds. In contrast, the full-amendment procedure may take anywhere from 1 to $m - 1$ rounds depending on the situation.

From the definition, it is easy to see that the h -amendment procedure is Condorcet-consistent for all possible values of h . On the other hand, the successive procedure is not Condorcet-consistent (see the work of Miller [43] for detailed discussion). To verify this, consider an election with three votes: $a \ b \ c \ d$, $d \ a \ b \ c$, and $b \ c \ a \ d$. Obviously, a is the Condorcet winner, but with respect to the agenda (a, b, c, d) , the successive winner is b (a is not the successive winner because only one out of three votes ranks a before all her successors). Nevertheless, it is easy to see that if an election admits a Condorcet winner, then, as long as the Condorcet winner is the last one in the agenda, this Condorcet winner is also the successive winner of the election (see the work of Rasch [46] for detailed discussion). In addition, it holds that the h -amendment/successive winner c of each election is the Condorcet winner of the election restricted to c and all successors of c . As pointed out earlier, the full-amendment winner of an election with respect to an agenda beats both the amendment winner and the successive winner of the same election with respect to the same agenda, if they do not coincide. One can also observe that, in the same election, the full-amendment winner can be neither a successor of the amendment winner nor a successor of the successive winner.

¹ The successive procedure has also been studied under the name Euro-Latin procedure (see, e.g., the work of Farquharson [28]).

² The amendment procedure has also been studied under the names Anglo-American procedure, elimination procedure, and ordinary committee procedure (see, e.g., the works of Black [8], Farquharson [28]).

Given an oriented graph G with the vertex set C and an agenda \triangleright on C , the h -amendment/successive winner of G with respect to \triangleright is the h -amendment/successive winner of an election whose majority graph is G with respect to \triangleright . See Example 1 for an illustration.

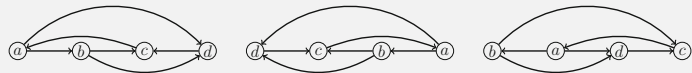
Example 1: An illustration of different voting procedures.

Consider an election (C, V) , where $C = \{a, b, c, d\}$, and V consists of three votes. The three votes in V (left side) and the majority graph of (C, V) together with three different agendas are given below.

vote 1: $b \ d \ c \ a$

vote 2: $c \ a \ b \ d$

vote 3: $a \ d \ b \ c$



With respect to the agenda (a, b, c, d) , b is the h -amendment winner for $h \in \{2, 3\}$, and d is both the amendment winner and the successive winner. With respect to the agenda (d, c, b, a) , a is both the successive winner and the h -amendment winner for all $h \in [3]$. With respect to the agenda (b, a, d, c) , c is the amendment winner, and d is the successive winner and the h -amendment winner for $h \in \{2, 3\}$.

We remark that, at first glance, it may seem that the $(h + 1)$ -amendment winner of an election with respect to an agenda cannot be a successor of the election's h -amendment winner under the same agenda. However, this is not the case, as the h -amendment winner may be eliminated by some of its predecessors (which may also be eliminated later) when the $(h + 1)$ -amendment procedure is applied. To illustrate this, consider the election derived from the one in Example 1 by introducing an additional candidate, e , who is ranked first in the second vote and immediately after d in the other two votes, such that e is beaten by d but beats both a and b . With respect to the agenda (e, a, d, b, c) , b is the amendment winner, while c is the 2-amendment winner.

2.2 Election Control Problems

Let τ be a voting procedure. We study eight standard control problems which are special cases of the problems defined below.

CONSTRUCTIVE MULTIMODE CONTROL for τ (CMC- τ)

Given: A set C of registered candidates, a distinguished candidate $p \in C$, a set D of unregistered candidates, a multiset V of registered votes over $C \cup D$, a multiset W of unregistered votes over $C \cup D$, an agenda \triangleright on $C \cup D$, and four nonnegative integers k_{AV} , k_{DV} , k_{AC} , and k_{DC} .

Question: Are there $C' \subseteq C \setminus \{p\}$, $D' \subseteq D$, $V' \subseteq V$, and $W' \subseteq W$ such that
 (1) $|C'| \leq k_{DC}$, $|D'| \leq k_{AC}$, $|V'| \leq k_{DV}$, $|W'| \leq k_{AV}$, and
 (2) p is the τ winner of $((C \setminus C') \cup D', (V \setminus V') \cup W')$ with respect to the agenda \triangleright restricted to $(C \setminus C') \cup D'$?

To put it in plain words, in the CMC- τ problem, an election controller attempts to make a given distinguished candidate the winner by deleting a limited number of

Table 2 Specifications of some election control problems. In the abbreviations, the letter X is either CC standing for “constructive control” or DC standing for “destructive control”

Abbreviations	Restrictions
XAV- τ	$k_{DV} = k_{AC} = k_{DC} = 0$, and $D = \emptyset$
XDV- τ	$k_{AV} = k_{AC} = k_{DC} = 0$, and $W = D = \emptyset$
XAC- τ	$k_{AV} = k_{DV} = k_{DC} = 0$, and $W = \emptyset$
XDC- τ	$k_{AV} = k_{DV} = k_{AC} = 0$, and $W = D = \emptyset$

For $X = CC$ the problems are special cases of CMC- τ , and for $X = DC$ the problems are special cases of DMC- τ . The second letters A and D in the abbreviations respectively stand for “adding” and “deleting”, and the third letters V and C respectively stand for “voters” and “candidates”

registered candidates and votes and adding a limited number of unregistered candidates and votes.

DESTRUCTIVE MULTIMODE CONTROL for τ (DMC- τ) is defined similar to CMC- τ with only the difference that the goal of the controller is to make the given distinguished candidate not the winner (i.e., we require in requirement (2) in the above definition that p is not the τ winner of $((C \setminus C') \cup D', (V \setminus V') \cup W')$).³

The eight standard control problems are special cases of CMC- τ and DMC- τ . The abbreviations of the problem names and their specifications are provided in Table 2. For simplicity, when we write an instance of a standard control problem, we omit its components of constant values 0 and \emptyset , and use k to denote the one in $\{k_{AV}, k_{DV}, k_{AC}, k_{DC}\}$ not requested to be zero. For example, an instance of CCAV- τ is written as $(C, p, V, W, \triangleright, k)$, where k represents k_{AV} .

To improve readability and avoid repetitive phrasing, we adopt the following simplifications in discussions concerning an election (C, V) restricted to a subset C' of candidates. (Such discussions are particularly relevant to election control problems involving the addition or deletion of candidates.) Instead of explicitly stating that a candidate c is the τ winner of (C, V) with respect to an agenda \triangleright restricted to C' , we simply say that c is the τ winner of (C', V) with respect to \triangleright . When the context makes the agenda clear, we may further simplify by stating that c is the τ winner of (C', V) .

In the paper, we establish numerous hardness results for restricted cases of the control problems where the distinguished candidate p occupies a specific position (e.g., the first or the last) in the agenda. Additionally, when we assert that a hardness result holds as long as p is not the first candidate in the agenda, we mean that the result holds when restricted to instances where p is in the i th position in the given agenda, for all positive integers $i \geq 2$.

A voting procedure is immune to a constructive control type if it is impossible to turn a nonwinning candidate into a winner by performing the corresponding controlling operation (adding/deleting voters/candidates). Analogously, a voting procedure is immune to a destructive control type if it is impossible to turn a winner into a nonwinning candidate by performing the corresponding control operation.

³ Many traditional voting procedures, such as Borda, Copeland, Maximin, etc., do not need an agenda to determine the winners. In these cases, \triangleright is dropped in the definitions of CMC- τ and DMC- τ .

2.3 Parameterized Complexity

A parameterized problem is a subset of $\Sigma^* \times \mathbb{N}$, where Σ is a finite alphabet. A complexity hierarchy has been developed to classify parameterized problems into numerous classes:

$$\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{XP}.$$

For a detailed discussion of these classes, we refer the reader to the monograph by Downey and Fellows [22].

For each $X \in \Sigma^*$, let $|X|$ denote the size of X . A parameterized problem is fixed-parameter tractable (FPT) if there is an algorithm which solves each instance (X, κ) of the problem in time $f(\kappa) \cdot |X|^{O(1)}$, where f can be any computable function in the parameter κ . A parameterized problem is in the class XP if there is an algorithm which solves each instance (X, κ) of this problem in time $|X|^{f(\kappa)}$, where f can be any computable function in the parameter κ .

Definition 1 (*Parameterized Reduction*) A parameterized problem P is parameterized reducible to a parameterized problem Q if there is an algorithm which takes as input an instance (X, κ) of P and outputs an instance (X', κ') of Q such that

- the algorithm runs in time $f(\kappa) \cdot |X|^{O(1)}$;
- (X, κ) is a Yes-instance of P if and only if (X', κ') is a Yes-instance of Q ; and
- $\kappa' \leq g(\kappa)$ for some computable function g in κ .

A parameterized problem is W[1]-hard (respectively, W[2]-hard) if all problems in W[1] (respectively, W[2]) are parameterized reducible to this problem. A W[1]-hard (respectively, W[2]-hard) problem that is in W[1] (respectively, W[2]) is W[1]-complete (respectively, W[2]-complete). W[1]-hard and W[2]-hard problems do not admit any FPT-algorithms unless the above hierarchy collapses to some level. To prove that a new problem is W[1]-hard (respectively, W[2]-hard), one needs to select a known W[1]-hard (respectively, W[2]-hard) problem and establish a parameterized reduction from the known problem to the new one.

Beyond the class of XP, there exists another significant class of parameterized problems known as paraNP-hard. A parameterized problem is classified as paraNP-hard if it remains NP-hard for some fixed value of the parameter. For a more comprehensive exploration of parameterized complexity theory, we recommend the textbooks by Cygan et al. [18], Downey and Fellows [22], and Niedermeier [45].

2.4 Parameters We Consider

Our results for the eight election control problems encompass the following natural parameters.

Solution size k The parameter k , representing the number of voters or candidates allowed to be added or deleted, is especially relevant in scenarios where the election controller faces resource constraints or seeks to minimize the exertion of

influence. For instance, consider a university senate deciding on a critical policy change, such as revising tuition fees. In this scenario:

- Adding voters corresponds to persuading external experts, alumni, or student representatives who were not initially planning to participate, to join the voting process. However, this might entail significant communication efforts, logistical arrangements, or financial costs (e.g., compensating for their time or travel expenses).
- Deleting voters may reflect attempts to convince certain stakeholders (e.g., administrators or faculty members) to abstain from voting, requiring delicate negotiations and justification to avoid reputational damage or conflict.

Suppose the election controller supports the policy change but operates under a limited budget, allowing them to influence the participation of at most k voters. Here, minimizing k aligns with practical constraints such as limited time to communicate with potential participants, a fixed budget for compensating or incentivizing voters, and the controller's preference to avoid significant disruptions or visible interventions.

In another example, consider a national election where k represents the number of candidates to be added or removed from the ballot to favor a preferred candidate. Adding candidates may involve recruiting new participants, which can be administratively challenging and resource-intensive. Similarly, removing candidates might require legal actions, political negotiations, or significant financial resources, all of which are constrained by k .

We show that most hard control problems exhibit fixed-parameter intractability concerning this parameter. Notably, for the problems of election control by adding voters/candidates, the intractability persists even when combined parameterized by the solution size and the number of registered voters/candidates, or when there is only one or two registered candidates.

Number of voters/candidates not deleted This parameter becomes particularly relevant in scenarios where the election controller is tasked with selecting a small subset of voters/candidates to participate in the decision-making procedure, with the unselected voters/candidates effectively being considered as deleted. For example, consider the context of corporate board elections or committee decisions in organizations. In such settings, the controller might wish to identify a small representative group of voters—such as key stakeholders, senior employees, or domain experts—who will cast votes on behalf of a larger group, thereby streamlining the decision-making process. Here, the parameter captures the practical constraint of selecting a manageable number of voters while ensuring that the election outcome remains aligned with the controller's objectives.

For problems involving voter or candidate deletions, our results demonstrate that these hardness control problems are fixed-parameter intractable with respect to this parameter. This intractability underscores the challenge of achieving the controller's goals when the number of retained voters or candidates is tightly constrained, reinforcing the significance of understanding such scenarios both theoretically and in practice.

Number of predecessors/successors of the distinguished candidate Previous works indicate that candidates may benefit from their positions on the agenda. For

example, in a scenario with three candidates and a decision determined by the amendment procedure, it becomes evident that the candidate listed last on the agenda is more likely to be selected.⁴ Hence, it is intriguing to explore if the control problem faced by the election controller becomes more manageable when the distinguished candidate has only a few predecessors or successors. Our findings demonstrate that concerning this parameter, certain problems are fixed-parameter intractable, while others are FPT.

For the h -amendment and the $(m - h)$ -amendment procedures, we also analyze the parameter h . We prove that many control problems remain computationally hard even for constant values of h , thereby identifying their classification within the hierarchy of paraNP-hard problems with respect to h .

Two additional natural parameters are the number m of candidates and the number n of voters. It is straightforward to verify that the candidate control problems (CCAC, CCDC, DCAC, DCD) are FPT with respect to m . For the voter control problems parameterized by m , one of our results (Theorem 4) establishes their FPT status. Detailed discussions on this matter will follow the presentation of Theorem 4. Consequently, our investigation implicitly provides a comprehensive understanding of the parameterized complexity of the eight standard control problems for the parameter m .

Regarding the parameter n , it is easy to see that the voter control problems (CCAV, CCDV, DCAV, DCDV) are FPT. Furthermore, several polynomial-time solvability results for these problems are summarized in Table 1. However, beyond these tractability results, the FPT status of certain candidate control problems parameterized by n remains an open question. Specifically, this includes CCAC-Full-Amendment, CCAC-Successive, CCDC-Successive, DCAC-Successive, and DCD-Successive. We note that a systematic study of the candidate control problems with respect to n for many traditional voting procedures has been conducted by Chen et al. [14].

2.5 Auxiliary Problems

We elaborate on problems used for establishing our hardness results. Let $G = (U, F)$ be a simple undirected graph. For a vertex $u \in U$, $N_G(u) = \{u' \in U : \{u, u'\} \in F\}$ is the set of neighbors of u in G , and $N_G[u] = N_G(u) \cup \{u\}$ is the set of closed neighbors of u in G . For $U' \subseteq U$, let $N_G(U') = \bigcup_{u \in U'} N_G(u) \setminus U'$, and let $N_G[U'] = N_G(U') \cup U'$. In addition, $G - U'$ is the graph obtained from G by deleting all vertices in U' . Two vertices are adjacent if there exists an edge between them. An independent set of G is a subset of vertices that are pairwise nonadjacent. A clique of G is a subset of pairwise adjacent vertices. A vertex u dominates a vertex u' if either they are the same vertex or they are adjacent. For two subsets $S, S' \subseteq U$, we say that S dominates S' if every vertex in S' is dominated by at least one vertex in S . A dominating set of G is a subset of vertices which dominate all vertices of G . A perfect code of G is a dominating set P such that every vertex in G is dominated by exactly one vertex in P , i.e., P is a dominating set of G such that for every two distinct $u, u' \in P$ it holds that $N_G[u] \cap N_G[u'] = \emptyset$. Note that every perfect code is also an independent set.

⁴ If a Condorcet winner exists, the Condorcet winner will be selected regardless of her position. In all other cases, the procedure consistently designates the last candidate on the agenda as the winner.

For two positive integers i and j , $K_{i,j}$ denotes the complete bipartite graph with i vertices in one component and j vertices in the other component of the bipartition of $K_{i,j}$.

PERFECT CODE

Given: A graph G and an integer κ .

Question: Does G have a perfect code of size κ ?

CLIQUE

Given: A graph G and an integer κ .

Question: Does G have a clique of size κ ?

BICLIQUE

Given: A bipartite graph G and an integer κ .

Question: Does G contain a complete bipartite graph $K_{\kappa,\kappa}$ as a subgraph?

RED- BLUE DOMINATING SET (RBDS)

Given: A bipartite graph G with the vertex bipartition (R, B) and an integer κ .

Question: Is there a subset $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G ?

In the above definition, we call vertices in R red vertices and call those in B blue vertices.

It is known that PERFECT CODE, CLIQUE, BICLIQUE, and RBDS are NP-complete and, moreover, with respect to κ , PERFECT CODE, CLIQUE, and BICLIQUE⁵ are W[1]-complete (see, e.g., the works of Cesati [13], Downey and Fellows [21], Lin [37]), and RBDS is W[2]-complete (see, e.g., the works of Downey, Fellows, and Stege [23], and Garey and Johnson [32]).

We note that in several of our reductions where the RBDS problem is used, we make the following assumptions:

- (1) $|B| > \kappa > 1$,
- (2) the graph G does not contain any isolated vertices (i.e., vertices without any neighbors),

⁵ BICLIQUE is originally defined over general graphs, but we use its restricted version to bipartite graphs to establish several of our W[1]-hardness results. Whether BICLIQUE is W[1]-hard had remained as a significant long-standing open question until Lin [37] resolved it in the affirmative in 2015. More importantly, Lin's reduction [37, Theorem 1.1, Corollary 1.3] applies to the special case where the input graph is a bipartite graph.

- (3) all vertices in R have the same degree, which we denote ℓ , and
- (4) $\ell + \kappa \leq |B|$.

These assumptions do not affect the $W[2]$ -hardness of the problem with respect to κ . The first assumption, $|B| > \kappa > 1$, is straightforward, as otherwise the problem is polynomial-time solvable. The second assumption is also easy to verify. If an instance does not satisfy the third assumption, we can obtain an equivalent instance by the following operation: Let ℓ be the maximum degree of vertices in R . For each red vertex $r \in R$ whose degree is strictly smaller than ℓ , we add new degree-1 vertices adjacent only to r until r reaches degree ℓ . An important observation is that there is an optimal solution (a subset $B' \subseteq B$ dominating R with the minimum cardinality) of the new instance that does not contain any of the newly introduced degree-1 vertices. This ensures the equivalency of the two instances. Lastly, note that if $\ell + \kappa > |B|$, the given instance must be a Yes-instance (any subset of κ blue vertices dominates all red vertices).

2.6 Some General Notations Used in Our Reductions

In our hardness reductions, we will consistently use the following notations. For a set X , we use \vec{X} to denote an arbitrary but fixed linear order on X , unless stated otherwise. For a subset $X' \subseteq X$, $\vec{X}[X']$ denotes the order \vec{X} restricted to X' . In addition, $\vec{X} \setminus X'$ is \vec{X} without the elements of X' . For two linear orders \vec{X} and \vec{Y} , the notation (\vec{X}, \vec{Y}) represents their concatenation, where \vec{X} precedes \vec{Y} . For example, if $\vec{X} = (a, b, c)$ and $\vec{Y} = (1, 2, 3)$, then $(\vec{X}, \vec{Y}) = (a, b, c, 1, 2, 3)$.

2.7 Some Remarks

All the parameterized reductions derived in the paper run in polynomial time. Since all eight standard control problems under the h -amendment and the successive procedures are in NP, in the following sections, if we prove that a problem is $W[1]$ -hard or $W[2]$ -hard, it implies that the problem is also NP-complete. Particularly, for all our new $W[1]$ -hardness or $W[2]$ -hardness results shown in Table 1, the corresponding problems are NP-complete. To avoid repetitiveness, we will not assert these implied results in the corresponding theorems.

3 Our Results

In this section, we study the complexity of the eight standard election control problems. In particular, we explore how the position of the distinguished candidate in the given agenda shapes the parameterized complexity landscape of the problems. Our exploration starts with the following lemma.

Lemma 1 *Let $E = (C, V)$ be an election, and let \triangleright be an agenda on C . For all positive integers h , $\triangleright[1]$ is the h -amendment winner of E with respect to \triangleright if and only if $\triangleright[1]$ is the Condorcet winner of E .*

Proof Let $E = (C, V)$ and \triangleright be as stipulated in the lemma. Suppose $\triangleright[1]$ is the Condorcet winner of E , i.e., $\triangleright[1]$ beats every other candidate with respect to V . Consequently, in each round of the h -amendment procedure, all candidates ranked after $\triangleright[1]$ (up to h in number) are eliminated. This ensures that $\triangleright[1]$ remains the h -amendment winner of E with respect to \triangleright .

Conversely, assume that $\triangleright[1]$ is the h -amendment winner of E . By the definition of the h -amendment procedure, $\triangleright[1]$ is not eliminated in any round. This implies that $\triangleright[1]$ beats every other candidate with respect to V , establishing that $\triangleright[1]$ is the Condorcet winner of E . \square

It is known that CCAV-Condorcet and CCDV-Condorcet are NP-complete (see the work of Bartholdi, Tovey, and Trick [4]). Moreover, CCAV-Condorcet is $W[1]$ -hard with respect to the number of registered votes plus the number of added votes, CCDV-Condorcet is $W[1]$ -hard with respect to the number of deleted votes (see the work of Liu et al. [38]), CCDV-Condorcet is $W[2]$ -hard with respect to the number of votes not deleted (see the work of Liu and Zhu [39])⁶. These results and Lemma 1 offer us the following corollary.

Corollary 1 [38, 39] *For any h -amendment procedure τ , the following hold:*

- CCAV- τ is $W[1]$ -hard with respect to $|V| + k$, the number of registered votes plus the maximum number of unregistered votes allowed to be added.
- CCDV- τ is $W[1]$ -hard with respect to k , the maximum number of votes that can be deleted.
- CCDV- τ is $W[2]$ -hard with respect to $|V| - k$, the minimum number of votes that cannot be deleted.

Moreover, the above results hold even when the distinguished candidate is ranked first on the agenda.

In addition, as DCAV-Condorcet and DCDV-Condorcet are polynomial-time solvable (see the work of Bartholdi, Tovey, and Trick [4]), we can deduce the following corollary concerning destructive control by adding or deleting voters.

Corollary 2 [4] *For τ being any h -amendment procedure, DCAV- τ and DCDV- τ are polynomial-time solvable when the distinguished candidate is the first one in the agenda.*

Regarding candidate control, it is known that Condorcet is immune to CCAC and DCDC, and CCDC-Condorcet and DCAC-Condorcet are polynomial-time solvable (see the work of Bartholdi, Tovey, and Trick [4]). The following results follow.

⁶ Although not explicitly stated in the paper of Liu and Zhu [39] that CCDV-Condorcet is $W[2]$ -hard with respect to the number of votes not deleted, the reduction in the proof of Theorem 1 in [39] for a $W[2]$ -hardness of CCDV-Maximin can be directly used to show the result. Particularly, in the proof of Theorem 1 in [39], the authors established an instance of CCDV-Maximin from the $W[2]$ -hard problem DOMINATING SET. They showed that if the instance of DOMINATING SET is a Yes-instance, there are $2\kappa + 1$ votes in the constructed election with respect to which the distinguished candidate beats all the other candidates, where κ is the parameter of the DOMINATING SET instance. For the other direction, they showed that if there are $2\kappa + 1$ votes with respect to which the distinguished candidate is the Maximin winner, the Maximin winner is the Condorcet winner.

Corollary 3 [4] *Let τ be an h -amendment procedure. If the distinguished candidate is the first one in the agenda, then τ is immune to CCAC and DCDC, and CCDC- τ and DCAC- τ are polynomial-time solvable.*

In the following subsections, we show that some of the P-results in Corollary 3 hold irrespective of the position of the distinguished candidate, while some others, although generally intractable, are FPT when parameterized by the number of predecessors of the distinguished candidate.

We note that throughout the paper, for all results regarding CCAC, the position of the distinguished candidate is counted with respect to candidates in $C \cup D$.

The remainder of this section focuses on cases not covered by the above corollaries. We begin by examining the h -amendment procedures and the $(m - h)$ -amendment procedures (Sects. 3.1 and 3.2). Following this, we explore the successive procedure (Sect. 3.3). Within each section, we first consider control by adding or deleting voters and then study control by adding or deleting candidates. We note that h is assumed to be positive and integral. For readability, this assumption will not be repeated throughout the discussion.

3.1 The h -Amendment Procedure for Constant h

By Corollary 1, CCAV- h -Amendment is intractable if the distinguished candidate is the first one in the agenda. In the following, we show a similar result for the case where the distinguished candidate is the last one in the agenda.

Theorem 1 *For every constant h , CCAV- h -Amendment is W[2]-hard with respect to the number of added votes plus the number of registered votes. This holds even if the distinguished candidate is the last candidate in the agenda.*

Proof We first prove the theorem for the case where $h = 1$ (i.e., the amendment procedure) using a reduction from RBDS. Subsequently, we demonstrate how to modify the reduction for other constant values of h .

Let (G, κ) be an instance of RBDS, where G is a bipartite graph with bipartition (R, B) . We construct an instance of CCAV-Amendment as follows. For each vertex in R , we create a corresponding candidate, denoted by the same symbol for simplicity. Additionally, we introduce two candidates, q and p , designating p as the distinguished candidate. Let $C = R \cup \{p, q\}$, and let $m = |C|$ denote the total number of candidates. We create $\kappa + 3$ registered votes as follows:

- $\kappa + 1$ votes with the preference $q \xrightarrow{R} p$;
- one vote with the preference $p \xrightarrow{R} q$; and
- one vote with the preference $\xrightarrow{R} p \ q$.

Let V be the multiset of the registered votes created above. Let $\triangleright = (q, \xrightarrow{R}, p)$ be an agenda, where q and p are the first and last candidates, respectively, and \xrightarrow{R} represents the ordered set of candidates between q and p . Figure 1 depicts the weighted majority graph of (C, V) and the agenda.

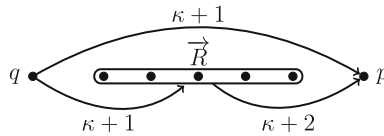


Fig. 1 The weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 1. All arcs among vertices in R are forward arcs with a uniform weight of $\kappa + 3$. The agenda is represented by the left-to-right ordering of the vertices

The unregistered votes are created according to the blue vertices. Precisely, for each blue vertex $b \in B$, we create one vote \succ_b with the preference

$$(\vec{R} \setminus N_G(b)) \succ_b p \succ_b q \succ_b (\vec{R}[N_G(b)]).$$

Let $W = \{\succ_b : b \in B\}$. Let $k = \kappa$. The instance of CCAV- AMENDMENT is $(C, p, V, W, \triangleright, k)$. We prove the correctness of the reduction as follows.

(\Rightarrow) Assume that there exists a subset $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . Let $W' = \{\succ_b : b \in B'\}$ be the multiset of the κ unregistered votes corresponding to B' . Let $E = (C, V \cup W')$. We show that p is the amendment winner of E with respect to \triangleright . First, as B' dominates R , from the above construction, for every candidate $r \in R$, the vote \succ_b , where $b \in B'$ and b dominates r , ranks q before r . Therefore, in E , at least $\kappa + 2$ votes rank q before r . This means that q beats all candidates from R in E . As q is the first candidate of the agenda, q is the winner of the $(m - 1)$ th round. Second, as all unregistered votes rank p before q , p beats q in E . Consequently, p is the winner of the last round and hence is the amendment winner of E with respect to \triangleright .

(\Leftarrow) Assume that there is a subset $W' \subseteq W$ of at most k unregistered votes such that p wins $E = (C, V \cup W')$ with respect to \triangleright . Observe first that $|W'| \geq k - 1$, since otherwise q still beats all the other candidates and hence remains as the winner in E , a contradiction. Second, observe that no matter which at most k votes are contained in W' , every red vertex beats all her successors in E . Let $B' = \{b \in B : \succ_b \in W'\}$ be the set of blue vertices corresponding to W' . We claim that B' dominates R in G . Assume, for the sake of contradiction, that there exists a red vertex $r \in R$ not dominated by any vertex from B' . From the construction of the unregistered votes, it follows that all votes in W' rank r before q . Consequently, there are exactly $2 + |W'|$ votes in $V \cup W'$ that rank r before q , which implies that r beats or ties with q in E . Let r' denote the leftmost red vertex in the agenda that beats or ties with q in E . Since r' beats all of its successors, r' wins E . However, this contradicts the assumption that p is the winner of E with respect to \triangleright . Therefore, B' dominates R in G .

Now we show how to modify the above reduction for the case where $h > 1$. For each candidate corresponding to a red vertex in R , if we make $h - 1$ copies of this candidate and replace the candidate with a fixed order of these newly created candidates in the above votes, the reduction works for other values of h . Precisely, for each red vertex $r \in R$, we create an ordered block of h candidates, and we replace the candidate created for r with this ordered block in all created votes. By using similar arguments,

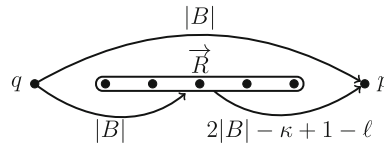


Fig. 2 The weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 2. All arcs among the vertices in R are forward arcs with a weight of at least $|B| - \kappa + 1$. The agenda is represented by the left-to-right ordering of the vertices

one can check that the RBDS instance is a Yes-instance if and only if we can make p the h -amendment winner by adding at most k unregistered votes. \square

For constructive control by deleting voters, we have a similar result.

Theorem 2 *For every constant h , CCDV- h -Amendment is W[2]-hard with respect to the number of deleted votes. This holds even if the distinguished candidate is the last candidate in the agenda.*

Proof Similar to the proof of Theorem 1, we provide a detailed proof for the case where $h = 1$, based on a reduction from RBDS. We then describe how to adapt this reduction for other values of h .

Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . We make the following assumptions which do not change the W[2]-hardness of RBDS, as discussed in Sect. 2.5. First, we assume that $|B| > \kappa > 1$. Second, we assume that G does not contain any isolated vertices. Third, we assume that all vertices of R have the same degree ℓ for some positive integer $\ell \leq |B|$. Lastly, we assume that $\ell + \kappa \leq |B|$. We construct an instance $(C, p, V, \triangleright, k)$ of CCDV-Amendment as follows.

The candidate set and the agenda are exactly the same as in the proof of Theorem 1, i.e., $C = R \cup \{p, q\}$ and $\triangleright = (q, \vec{R}, p)$. We create $2|B| - \kappa + 1$ votes in V :

- $|B| - \kappa + 1 - \ell$ votes with the preference $\vec{R} \ p \ q$;
- ℓ votes with the preference $p \ q \ \vec{R}$; and
- for each $b \in B$, one vote \succ_b with the preference

$$\left(\vec{R}[N_G(b)] \right) \ q \ \left(\vec{R} \setminus N_G(b) \right) \ p.$$

Let $V_B = \{\succ_b : b \in B\}$. Figure 2 depicts the weighted majority graph of (C, V) and the agenda.

Let $k = \kappa$. One can check that under the assumptions $\kappa \leq 1$ and $\ell + \kappa < |B|$, all candidates beat p , implying that p cannot be the amendment winner of (C, V) . We show the correctness of the reduction as follows.

(\Rightarrow) Suppose that there is a subset $B' \subseteq B$ of cardinality κ which dominates R in G . Let $V' = \{\succ_b : b \in B'\}$, and let $E = (C, V \setminus V')$. We show below that in E , q beats all candidates from R , but is beaten by p , and hence p becomes the winner in E . First, let r be any arbitrary candidate of R . In V there are exactly $\ell + (|B| - \ell) = |B|$ votes ranking q before r . As B' dominates R , there exists at least one $b \in B'$ such

that b dominates r . According to the construction of the votes, the \succ_b , which is from V' , ranks r before q , implying that there are at most $\kappa - 1$ votes in V' ranking q before r . As a result, in $V \setminus V'$ there are at least $|B| - \kappa + 1$ votes ranking q before r . As $|V \setminus V'| = 2|B| - 2\kappa + 1$, it holds that q beats r in E . As all votes in V' rank q before p , there are exactly $|B| - \kappa + 1$ votes in $V \setminus V'$ ranking p before q , implying that p beats q in E .

(\Leftarrow) Assume that there is a $V' \subseteq V$ such that $|V'| \leq \kappa$ and p wins $E = (C, V \setminus V')$ with respect to \triangleright . Observe first that as $\ell + \kappa \leq |B|$, no matter which at most κ votes are in V' , every candidate from R beats all of its successors with respect to $V \setminus V'$. Therefore, it must be that q beats all candidates of R , and p beats or ties with q in E . The latter implies that $|V'| \geq \kappa - 1$ and $V' \subseteq V_B$. Let $B' = \{b \in B : \succ_b \in V'\}$ be the set of blue vertices corresponding to V' . We claim that B' dominates R in G . Suppose, for the sake of contradiction, that there exists $r \in R$ not dominated by any vertex from B' . From the construction of the votes, q is ranked before r in all votes in V' . So, there are $(|B| - \kappa + 1 - \ell) + \ell = |B| - \kappa + 1$ votes ranking r before q in E , contradicting that q beats r in E . So, we know that B' dominates R .

To prove the $W[2]$ -hardness for other values of h , we modify the above reduction in the same way as we did in the proof of Theorem 1. That is, we replace in each of the above created vote the candidate created for r with an ordered block of h candidates.

□

Now we show that the $W[2]$ -hardness of CCDV- h -Amendment remains with respect to the dual parameter of the solution size.

Theorem 3 *For every constant h , CCDV- h -Amendment is $W[2]$ -hard with respect to the number of votes not deleted. This holds even if the distinguished candidate is the last one in the agenda.*

Proof We prove the theorem via reductions from RBDS. We consider first the case where $h = 1$. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . We construct an instance of CCDV-Amendment as follows. Let $C = R \cup \{q, q', p\}$, and let $\triangleright = (q, q', \vec{R}, p)$. We create the following $|B| + \kappa + 1$ votes:

- a multiset V_1 of κ votes, each with the preference $p \ q \ \vec{R} \ q'$;
- a singleton V_2 of one vote with the preference $\vec{R} \ p \ q \ q'$; and
- for each blue vertex $b \in B$ one vote \succ_b with the preference

$$q' \ (\vec{R} \setminus N_G(b)) \ q \ (\vec{R}[N_G(b)]) \ p.$$

For a given $B' \subseteq B$, let $V_{B'} = \{\succ_b : b \in B'\}$ be the multiset of votes created for vertices in B' . Let $V = V_1 \cup V_2 \cup V_{B'}$. In total, we have $|B| + \kappa + 1$ votes. Let $k = |B| - \kappa$. The instance of CCDV-Amendment is $(C, p, V, \triangleright, k)$. Observe that after deleting at most k votes from the election (C, V) , at least $2\kappa + 1$ votes remain. In the following, we establish the correctness of the reduction.

(\Rightarrow) Assume that there is a $B' \subseteq B$ of cardinality κ such that B' dominates R in G . Let $V' = V_1 \cup V_2 \cup V_{B'}$, and let $E = (C, V')$. Recall that $|V_1| + |V_2| = \kappa + 1$. We

claim that p is the amendment winner of E with respect to \triangleright . Clearly, q beats q' in E . For each $r \in R$, let $b \in B'$ be a vertex dominating r in G . From the construction of the votes, q is ranked before r in \succ_b . Therefore, there are at least $|V_1| + 1$ votes in V' ranking q before r , implying that q beats r in E . This ensures that all candidates in $R \cup \{q'\}$ are eliminated before the last round. Furthermore, since all $\kappa + 1$ votes in $V_1 \cup V_2$ rank p before q , q is eliminated in the last round, leaving p as the amendment winner of E . Hence, the constructed instance of CCDV-Amendment is a Yes-instance.

(\Leftarrow) Suppose that there is a $V' \subseteq V$ of at least $2\kappa + 1$ votes such that p is the amendment winner of (C, V') with respect to \triangleright . Let $E = (C, V')$. Observe that V' cannot contain more than $\kappa + 1$ votes from V_B , since otherwise q' beats all the other candidates and hence it is impossible that p is the amendment winner of E . If V' contains exactly $\kappa + 1$ votes from V_B , then either q' is the amendment winner of E (when V' contains exactly κ votes from $V_1 \cup V_2$), or someone in R is the amendment winner (when $(V_1 \cup V_2) \subseteq V'$), both of which contradict the winning of p in E . Then, as $|V'| \geq 2\kappa + 1$, and $|V_1| + |V_2| = \kappa + 1$, it must hold that V' contains exactly κ votes from V_B . As a consequence, V' contains all the $\kappa + 1$ votes from $V_1 \cup V_2$. Let $V_{B'}$, $B' \subseteq B$, be the intersection of V' and V_B . We claim that B' dominates R in G . For the sake of contradiction, assume that there is a red vertex $r \in R$ not dominated by any vertex from B' . Then, by the construction of the votes, all votes in $V_{B'}$ rank r before q , implying that r beats q in E . Moreover, it is easy to see that q beats q' . By the definition of the agenda \triangleright , this implies that the winner of the second-to-last round is from R . However, as everyone in R beats p in E , this contradicts the winning of p in E . Since B' dominates R in G , and $|B'| = |V_{B'}| = \kappa$, the RBDS instance is a Yes-instance.

To prove the $W[2]$ -hardness for other values of h , we modify the above reduction by replacing each candidate in $R \cup \{q'\}$ with an ordered block of h candidates. \square

Now we study destructive control by adding/deleting voters. Particularly, we extend the P-results for the h -amendment procedures in Corollary 2 to a unified FPT-algorithm, considering the number of predecessors of the distinguished candidate in the agenda. To achieve this, we examine an intermediate problem and formulate it using integer linear programming (ILP).

MAJORITY GRAPH CONTROL BY EDITING VOTERS (MGCEV)

- Given:** A set C of candidates, two multisets V and W of votes over C , an oriented graph G with the vertex set C , and two integers k and k' .
- Question:** Are there $V' \subseteq V$ and $W' \subseteq W$ such that $|V'| = k$, $|W'| = k'$, and the majority graph of $(C, V' \cup W')$ is exactly G ?
-

The following lemma is well-known.

Lemma 2 [31, 35, 36] *ILP is FPT with respect to the number of variables.*

Based on Lemma 2, we can show the following result.

Theorem 4 *MGCEV is FPT with respect to the number of candidates.*

Proof Let $I = (C, V, W, G, k, k')$ be an instance of MGCEV. Let $m = |C|$ denote the number of candidates. Let \mathcal{L} be the set of all linear orders on C . It holds that $|\mathcal{L}| = m!$. For each $\succ \in \mathcal{L}$, we use n_V^\succ (respectively, n_W^\succ) to denote the number of votes in V (respectively, W) with the preference \succ . For each $\succ \in \mathcal{L}$, we introduce two nonnegative integer variables, x_\succ and y_\succ , which represent the number of votes in V and W , respectively, with the preference \succ that are included in a desired feasible solution. The constraints are as follows.

- As we aim to identify two multisets $V' \subseteq V$ and $W' \subseteq W$, with cardinalities k and k' , respectively, we have the following constraints: $\sum_{\succ \in \mathcal{L}} x_\succ = k$ and $\sum_{\succ \in \mathcal{L}} y_\succ = k'$.
- To ensure that the majority graph of $(C, V' \cup W')$ is exactly G , we create the following two classes of constraints:
 - for every arc from a candidate c to another candidate c' in the graph G , we have that

$$\sum_{\succ \in \mathcal{L}: c \succ c'} x_\succ + \sum_{\succ \in \mathcal{L}: c \succ c'} y_\succ > \frac{k + k'}{2},$$

- for every two candidates c and c' in G without an arc between them, we require that

$$\sum_{\succ \in \mathcal{L}: c \succ c'} x_\succ + \sum_{\succ \in \mathcal{L}: c \succ c'} y_\succ = \frac{k + k'}{2}.$$

- Lastly, for each variable x_\succ and y_\succ , we have that $n_V^\succ \geq x_\succ \geq 0$ and $n_W^\succ \geq y_\succ \geq 0$.

This completes the construction of the ILP instance. Clearly, the given instance I is a Yes-instance if and only if the above ILP has a feasible solution. The theorem follows from Lemma 2 and the fact that the number of variables of the above ILP is bounded from above by a function in m . \square

Before presenting our parameterized results concerning the number of predecessors of the distinguished candidate, we note that Theorem 4 can be straightforwardly leveraged to derive FPT-algorithms for the four standard voter control problems under the h -amendment procedures and the successive procedure, with respect to the number m of candidates. To verify this, first observe that the majority graph of an election and an agenda suffices to determine the winner. Based on this observation, solving a voter control problem can be reduced to enumerating all oriented graphs over the candidate set as vertices, such that the distinguished candidate p is either the winner (in the case of constructive control) or not the winner (in the case of destructive control) for the enumerated graph and the given agenda. For each enumerated graph, we verify whether it is possible to add or delete a given number of voters so that the majority graph of the resulting election matches the enumerated graph. This verification is equivalent to solving the MGCEV problem. Theorem 4 guarantees that this can be performed in FPT-time with respect to m . The original instance is a Yes-instance if

and only if at least one enumeration yields a Yes-instance of the MGCEV problem. Since the number of oriented graphs with m vertices is bounded above by a function of m , it follows that the voter control problems are FPT when parameterized by m .

Returning to our main focus, we now present stronger FPT-results for the destructive voter control problems by leveraging Theorem 4. Specifically, we establish the fixed-parameter tractability of a generalized problem that subsumes both the DCAV problem and the DCDV problem as special cases. This result is parameterized by the number of predecessors of the distinguished candidate, which is strictly smaller than m .

EXACT DESTRUCTIVE CONTROL BY EDITING VOTERS FOR τ (E- DCEV- τ)

- Given:** A set C of candidates, a distinguished candidate $p \in C$, a multiset V of registered votes over C , a multiset W of unregistered votes over C , an agenda \triangleright on C , and two nonnegative integers k and k' .
- Question:** Are there $V' \subseteq V$ and $W' \subseteq W$ such that $|V'| = k$, $|W'| = k'$, and p is not the τ winner of $(C, V' \cup W')$ with respect to \triangleright ?
-

Theorem 5 *E- DCEV- h -Amendment is FPT with respect to h plus the number of predecessors of the distinguished candidate.*

Proof Let $I = (C, p, V, W, \triangleright, k, k')$ be an instance of E- DCEV- h -Amendment. Let ℓ be the number of predecessors of p in the agenda \triangleright . Therefore, $p = \triangleright[\ell + 1]$. Let $\ell' = \min\{\ell + h + 1, m\}$. In the following, we derive an algorithm running in FPT-time in ℓ' .

Observe that p is not the h -amendment winner of an election (C, U) where $U \subseteq V \cup W$ with respect to \triangleright if and only if one of the following two cases occurs:

- (1) p is not the h -amendment winner of (C, U) restricted to $\triangleright[1, \ell']$ (with respect to the agenda \triangleright restricted to $\triangleright[1, \ell']$).
- (2) p is the h -amendment winner of (C, U) restricted to $\triangleright[1, \ell']$, but some candidate after $\triangleright[\ell']$ in the agenda beats or ties with p (in this case it must be that $\ell' < m$).

For the first case, only the head-to-head comparisons among candidates in $\triangleright[1, \ell']$ matter. For the second case, in addition to $\triangleright[1, \ell']$, we need to guess one candidate $q \in \triangleright[\ell' + 1, m]$ who is supposed to beat or tie with p in the final election. Once such a candidate q is fixed, we need only to focus on the head-to-head comparisons among candidates in $\triangleright[1, \ell'] \cup \{q\}$.

Our algorithm proceeds as follows. We define a configuration as an oriented graph whose vertices are a subset of candidates. We enumerate all possible configurations, with our focused candidates being the vertices, and check if at least one of the enumerated configurations leads to the occurrence of at least one of the above two cases. Precisely, the enumerated configurations consist of the following oriented graphs:

- all oriented graphs whose vertex set is $\triangleright[1, \ell']$; and
- for each $q \in C \setminus \triangleright[1, \ell']$, all oriented graphs whose vertex set is $\triangleright[1, \ell'] \cup \{q\}$.

It is clear that there are FPT-many configurations to enumerate with respect to ℓ' . For each configuration H , we check in polynomial time whether one of the two cases

occurs, based on the definition of the h -amendment procedure. More precisely, we first compute the h -amendment winner of H with respect to the agenda \triangleright restricted to only our focused candidates (corresponding to vertices of H). If H has vertex set $\triangleright[1, \ell']$ and the winner is not p , this configuration results in Case (1). If H has vertex set $\triangleright[1, \ell'] \cup \{q\}$, where $q \in C \setminus \triangleright[1, \ell']$, such that p is the h -amendment winner of $H - \{q\}$ with respect to \triangleright , and there is no arc from p to q in H , then this configuration results in Case (2). If a configuration results in neither Case (1) nor Case (2), we discard it. Otherwise, we utilize the FPT-algorithm in Theorem 4 to check if there are desired $V' \subseteq V$ and $W' \subseteq W$ that achieve the configuration. To be precise, let C' be the vertex set of H . Then, we solve an instance $(C', V|_{C'}, W|_{C'}, H, k, k')$ of MGCEV by Theorem 4, and return the output for the configuration.

If we obtain a Yes-answer from at least one of the enumerated configurations, we conclude that I is a Yes-instance; otherwise we conclude that I is a No-instance. \square

The following result is a consequence of Theorem 5.

Corollary 4 *DCAV- h -Amendment and DCDV- h -Amendment are FPT with respect to the combined parameter of h and the number of predecessors of the distinguished candidate.*

In contrast to these tractability results, we show that if the distinguished candidate moves to the last position of the agenda, both problems become intractable from the parameterized complexity point of view.

Theorem 6 *For every constant h , DCAV- h -Amendment is W[1]-hard with respect to the number of added votes plus the number of registered votes. This holds even if the distinguished candidate is the last one in the agenda.*

Proof We begin with a proof for the amendment procedure (i.e., when $h = 1$), based on the PERFECT CODE problem. Let (G, κ) be an instance of the PERFECT CODE problem, where G is a graph with vertex set U . We denote the vertex set as $U = \{u_1, \dots, u_n\}$. We create an instance of DCAV-AMENDMENT as follows.

For each $u_i \in U$, we create two candidates, denoted by x_i and y_i , respectively. Let $X = \{x_i : i \in [n]\}$, and let $Y = \{y_i : i \in [n]\}$. In addition, we introduce two candidates, p and q , where p is the distinguished candidate. Let $C = X \cup Y \cup \{p, q\}$. The agenda is $\triangleright = (q, \vec{X}, \vec{Y}, p)$. We create the following $3\kappa + 3$ registered votes:

- one vote with the preference $p \ q \ \vec{X} \ \vec{Y}$;
- $\kappa + 2$ votes with the preference $q \ p \ \vec{X} \ \vec{Y}$;
- $\kappa - 2$ votes with the preference $p \ \vec{X} \ q \ \vec{Y}$; and
- $\kappa + 2$ votes with the preference $p \ \vec{X} \ \vec{Y} \ q$.

Let V denote the multiset of the above registered votes. Figure 3 shows the weighted majority graph of (C, V) and the agenda. As p beats every other candidate, p is the amendment winner of (C, V) .

For each $u_i \in U$, let $N_X[u_i]$ (respectively, $N_Y[u_i]$) be the set of candidates from X (respectively, Y) corresponding to the closed neighbors of u_i in G . Precisely, $N_X[u_i] =$

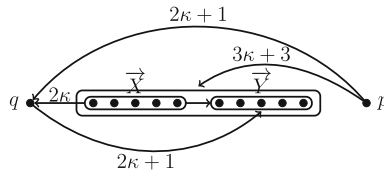


Fig. 3 An illustration of the weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 6. All arcs among the vertices in X (respectively, Y) are forward arcs with a uniform weight of $3\kappa + 3$. The agenda is represented by the left-to-right ordering of the vertices

$\{x_j \in X : u_j \in N_G[u_i]\}$ and $N_Y[u_i] = \{y_j \in Y : u_j \in N_G[u_i]\}$. For each $u_i \in U$, we create one unregistered vote \succ_i with the preference

$$\vec{X}[N_X[u_i]] \left(\vec{Y} \setminus N_Y[u_i] \right) q \ p \left(\vec{X} \setminus N_X[u_i] \right) \vec{Y}[N_Y[u_i]].$$

Let $W = \{\succ_i : u_i \in U\}$.

Let $k = \kappa$. The instance of DCAV- AMENDMENT is $(C, p, V, W, \triangleright, k)$. It remains to prove the correctness of the reduction.

(\Rightarrow) Suppose that G has a perfect code S of size κ . Let W' be the set of unregistered votes corresponding to S . Let $E = (C, V \cup W')$. We show below that q wins E by proving that q beats every other candidate with respect to $V \cup W'$. First, as all unregistered votes rank q before p , there are exactly $2\kappa + 2$ votes in E ranking q before p , implying that q beats p in E . Let x_i be a candidate from X . Because S is a perfect code, there is exactly one $u_j \in S$ such that $x_i \in N_X[u_j]$. From the construction of the unregistered votes, we know that there are exactly $\kappa - 1$ votes in W' ranking q before x_i , resulting in $\kappa + 3 + \kappa - 1 = 2\kappa + 2$ votes ranking q before x_i in the election E . Similarly, we can show that for every $y_i \in Y$, exactly $1 + (\kappa + 2) + (\kappa - 2) + 1 = 2\kappa + 2$ votes ranking q before y_i in E . from $|V \cup W'| = 4\kappa + 3$, it follows that q beats all candidates from $X \cup Y$ in E .

(\Leftarrow) Suppose that $W' \subseteq W$ is a subset of k unregistered votes such that p is not the amendment winner of $E = (C, V \cup W')$ with respect to \triangleright . Observe that no matter which at most κ votes are contained in W' , p beats all candidates in $X \cup Y$. This implies that q is the only candidate which is capable of preventing p from winning. In other words, q is the amendment winner of E with respect to \triangleright . It follows that q beats all the other candidates in E . Then, observe that $|W'| = k$, since otherwise p is not beaten by q in E . Let S be the set of vertices corresponding to W' . Similar to the analysis for the other direction, as for every $y_i \in Y$ we have $2\kappa + 1$ votes in V ranking q before y_i , there is at least one vote in W' ranking q before y_i . By the construction of the unregistered votes, we know that there is at least one $u_j \in S$ such that $y_i \in N_Y[u_j]$, and hence $u_i \in N_G[u_j]$. As y_i can be any candidate in Y , this implies that S is a dominating set of G . Now, we show that for every two vertices $u_i, u_{i'} \in S$, it holds that $N_G[u_i] \cap N_G[u_{i'}] = \emptyset$. Assume, for the sake of contradiction, that there are $u_i, u_{i'} \in S$ and a vertex $u_{j'}$ such that $u_{j'} \in N_G[u_i] \cap N_G[u_{i'}]$. Then, as the two votes \succ_i and $\succ_{i'}$ corresponding to u_i and $u_{i'}$ both rank $x_{j'}$ before q , there can be at most $\kappa - 2$ votes in W' ranking q before $x_{j'}$, resulting in at most $\kappa + 3 + (\kappa - 2) = 2\kappa + 1$

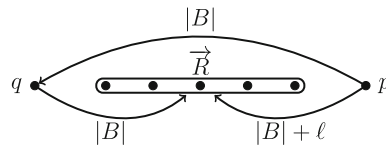


Fig. 4 An illustration of the weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 7. All arcs among the vertices in R are forward with a weight of at least $2|B| - \kappa - \ell + 1$. The agenda is represented by the left-to-right ordering of the vertices

votes in $V \cup W'$ ranking q before $x_{j'}$, contradicting that q beats $x_{j'}$ in E . We can conclude now that S is a perfect code of G .

To prove for the case where $h > 1$, we replace each x_i and each y_i with an ordered block of size h in the above reduction. \square

For control by deleting voters, we have similar results as encapsulated in the following two theorems.

Theorem 7 *For every constant h , DCDV- h -Amendment is $W[2]$ -hard with respect to the number of deleted votes, even if the distinguished candidate is the last one in the agenda.*

Proof We consider first the amendment procedure. We give a reduction from RBDS to DCDV-Amendment. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . Similar to the proof of Theorem 2, we assume that each red vertex in G has the same degree $\ell > 0$, $|B| > \kappa > 1$, and $\ell + \kappa < |B|$. We create an instance of DCDV-Amendment as follows. The candidates and the agenda are the same as in the proof of Theorem 2, i.e., $C = R \cup \{p, q\}$, where p serves as the distinguished candidate, and $\triangleright = (q, \vec{R}, p)$. We create the following $2|B| + 1 - \kappa$ votes in V :

- $|B| + 1 - \ell - \kappa$ votes with the same preference $\vec{R} \ q \ p$;
- ℓ votes with the same preference $q \ p \ \vec{R}$; and
- for each blue vertex $b \in B$, one vote \triangleright_b with the preference $p \ (\vec{R} [N_G(b)]) \ q \ (\vec{R} \setminus N_G(b))$.

Figure 4 illustrates the weighted majority graph of (C, V) and the agenda.

To complete the reduction, we let $k = \kappa$. The instance of DCDV-Amendment is $(C, p, V, \triangleright, k)$. It is easy to check that p beats everyone else, and hence is the amendment winner of (C, V) with respect to \triangleright . We show the correctness of the reduction as follows.

(\Rightarrow) Suppose that there is a subset $B' \subseteq B$ of cardinality κ which dominates R in G . Let $V' = \{\triangleright_b : b \in B'\}$ be the set of votes corresponding to B' . Let $E = (C, V \setminus V')$. Clearly, E contains exactly $2|B| + 1 - 2\kappa$ votes. We show below that q is the amendment winner of E with respect to \triangleright by proving that q beats all the other candidates. First, as all votes in V' rank p before q , the number of votes in $V \setminus V'$ ranking q before p remains as $|B| + 1 - \kappa$, implying that q beats p in E . We consider candidates in R now. Let r be any arbitrary candidate from R . As B' dominates R , there is at least one

vertex $b \in B'$ which is adjacent to r in G . Then, the vote \succ_b corresponding to b ranks r before q , meaning that there can be at most $\kappa - 1$ votes in V' ranking q before r . This further implies that in $V \setminus V'$, at least $\ell + (|B| - \ell - (\kappa - 1)) = |B| + 1 - \kappa$ votes rank q before r , and hence q also beats r in E . As this holds for all $r \in R$, we know that q beats all candidates from R in E .

(\Leftarrow) Assume that there exists a subset $V' \subseteq V$ such that $|V'| \leq k$ and p is not the amendment winner of the election $E = (C, V \setminus V')$ with respect to \triangleright . First, observe that regardless of which at most k votes are included in V' , the candidate p beats all candidates from R in E . This implies that the amendment winner of E with respect to \triangleright must be q . In other words, q beats all other candidates in E . Furthermore, note that V' must consist of exactly k votes, all of which correspond to vertices in B . Otherwise, p remains the winner. Thus, we have $|V \setminus V'| = 2|B| + 1 - 2\kappa$. Now, for every $r \in R$, since there are precisely $\ell + (|B| - \ell) = |B|$ votes in V ranking q before r , at most $\kappa - 1$ votes from V' can rank q before r . This ensures that for every $r \in R$, there exists at least one vote $\succ_b \in V'$ where r is ranked before q . By the definition of \succ_b , it follows that b dominates r in G . Therefore, the subset of blue vertices corresponding to V' dominates R in G .

To establish hardness for every constant $h > 1$, we replace each candidate corresponding to a red vertex with an ordered block of size h in the above-constructed election. The correctness analysis remains similar. \square

Theorem 8 *For every constant h , DCDV- h -Amendment is W[2]-hard with respect to the number of remaining votes, even if the distinguished candidate is the last one in the agenda.*

Proof We prove the theorem by a reduction from RBDS. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . Without loss of generality, we assume that $\kappa < |B|$. Let $C = R \cup \{q, p\}$, with p as the distinguished candidate, and let $\triangleright = (q, \vec{R}, p)$. We create the following votes:

- a multiset V_1 of κ votes with the preference $q \ p \ \vec{R}$;
- a singleton V_2 of one vote with the preference $\vec{R} \ q \ p$; and
- for each blue vertex $b \in B$, one vote \succ_b with the preference

$$p \ (\vec{R} \setminus N_G(b)) \ q \ (\vec{R}[N_G(b)]).$$

For a given $B' \subseteq B$, we use $V_{B'} = \{\succ_b : b \in B'\}$ to denote the multiset of votes created for the blue vertices in B' . Let $V = V_1 \cup V_2 \cup V_B$. It holds that $|V| = |B| + \kappa + 1$. Let $k = |B| - \kappa$. The instance of DCDV- h -Amendment is $(C, p, V, \triangleright, k)$. Obviously, under the assumption that $\kappa < |B|$, the h -amendment winner of (C, V) with respect to \triangleright is p . It remains to show the correctness of the reduction.

(\Rightarrow) Assume that there is a subset $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . Let $V' = V_1 \cup V_2 \cup V_{B'}$, and let $E = (C, V')$. Obviously, $|V'| = 2\kappa + 1$. We claim that q is the h -amendment winner of E with respect to the agenda \triangleright . As q is the first candidate in \triangleright , we need to show that q beats all the other candidates in E . It is clear that q beats p in E . For each red vertex $r \in R$, there exists at least one

$b \in B'$ dominating r in G . From the construction of the votes, we know that q is ranked before r in the vote \succ_b . Therefore, there are in total at least $|V_1| + 1 = \kappa + 1$ votes ranking q before r in V' , implying that q beats r . As this holds for all $r \in R$, the correctness for this direction follows.

(\Leftarrow) Assume that there is a subset $V' \subseteq V$ of at least $2\kappa + 1$ votes such that p is not the h -amendment winner of (C, V') with respect to \triangleright . Observe that as $|V_1| + |V_2| = \kappa + 1$ and all votes in V_B rank p in the first place, it must be that $(V_1 \cup V_2) \subseteq V'$, and V' contains exactly κ votes from V_B (since otherwise p remains as the h -amendment winner of (C, V')). Let $V_{B'} = V' \cap V_B$, where $B' \subseteq B$. As just discussed, it holds that $|V_{B'}| = \kappa$. We also observe that p beats all candidates in R in (C, V') no matter which κ votes from V_B are contained in V' . Therefore, it must be that q is the h -amendment winner of (C, V') , i.e., q beats all the other candidates. We claim that B' dominates R . Suppose, for the sake of contradiction, that this is not the case. Then, there exists at least one red vertex $r \in R$ not dominated by any vertex from B' in G . Then, from the construction of the votes, r is ranked before q in all votes of $V_{B'}$. Together with the vote in V_2 , there are $\kappa + 1$ votes in V' ranking r before q , implying that r beats q in (C, V') . However, this contradicts that q is the h -amendment winner of (C, V') with respect to \triangleright . \square

We now investigate election control through the addition or deletion of candidates. Unlike voter control operations, we demonstrate that candidate control operations generally remain solvable in polynomial time, regardless of the position of the distinguished candidate in the agenda. These results extend the findings for the amendment procedure stated in Corollary 3.

We first consider CCAC-Amendment. We need the following notions for our study.

Let $E = (C, V)$ be an election, and let $\triangleright = (c_1, c_2, \dots, c_m)$ be an agenda on C . For a subset $C' \subseteq C$ and two candidates $c, c' \in C'$ such that $c \triangleright c'$, a $(c \leftarrow c')$ -beating path with respect to (C', V) and \triangleright is a sequence of candidates $(c_{\pi(0)}, c_{\pi(1)}, \dots, c_{\pi(t)})$, where $\{\pi(0), \dots, \pi(t)\} \subseteq [m]$, satisfying the following conditions:

- $c_{\pi(i)} \in C'$ for all $i \in [t] \cup \{0\}$, i.e., all candidates in the $(c \leftarrow c')$ -beating path belong to C' ;
- $c_{\pi(0)} = c'$ and $c_{\pi(t)} = c$;
- $\pi(i-1) > \pi(i)$ for all $i \in [t]$, i.e., $c_{\pi(t)} \triangleright c_{\pi(t-1)} \triangleright \dots \triangleright c_{\pi(0)}$;
- for all $i \in [t]$, $c_{\pi(i)}$ is either beaten by or ties with $c_{\pi(i-1)}$; and
- for all $i \in [t]$, $c_{\pi(i)}$ beats all of its successors from C' that lie between $c_{\pi(i)}$ and $c_{\pi(i-1)}$ in the agenda \triangleright , i.e., $c_{\pi(i)}$ beats $\triangleright[j]$ for all $\triangleright[j] \in C'$ such that $\pi(i) < j < \pi(i-1)$.

In particular, for any candidate $c \in C$, we define the trivial $(c \leftarrow c)$ -beating path as the sequence consisting solely of (c) . We refer to Fig. 5 for an illustration.

Checking whether there is a $(c \leftarrow c')$ -beating path with respect to (C', V) and \triangleright can be clearly done in polynomial time. Moreover, a candidate c is the amendment winner of (C, V) with respect to \triangleright if and only if c beats all her successors, and there is a $(\triangleright[1] \leftarrow c)$ -beating path with respect to (C, V) and \triangleright .

We are now prepared to present our algorithm for CCAC-Amendment.

Theorem 9 *CCAC-Amendment is polynomial-time solvable.*

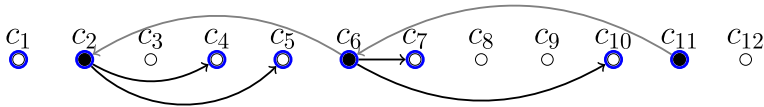


Fig. 5 The sequence (c_{11}, c_6, c_2) (highlighted as dark vertices) represents a $(c_2 \leftarrow c_{11})$ -beating path with respect to the subset $C' = \{c_1, c_2, c_4, c_5, c_6, c_7, c_{10}, c_{11}\}$ (vertices enclosed in blue circles), the election (C', V) , and the agenda $\triangleright = (c_1, c_2, \dots, c_{12})$. A dark arc (a, b) signifies that a beats b , while a gray arc (a, b) indicates that a either beats or ties with b

Proof We derive a dynamic programming algorithm for the CCAC-Amendment problem as follows. Let $I = (C, p, D, V, \triangleright, k)$ be an instance of CCAC-Amendment, where C and D are respectively the set of registered candidates and the set of unregistered candidates, $p \in C$ is the distinguished candidate, V is a multiset of votes, \triangleright is an agenda, and k is the number of unregistered candidates that we can add. Let c^* be the leftmost candidate from C in the agenda \triangleright . That is, $c^* \triangleright c$ holds for all $c \in C \setminus \{c^*\}$. If p does not beat all her successors in C , we directly conclude that I is a No-instance. So, let us assume that p beats all her successors from C in \triangleright . If p is already the amendment winner, I is a Yes-instance; we are done. Otherwise, we determine whether we can add at most k candidates from D into C to make p the amendment winner. Let $D_1 \subseteq D$ be the set of predecessors of p contained in D . Observe that it does not make any sense to add any candidates from $D \setminus D_1$.

We maintain a table $H(d)$ indexed by candidates $d \in D_1$. We define $H(d)$ as the minimum number of successors of d from D_1 that need to be added, under the condition that d is added into C , such that a $(d \leftarrow p)$ -beating path exists with respect to the resulting election and \triangleright . If such candidates do not exist, we define $H(d) = +\infty$. We process the candidates in D_1 sequentially, following the decreasing order of their positions in the agenda \triangleright (i.e., from the rightmost to the leftmost). In particular, the table is computed as follows.

First, let $d \in D_1$ be the rightmost predecessor of p from D_1 . We let $H(d) = 1$ if there is a $(d \leftarrow p)$ -beating path with respect to $(C \cup \{d\}, V)$ and \triangleright . Otherwise, we let $H(d) = +\infty$.

Next, we recursively compute other entries. To this end, let $d \in D_1$ denote the currently considered candidate. Assuming that $H(d')$ has been computed for all $d' \in D_1$ such that $d \triangleright d'$, we compute $H(d)$ as follows.

- If there is a $(d \leftarrow p)$ -beating path with respect to $(C \cup \{d\}, V)$ and \triangleright , we let $H(d) = 1$.
- Otherwise, we define $H(d) = +\infty$ if no $(d \leftarrow d')$ -beating path exists with respect to $(C \cup \{d, d'\}, V)$ and \triangleright for any successor $d' \in D_1$ of d .
- Otherwise, we let

$$H(d) = 1 + \min_{\substack{d' \in D_1, d \triangleright d' \\ \exists (d \leftarrow d')\text{-beating path w.r.t. } (C \cup \{d, d'\}, V) \text{ and } \triangleright}} H(d').$$

After computing all entries of the table, we conclude that the given instance I is a Yes-instance if and only if one of the following two cases occurs:

- there is a candidate $d \in D_1$ after c^* in the agenda \triangleright such that $H(d) \leq k$, and there is a $(c^* \leftarrow d)$ -beating path with respect to $(C \cup \{d\}, V)$ and \triangleright .
- there is a candidate $d \in D_1$ before c^* in the agenda \triangleright such that $H(d) \leq k$.

The algorithm runs in polynomial time since the table has at most $|D_1|$ entries and computing each entry can be done in polynomial time as shown above. \square

For constructive control by deleting candidates, we have the same result.

Theorem 10 *CCDC-Amendment is polynomial-time solvable.*

Proof Let $I = (C, p, V, \triangleright, k)$ be an instance of the CCDC-Amendment problem, where C is a set of candidates, $p \in C$ is a distinguished candidate, V is a multiset of votes on C , \triangleright is an agenda on C , and k is an integer.

A necessary condition for p to be the amendment winner is that p beats all her successors. Therefore, the first step of our algorithm is to remove all the successors of p not beaten by p , and decrease k accordingly.

If $k < 0$ after doing so, we directly conclude that I is a No-instance. Otherwise, if p has at most k predecessors, we immediately conclude that I is a Yes-instance, because in this case, we can make p the winner by deleting all its predecessors without exceeding the budget k .

We assume now that p has at least $k + 1$ predecessors. Let (c_1, c_2, \dots, c_t) be the agenda \triangleright restricted to the predecessors of p . It holds that $t \geq k + 1$. For notational brevity, let $c_{t+1} = p$. For $i, j \in [t]$ such that $i < j$, let $C[i, j]$ denote the union of $\{c_i, c_j\}$ and the set of candidates between c_i and c_j in \triangleright . Additionally, for $i = j$, we let $C[i, j] = \{c_i\}$, and for $i > j$, we define $C[i, j] = \emptyset$. We maintain a table $H(c_i)$ for $i \in [t + 1]$. Specifically, $H(c_i)$ is defined as the minimum number of candidates that need to be removed from $C[i + 1, t]$ such that there is a $(c_i \leftarrow p)$ -beating path in the remaining election with respect to \triangleright .

We compute the entries in the order of $H(c_{t+1}), H(c_t), \dots, H(c_1)$. First, we set $H(c_{t+1}) = 0$. For each c_i where $i \in [t]$, if c_i is not beaten by any candidate from $C[i + 1, t + 1]$, we let $H(c_i) = +\infty$; otherwise, we compute

$$H(c_i) = \min_{\substack{i < j \leq t + 1, \\ c_j \text{ beats or ties with } c_i}} (H(c_j) + j - i - 1) (H(c_j) + j - i - 1).$$

Here, $j - i - 1$ accounts for that all candidates in $C[i + 1, j - 1]$ are considered to be deleted.

The given instance I is a Yes-instance if and only if there exists a candidate c_i with $i \in [t]$ such that $H(c_i) \leq k - i + 1$. The existence of such a candidate c_i implies that we can delete all the $i - 1$ predecessors of c_i , along with at most $k - i + 1$ candidates between c_i and p in \triangleright , to ensure the presence of a $(c_i \leftarrow p)$ -beating path in the remaining election. Since p beats all her successors, this guarantees that p will become the winner after the deletion of the aforementioned candidates. Moreover, note that p has at least $k + 1$ predecessors, which ensures that at least one candidate c_j with $j \in [t]$ is not deleted.

Since the table contains polynomially many entries, and each entry can be computed in polynomial time, the overall algorithm runs in polynomial time. \square

As DCAC-Amendment and DCDC-Amendment are respectively polynomial-time Turing reducible to CCAC-Amendment and CCDC-Amendment, we obtain the following corollary as a consequence of Theorem 9 and Theorem 10.

Corollary 5 *DCAC-Amendment and DCDC-Amendment are polynomial-time solvable.*

3.2 The $(m - h)$ -Amendment Procedure for Constant h

This section focuses on election control for the $(m - h)$ -amendment procedures, where m denotes the number of candidates and h is a positive integer constant. We stipulate that for a fixed constant h , when the $(m - h)$ -amendment procedure is applied to an election with $m \leq h$ candidates and an agenda, the winner is the first candidate in the agenda.

Our first result is stated as follows.

Theorem 11 *For every constant h , CCAV- $(m - h)$ -Amendment is W[1]-hard with respect to the combined parameter of the number of added votes and the number of registered votes. This holds even when the distinguished candidate is the last candidate in the agenda.*

Proof We prove the theorem through a reduction from the PERFECT CODE problem. Consider an instance (G, κ) of PERFECT CODE, where G is a graph, and κ is an integer. Let U denote the vertex set of G . We first give a reduction for the case where $h = 1$. Particularly, we construct an instance of CCAV-Full-Amendment as follows.

Let (u_1, u_2, \dots, u_n) be any arbitrary linear order of U . For each u_i , we create two candidates denoted by x_i and y_i . In addition, we create one candidate p . Let $X = \{x_i : i \in [n]\}$, let $Y = \{y_i : i \in [n]\}$, and let $C = X \cup Y \cup \{p\}$. Furthermore, let $\vec{X} = (x_1, x_2, \dots, x_n)$, and let $\vec{Y} = (y_1, y_2, \dots, y_n)$. The agenda is $\triangleright = (\vec{X}, \vec{Y}, p)$. For each u_i where $i \in [n]$, let

$$N_X[u_i] = \{x_j \in X : j \in [n], u_j \in N_G[u_i]\}$$

be the set of candidates in X corresponding to the closed neighborhood of u_i in G . Analogously, let

$$N_Y[u_i] = \{y_j \in Y : j \in [n], u_j \in N_G[u_i]\}.$$

We create the following registered votes:

- $\kappa + 2$ votes with the preference $p \vec{X} \vec{Y}$;
- $\kappa - 2$ votes with the preference $\vec{X} p \vec{Y}$; and
- $\kappa + 2$ votes with the preference $\vec{X} \vec{Y} p$.

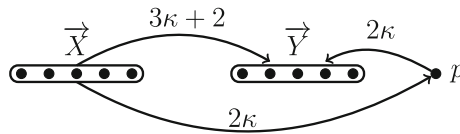


Fig. 6 An illustration of the weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 11. Arcs among the vertices in X (respectively, Y) are forward arcs with a uniform weight of $3\kappa + 2$. The agenda is represented by the left-to-right ordering of the vertices

Let V be the multiset of the above $3\kappa + 2$ registered votes. Figure 6 shows the majority graph of (C, V) .

Now we construct the unregistered votes. For each u_i , $i \in [n]$, we create one unregistered vote \succ_i with the preference

$$\left(\vec{X}[N_X[u_i]]\right) \left(\vec{Y} \setminus N_Y[u_i]\right) p \left(\vec{X} \setminus N_X[u_i]\right) \left(\vec{Y}[N_Y[u_i]]\right).$$

Let W denote the multiset of the above unregistered votes. Let $k = \kappa$, i.e., we are allowed to add at most κ unregistered votes. The CCAV-Full-Amendment instance is $(C, p, V, W, \triangleright, k)$.

In the following, we show that the given PERFECT CODE instance is a Yes-instance if and only if the above constructed CCAV-Full-Amendment instance is a Yes-instance. Observe that as for every $c \in C \setminus \{p\}$, all the $3\kappa + 2$ registered votes rank c before all her successors except p , no matter which at most κ unregistered votes are added, c still beats all her successors in $C \setminus \{p\}$.

(\Rightarrow) Assume that the graph G admits a perfect code $U' \subseteq U$ of size κ . Let $W' = \{\succ_i : u_i \in U'\}$ be the set of unregistered votes corresponding to U' . Let $E = (C, V \cup W')$. As p is the last candidate in the agenda, and every candidate from $C \setminus \{p\}$ beats all her successors from $C \setminus \{p\}$, p wins E if and only if p is not beaten by anyone else in E . Consider two candidates $x_i \in X$ and $y_i \in Y$ where $i \in [n]$. As U' is a perfect code of G , there is exactly one $u_j \in U'$ such that $u_i \in N_G[u_j]$. By the definition of unregistered votes, this implies that in W' , only the vote \succ_j created for u_j ranks x_i before p and, moreover, only the same vote \succ_j ranks p before y_i . It follows that $n_{V \cup W'}(p, x_i) = (\kappa + 2) + (\kappa - 1) = 2\kappa + 1$ and $n_{V \cup W'}(p, y_i) = (\kappa + 2) + (\kappa - 2) + 1 = 2\kappa + 1$. As there are exactly $3\kappa + 2 + \kappa = 4\kappa + 2$ votes in E , we know that p ties with both x_i and y_i . As i can be any integer in $[n]$, we know that p ties with all candidates from $X \cup Y$ in E .

(\Leftarrow) Assume that there is a subset $W' \subseteq W$ of at most κ votes such that p is the full-amendment winner of $E = (C, V \cup W')$ with respect to the agenda \triangleright . First, observe that every candidate from X beats each of its successors in E , except p , regardless of which at most κ votes are contained in W' . This is because all the $3\kappa + 2$ registered votes rank every candidate from X before all of its successors except p .

Moreover, if W' contains at most $\kappa - 1$ votes, then there exists at least one candidate from X who also beats p with respect to $V \cup W'$. This follows because there are already 2κ registered votes that rank every candidate from X before p . If no votes are added, all candidates from X beat p . Alternatively, if any unregistered vote, say

\succ_i , corresponding to a vertex u_i in G is added, any candidate from X , say x_i , that is ranked before p beats p in E .

In other words, if $|W'| \leq \kappa - 1$, there would exist at least one candidate in X who beats all of her successors. In this case, the leftmost such candidate in the agenda would be the winner of E . However, this contradicts the assumption that p wins E .

From the above analysis, we conclude that $|W'| = \kappa$. It follows that $|V \cup W'| = 4\kappa + 2$. Let $U' = \{u_i \in U : \succ_i \in W'\}$ be the set of vertices corresponding to W' . We claim that U' is a perfect code of G . Suppose, for the sake of contradiction, that this is not the case. Then, one of the following two cases occurs.

Case 1: $\exists u_i \in U$ such that $u_i \notin N_G[U']$

In this case, all unregistered votes in W' rank y_i before p . So, in E there are in total $\kappa + 2 + \kappa = 2\kappa + 2$ votes ranking y_i before p , implying that p is beaten by y_i in E , contradicting that p wins E .

Case 2: $\exists u_i \in U$ and $u_j, u_{j'} \in U'$ such that $u_i \in N_G[u_j] \cap N_G[u_{j'}]$.

Due to the construction of the unregistered votes, the two votes \succ_j and $\succ_{j'}$ in W' both rank x_i before p . In this case, there are at least $2\kappa + 2$ votes in $V \cup W'$ ranking x_i before p , implying that x_i beats p in E . However, this contradicts that p wins E .

Since both of the above cases lead to contradictions, U' is a perfect code of G .

To show the $W[1]$ -hardness of CCAV- $(m - h)$ -Amendment we add a set of $h - 1$ additional candidates p_1, p_2, \dots, p_{h-1} in the following way:

- (1) we add them before p with their relative order being $p_{h-1}, p_{h-2}, \dots, p_1$ in the agenda; and
- (2) in each vote, we rank them after p with their relative order being p_1, p_2, \dots, p_{h-1} .

So, in the new election, p beats each p_i where $i \in [h - 1]$, and the head-to-head comparison between every candidate x_i (respectively, y_i) and each p_j where $j \in [h - 1]$ is exactly the same as that between x_i (respectively, y_i) and p . \square

For control by deleting voters, we also obtain a parameterized intractability result.

Theorem 12 *For every constant h , CCDV- $(m - h)$ -Amendment is $W[2]$ -hard with respect to the number of deleted votes. This holds even when the distinguished candidate is the last candidate in the agenda.*

Proof We prove the theorem by reductions from the RBDS problem. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . We first provide a reduction for the case where $h = 1$. Similar to the proof of Theorem 2, we assume that every red vertex has the same degree, which we denote by ℓ . We also assume that $\kappa \geq 3$. The candidate set and the agenda are exactly the same as in the proof of Theorem 2. Precisely, we have that $C = R \cup \{p, q\}$ where p is the distinguished candidate, and $\triangleright = (q, \vec{R}, p)$. We create the following $2|B| + 2\ell - \kappa$ votes in V :

- $|B| - \kappa + 1$ votes with the preference $\vec{R} \ p \ q$;
- ℓ votes with the preference $q \ p \ \vec{R}$;

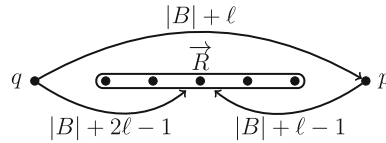


Fig. 7 An illustration of the weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 12. All arcs between the vertices in R are forward arcs with a weight of at least $2|B| - \kappa + \ell$. The agenda is depicted by the left-to-right ordering of the vertices

- $\ell - 1$ votes with the preference $p \succ q \succ \vec{R}$; and
- for each blue vertex $b \in B$, one vote \succ_b with the preference

$$q \left(\vec{R}[N_G(b)] \right) p \left(\vec{R} \setminus N_G(b) \right).$$

Let $k = \kappa$. The constructed instance of CCDV-FULL-AMENDMENT is $(C, p, V, \triangleright, k)$. The weighted majority graph of (C, V) and the agenda \triangleright are illustrated in Fig. 7.

It is easy to see that q beats everyone else and is the full-amendment winner of (C, V) with respect to \triangleright . The construction clearly can be created in polynomial time. In the following, we prove the correctness of the reduction.

(\Rightarrow) Assume that there is a subset $B' \subseteq B$ of cardinality κ such that B' dominates R in G . Let $V' = \{\succ_b : b \in B'\}$ be the set of votes corresponding to B' . Let $E = (C, V \setminus V')$. Clearly, $|V \setminus V'| = 2|B| - 2\kappa + 2\ell$. We shall show that p is not beaten by anyone else in E and hence is the full-amendment winner. As all votes in V' rank q before p , it holds that $n_{V \setminus V'}(p, q) = (|B| - \kappa + 1) + (\ell - 1) = |B| - \kappa + \ell$, meaning that p ties with q in E . Moreover, as B' dominates R , for every $r \in R$, there is at least one blue vertex $b \in B'$ dominating r . By the construction of the votes, we know that r is ranked before p in the vote $\succ_b \in V'$ corresponding to b . It follows that at most $\kappa - 1$ votes in V' rank p before r . By the construction of the votes, we know that there are at least $\ell + (\ell - 1) + (|B| - \ell) - (\kappa - 1) = |B| + \ell - \kappa$ votes ranking p before r in $V \setminus V'$, implying that p ties with r in E . As this holds for all $r \in R$, we know that p ties with all candidates in R . It follows that p is the full-amendment winner of E .

(\Leftarrow) Assume that we can make p the full-amendment winner by deleting at most $k = \kappa$ votes from V , i.e., $\exists V' \subseteq V$ such that $|V'| \leq k$ and p wins $E = (C, V \setminus V')$ with respect to \triangleright . It is easy to see that no matter which at most k votes are contained in V' , q beats all candidates in R , and every $r \in R$ beats all her successors in R in the election E . Therefore, it must be that p beats or ties with all the other candidates in E . This implies that all votes in V' must rank q before p and, moreover, it must be that $|V'| = k = \kappa$, since otherwise p is beaten by q in E . There are two groups of votes ranking q before p : those corresponding to the blue vertices, and those with the preference $q \succ p \succ \vec{R}$. We may assume that all votes in V' belong to the first group. This assumption holds because if V' contained any vote with the preference $q \succ p \succ \vec{R}$, we could construct another solution V'' by replacing this vote with any vote from the first group that is not currently in V' . Let r be a candidate from R . As $n_V(r, p) = |B| - \kappa + \ell + 1$ and $|V \setminus V'| = 2|B| - 2\kappa + 2\ell$, we know that there is at least one vote $\succ_b \in V'$ which

ranks r before p . By the reduction, we know that the vertex b corresponding to \succ_b dominates r in G . It is clear now that the subset $B' = \{b \in B : \succ_b \in V'\}$ of blue vertices corresponding to V' dominates R in G .

To prove the $W[2]$ -hardness of $\text{CCDV}-(m-h)$ -Amendment for other constant values of h , we modify the reduction above in the same manner as in the proof of Theorem 11. Specifically, we add $h-1$ additional candidates, denoted as p_1, p_2, \dots, p_{h-1} , in the same way as in the proof of Theorem 11. \square

Theorem 13 *For every constant h , $\text{CCDV}-(m-h)$ -Amendment is $W[2]$ -hard with respect to the number of remaining votes. This result holds even when the distinguished candidate occupies the last position in the agenda.*

Proof We prove the theorem via reductions from the RBDS problem. First, we consider the case where $h = 1$. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . Without loss of generality, assume that $\kappa < |B|$. We construct an instance of CCDV -Full-Amendment as follows.

For each red vertex, we create one candidate denoted still by the same symbol for notational simplicity. In addition, we create two candidates q and p , where p is the distinguished candidate. Let $C = R \cup \{q, p\}$. The agenda is $\triangleright = (q, \vec{R}, p)$. We create the following votes:

- a multiset V_1 of $\kappa - 1$ votes with the preference $p \ q \ \vec{R}$;
- a singleton V_2 of one vote with the preference $\vec{R} \ p \ q$; and
- for each blue vertex $b \in B$, one vote \succ_b with the preference

$$q \ (\vec{R} \setminus N_G(b)) \ p \ (\vec{R}[N_G(b)]).$$

For a given $B' \subseteq B$, let $V_{B'} = \{\succ_b : b \in B'\}$ be the submultiset of votes created for the blue vertices in B' . Let $V = V_1 \cup V_2 \cup V_B$. It holds that $|V| = |B| + \kappa$. Let $k = |B| - \kappa$. Obviously, under the assumption that $\kappa < |B|$, the full-amendment winner of (C, V) with respect to \triangleright is q . The instance of CCDV -FULL-AMENDMENT is $(C, p, V, \triangleright, k)$, which asks if there is a submultiset $V' \subseteq V$ of cardinality at least $|V| - k = 2\kappa$ such that p is the full-amendment winner of (C, V') with respect to \triangleright . It remains to show the correctness of the reduction.

(\Rightarrow) Assume that there is a subset $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . Let $V' = V_1 \cup V_2 \cup V_{B'}$, and let $E = (C, V')$. We claim that p is the full-amendment winner of E with respect to \triangleright . Since p is the last candidate in the agenda, it suffices to show that p is not beaten by any other candidate in E . It is clear that p ties with q in E . Let $r \in R$ be a red vertex. As B' dominates R , there is at least one vote \succ_b in $V_{B'}$ such that b dominates r . From the construction of the votes, we know that p is ranked before r in the vote \succ_b . Therefore, there are in total at least $|V_1| + 1 = \kappa$ votes ranking p before r in V' , implying that p is not beaten by r . As this holds for all $r \in R$, the correctness for this direction follows.

(\Leftarrow) Assume that there exists a submultiset $V' \subseteq V$ of at least 2κ votes such that p is the full-amendment winner of (C, V') with respect to \triangleright . Observe that, since $|V_1| + |V_2| = \kappa$ and all votes in V_B rank q in first place, it must hold that $(V_1 \cup V_2) \subseteq V'$,

and V' contains exactly κ votes from V_B . Otherwise, q would be the full-amendment winner of (C, V') , contradicting that p wins (C, V') . Define $V_{B'} = V' \cap V_B$, where $B' \subseteq B$. As noted, it follows that $|V_{B'}| = \kappa$.

We claim that B' dominates R in G . Suppose, for the sake of contradiction, that there exists a red vertex $r \in R$ not dominated by any vertex from B' . Then, by the construction of the votes, r is ranked before all its successors in all votes of $V_{B'}$. Combined with the vote in V_2 , there are $\kappa + 1$ votes in V' ranking r before all its successors. In this case, the leftmost $r \in R$ not dominated by B' becomes the full-amendment winner of (C, V') , contradicting the winning of p in (C, V') .

Therefore, the claim holds. Since $|B'| = \kappa$, the RBDS instance is a Yes-instance.

To prove the $W[2]$ -hardness of $\text{CCDV-}(m - h)\text{-Amendment}$ for other constant values of h , we modify the above reduction in the same way as done in the proof of Theorem 11. \square

Now we study destructive control by adding/deleting voters. Compared with previous results, the complexity transition of these problems for the $(m - h)$ -amendment procedures is more sharp. Particularly, by Corollary 2, these problems are polynomial-time solvable if the distinguished candidate is the first one in the agenda. However, when the distinguished candidate is the second, the third, or the last candidate in the agenda, we show that the problems become already intractable from the parameterized complexity point of view. This stands in contrast to the fixed-parameter tractability of the same problems for the h -amendment procedures with respect to the combined parameter of h and the number of predecessors of the distinguished candidate (Corollary 4).

Theorem 14 *For every constant h , $\text{DCAV-}(m - h)\text{-Amendment}$ is $W[1]$ -hard with respect to the number of added votes plus the number of registered votes as long as the distinguished candidate is not the first one in the agenda.*

Proof We first show that the reduction in the proof of Theorem 6 directly applies here (with only the agenda being different). Recall that the reduction was from the PERFECT CODE problem. We had the candidate set $C = X \cup Y \cup \{p, q\} \cup \{p_1, p_2, \dots, p_{h-1}\}$, where $X = \{x_1, \dots, x_m\}$ and $Y = \{y_1, \dots, y_m\}$ with x_i and y_i each being the two candidates created for a vertex u_i in the given graph G of the PERFECT CODE instance, and $(p, p_1, p_2, \dots, p_{h-1})$ is an ordered block. We let \triangleright be an agenda such that q is the first candidate, and, excluding p , the last $h - 1$ candidates in the agenda are p_1, p_2, \dots, p_{h-1} . (Other candidates except p can be ordered arbitrarily between q and p_1 , and p can be in any position after q in the agenda.) We created a multiset V of votes such that p beats all the other candidates and, moreover, except for q , this holds even after we add at most k unregistered votes. This means that q is the only candidate which is able to replace p as the new winner. By the definition of the $(m - h)$ -amendment procedure, q is the $(m - h)$ -amendment winner if and only if q beats all the other candidates. In the proof of Theorem 6, we constructed a multiset of unregistered votes such that there exists a perfect code of G if and only if at most k unregistered votes can be added to ensure that q beats all other candidates. The theorem follows. \square

One might wonder why, in Theorem 6, we require the distinguished candidate p to be the last candidate in the agenda, whereas in Theorem 14, we only stipulate that p

must not be the first candidate in the agenda. The distinction arises because, if p is not the last candidate in the agenda, preventing p from being an h -amendment winner does not necessarily require q to beat all other candidates. For example, suppose p immediately follows q in the agenda. In this case, to prevent p from being an amendment winner, it suffices for q to beat p , even if q is beaten by a candidate appearing after p in the agenda. Our FPT-result for DCAV-Full-Amendment, stated in Corollary 4, also reflects this distinction from a different perspective.

For destructive control by deleting voters, we have a similar result.

Theorem 15 *For every constant h , DCDV- $(m - h)$ -Amendment is $W[2]$ -hard with respect to the number of deleted votes, as long as the distinguished candidate is not the first one of the agenda.*

Proof We adopt the reduction in the proof of Theorem 7 to prove this theorem. Recall that the reduction was from RBDS. Specifically, given an instance $I = (G, \kappa)$ of RBDS, where G is a bipartite graph with the bipartition (R, B) , we created a set $C = R \cup \{q\} \cup \{p, p_1, p_2, \dots, p_{h-1}\}$ of candidates, where p is the distinguished candidate. Moreover, we created a multiset of votes such that

- (p, p_1, \dots, p_{h-1}) is an ordered block;
- p beats all the other candidates and, moreover, except for q , this holds even if we delete at most $k = \kappa$ votes; and
- I is a Yes-instance if and only if we can delete at most k votes such that q beats all other candidates.

It is fairly easy to verify that the reduction applies to DCDV- $(m - h)$ -Amendment for all agendas where q is the first candidate, and p_1, p_2, \dots, p_h are the last h candidates in the agenda without p . \square

Theorem 16 *For every constant h , DCDV- $(m - h)$ -Amendment is $W[2]$ -hard with respect to the number of remaining votes, as long as the distinguished candidate is not the first one of the agenda.*

Proof We prove the theorem via a reduction from the RBDS problem. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . Without loss of generality, we assume $|B| \geq \kappa + 2$. The candidate set is defined as $C = R \cup \{q, p\}$, where p is the distinguished candidate. The agenda \triangleright can be any ordering in which q occupies the first position. We construct the following votes:

- a multiset V_1 of κ votes, each with the preference $q \succ p \succ \vec{R}$;
- a singleton V_2 containing one vote with the preference $\vec{R} \succ q \succ p$; and
- for each blue vertex $b \in B$, one vote \succ_b with the preference

$$p \succ (\vec{R} \setminus N_G(b)) \succ q \succ (\vec{R}[N_G(b)]).$$

For each subset $B' \subseteq B$, let $V_{B'} = \{\succ_b : b \in B'\}$ denote the multiset of votes corresponding to the blue vertices in B' . Define $V = V_1 \cup V_2 \cup V_B$. Clearly, $|V| =$

$|B| + \kappa + 1$. Let $k = |B| - \kappa$. Under the assumption $|B| \geq \kappa + 2$, it is evident that the $(m - h)$ -amendment winner of (C, V) with respect to the agenda \triangleright is p .

It remains to show the correctness of the reduction.

(\Rightarrow) Assume there exists a subset $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . Let $V' = V_1 \cup V_2 \cup V_{B'}$, and let $E = (C, V')$. We claim that q is the $(m - h)$ -amendment winner of E with respect to the agenda \triangleright . Since q is in the first position of \triangleright , it suffices to show that q beats every other candidate in E (Lemma 1).

Observe that $|V'| = 2\kappa + 1$. As all the $\kappa + 1$ votes from $V_1 \cup V_2$ prefer q to p , it follows that q beats p in E . Let $r \in R$ be a red vertex. Since B' dominates R in G , there exists at least one vote \succ_b in $V_{B'}$ such that b dominates r , i.e., $r \in N_G(b)$. By the construction of \succ_b , we know that q is ranked before r in this vote. Thus, there are at least $|V_1| + 1 = \kappa + 1$ votes in V' ranking q before r , implying that q beats r . Since this holds for all $r \in R$, the correctness of this direction is established.

(\Leftarrow) Assume there exists a submultiset $V' \subseteq V$ of at least $2\kappa + 1$ votes such that p is not the $(m - h)$ -amendment winner of (C, V') with respect to \triangleright . As $|V_1| + |V_2| = \kappa + 1$ and all votes in V_B rank p in the first place, we know that $(V_1 \cup V_2) \subseteq V'$, and V' must contain exactly κ votes from V_B , because otherwise p remains the $(m - h)$ -amendment winner of (C, V') , a contradiction. Let $V_{B'} = V' \cap V_B$, where $B' \subseteq B$. It follows that $|V_{B'}| = \kappa$.

Observe that p beats all candidates from R in (C, V') , regardless of which κ votes from V_B are included in V' . Therefore, q must be the $(m - h)$ -amendment winner of (C, V') , meaning q beats all other candidates. We claim that B' dominates R .

Suppose, for the sake of contradiction, that there exists a red vertex $r \in R$ not dominated by any vertex in B' . Then, by the construction of the votes, r is ranked before q in all votes from $V_{B'}$. Together with the vote in V_2 , there are $\kappa + 1$ votes in V' ranking r before q , implying that r beats q in (C, V') . This contradicts the assumption that q beats all candidates in (C, V') . Thus, B' must dominate R in G .

As $|B'| = \kappa$, the given instance of the RBDS problem is a Yes-instance. \square

We now turn to candidate control operations. Unlike the h -amendment procedures, we first show that constructive control by adding candidates under the $(m - h)$ -amendment procedures is hard to solve, for each positive integer constant h . Notably, this is our only hardness result for candidate control under the various variants of amendment procedures.

Theorem 17 *For every constant h , CCAC- $(m - h)$ -Amendment is W[2]-hard with respect to the number of added candidates. This holds even when the distinguished candidate is the last one in the agenda.*

Proof We prove the theorem via a reduction from RBDS. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . We first consider the full-amendment procedure, i.e., the case where $h = 1$. We construct an instance of CCAC-Full-Amendment as follows.

For each vertex in G , we create one candidate, denoted by the same symbol for notational simplicity. Additionally, we create a distinguished candidate p . Let $C = R \cup \{p\}$, let $D = B$, let $k = \kappa$, and define the agenda as $\triangleright = (\vec{R}, \vec{B}, p)$. We create a multiset V of votes such that

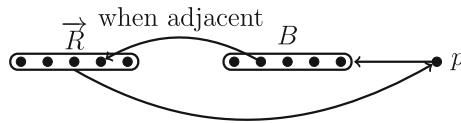


Fig. 8 An illustration of the majority graph of $(C \cup D, V)$ and the agenda used in the proof of Theorem 17. All arcs between the vertices in R are forward. Arcs between the vertices in B are irrelevant as they have no impact on the correctness of the reduction. The agenda is depicted by the left-to-right ordering of the vertices

- every candidate from R beats p ;
- p beats every candidate from B ;
- for each $r \in R$ and $b \in B$, if b dominates r in G , then b beats r ; otherwise, r beats b ; and
- every candidate $r \in R$ is beaten by all of its predecessors in the agenda \triangleright .

We refer to Fig. 8 for an illustration of the majority graph of $(C \cup D, V)$ and the agenda \triangleright .

By McGarvey’s theorem [41], such votes can be constructed in polynomial time. The corresponding CCAC-Full-Amendment instance is given by $(C, p, D, V, \triangleright, k)$.

The correctness of the reduction is easy to see. In particular, if there is a subset $B' \subseteq B$ of κ vertices which dominate R in G , then after adding the candidates corresponding to B' , for every $r \in R$, there is at least one candidate from B' beating r , excluding the winning of r . Candidates from B' cannot win as they are beaten by p . Therefore, after adding these candidates, p becomes the winner. If, however, the RBDS instance is a No-instance, then no matter which at most κ candidates from B are added, there is at least one candidate in R who beats all her successors in the agenda. As all candidates from R are before p in the agenda \triangleright , in this case, we cannot add at most κ candidates to make p the winner.

The above reduction can be modified in the following way to prove the hardness for every $h > 1$: replace the distinguished candidate by an ordered block of h candidates with the first one in the block being the distinguished candidate. \square

However, when the distinguished candidate has only a few predecessors, the problem becomes tractable from the perspective of parameterized complexity.

Theorem 18 *For every constant h , CCAC- $(m - h)$ -Amendment is FPT with respect to the number of predecessors of the distinguished candidate.*

Proof Let $I = (C, p, D, V, \triangleright, k)$ be an instance of CCAC- $(m - h)$ -Amendment, where $p \in C$ is the distinguished candidate. Let C_1 (respectively, D_1) be the set of predecessors of p contained in C (respectively, D), and let $C_2 = C \setminus (C_1 \cup \{p\})$ (respectively, $D_2 = D \setminus D_1$). Additionally, let $D'_2 \subseteq D_2$ be the set of candidates from D_2 beaten by p with respect to V .

We first give an FPT-algorithm for the full-amendment procedure, and then we show how to extend the algorithm for the $(m - h)$ -amendment procedures. Let $\ell = |C_1| + |D_1|$ be the number of predecessors of p in $C \cup D$.

If at least one candidate from C_2 is not beaten by p with respect to V , we immediately conclude that I is a No-instance. So, let us assume that p beats all candidates from C_2 .

We split I into at most 2^ℓ subinstances by enumerating all subsets $S \subseteq D_1$. In particular, each subinstance takes I and a subset $S \subseteq D_1$ as input, and determines if there is a subset $D' \subseteq D_2$ such that $|D'| \leq k - |S|$ and p is the full-amendment winner of $(C \cup S \cup D', V)$ with respect to \triangleright . Clearly, I is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. To solve a subinstance associated with an enumerated subset S , we create an instance of RBDS as follows.

- Red vertices are those in $C_1 \cup S$ who beat their respective successors from $C \cup S$.
- Blue vertices are those in D'_2 .
- We create an edge between a red vertex c and a blue vertex c' if and only if c' is not beaten by c with respect to V .
- Let $\kappa = \min\{k - |S|, |D'_2|\}$.

It is straightforward to verify that the RBDS instance is a Yes-instance if and only if the subinstance is a Yes-instance. Regarding the running time, we reiterate that RBDS is FPT with respect to the number of red vertices (see, e.g., the works of Dom, Lokshtanov, and Saurabh [20], Fomin, Kratsch, and Woeginger [30]). Then, as each RBDS instance constructed above has at most ℓ red vertices, and we have at most 2^ℓ subinstances to consider, the whole algorithm runs in FPT-time with respect to ℓ .

We now consider the $(m - h)$ -amendment procedures, where h is a positive constant integer. We enumerate all tuples (S, S') of two subsets such that

- $S \subseteq D_1$,
- $S' \subseteq D'_2$,
- $|S'| \leq \min\{h - 1, |D'_2|\}$, and
- $|S| + |S'| \leq k$.

Candidates in S are assumed to be exactly those from D_1 that are included in a desired feasible solution, while candidates in S' are also considered part of the solution. Moreover, candidates in S' are assumed to be the last candidates from D'_2 included in the solution. Let B denote the set of candidates from D'_2 that appear before any candidate from S' in \triangleright . Formally, $B \subseteq D'_2$, and for all $b \in B$ and all $c \in S'$, it holds that $b \triangleright c$. Let $\kappa = \min\{k - |S \cup S'|, |B|\}$. For each enumerated $\{S, S'\}$, we determine whether there exists $B' \subseteq B$ such that

- $|B'| \leq \kappa$, and
- p is the $(m - h)$ -amendment winner of $(C \cup S \cup S' \cup B', V)$ with respect to \triangleright .

The original instance I is a Yes-instance if and only if there is a Yes-answer to the above question for at least one enumerated (S, S') .

In the following, we describe how to solve a subproblem associated with an enumerated tuple (S, S') . Let \triangleright' be the agenda \triangleright restricted to $C \cup S \cup S'$. Let A be the set of the last $h - 1$ candidates in \triangleright' . We have two cases to consider.

Case: $p \in A$.

In this case, $B = \emptyset$ holds. Consequently, we conclude that I is a Yes-instance if p is the $(m - h)$ -amendment winner of $(C \cup S \cup S', V)$ with respect to \triangleright' , and discard (S, S') otherwise.

Case: $p \notin A$.

In this case, we construct an instance of the RBDS problem. Let $p = \triangleright'[\ell']$, where $\ell' \leq \ell$. The red vertices in the RBDS instance are from $C_1 \cup S$. Specifically, a candidate $\triangleright'[i]$, where $i \in [\ell' - 1]$, is a red vertex if and only if at least one of the following conditions holds:

- $i \leq h - 1$, and $\triangleright'[i]$ beats all candidates from $\triangleright'[i + 1, i + m' - h]$ with respect to V , where $m' = |C \cup S \cup S'|$;
- $i \geq h$, and $\triangleright'[i]$ beats all of its successors in $C \cup S \cup S'$.

Let R denote the set of the red vertices. Blue vertices are those in B . Edges are constructed as follows: an edge exists between a vertex $\triangleright'[i] \in R$ and a vertex $b \in B$ if and only if b is not beaten by $\triangleright'[i]$ with respect to V . The instance asks if there exists a subset of κ blue vertices that dominate all red vertices. If the resulting RBDS instance is a Yes-instance, we conclude that I is a Yes-instance; otherwise, (S, S') is discarded.

If all enumerated tuples are discarded, we conclude that I is a No-instance.

The algorithm runs in FPT-time with respect to ℓ for the following reasons: we enumerate at most $2^\ell \cdot |D|^{h-1}$ tuples, h is a constant, the RBDS problem is FPT with respect to the number of red vertices, and each RBDS instance generated contains at most ℓ red vertices. \square

For the operation of deleting candidates, we present a natural polynomial-time algorithm for the full-amendment procedure, based on two reduction rules outlined below.

Let $I = (C, p, V, \triangleright, k)$ be an instance of the CCDC-Full-Amendment problem.

Reduction Rule 1 *If p has a successor q in \triangleright that is not beaten by p with respect to V , remove q from C and \triangleright , and decrease k by one.*

Reduction Rule 2 *If the full-amendment winner q of (C, V) with respect to \triangleright precedes p in \triangleright , remove q from C and \triangleright , and decrease k by one.*

Since each application of these reduction rules removes one candidate, the rules can be applied at most $m - 1$ times, where m denotes the number of candidates. As the condition in each rule can be verified in polynomial time, exhaustively applying these reduction rules runs in polynomial time.

A reduction rule is said to be sound for a problem if applying it to any instance of the problem results in an equivalent instance of the same problem. The soundness of Reduction Rule 1 for the CCDC-Full-Amendment problem is evident: if a candidate p can become a winner by deleting other candidates, and there exists a successor q of p in \triangleright not beaten by p , then q must be deleted. We now prove the soundness of Reduction Rule 2.

Lemma 3 *Reduction Rule 2 for the CCDC-Full-Amendment problem is sound.*

Proof Let $I' = (C', p, V', \triangleright', k-1)$ be the instance obtained after applying Reduction Rule 2 to $I = (C, p, V, \triangleright, k)$ and q , where

- (1) q is the full-amendment winner of (C, V) with respect to \triangleright ,
- (2) $q \triangleright p$,

- (3) $C' = C \setminus \{q\}$,
- (4) V' is obtained from V by removing q from every vote in V ,
- (5) \triangleright' is the agenda \triangleright restricted to C' .

We prove below that I' is a Yes-instance of the CCDC-Full-Amendment problem if and only if I is a Yes-instance of the same problem.

(\Rightarrow) Assume there exists $D \subseteq C'$ such that $|D| \leq k-1$ and p is the full-amendment winner of $(C' \setminus D, V)$ with respect to \triangleright' . It follows that $D \cup \{q\}$ is a witness for I . Thus, I is a Yes-instance.

(\Leftarrow) Now assume that I is a Yes-instance, i.e., there exists $D \subseteq C$ such that $|D| \leq k$ and p is the full-amendment winner of $(C \setminus D, V)$ with respect to \triangleright .

If $q \in D$, then $D \setminus \{q\}$ is witness for I' .

Now consider the case where $q \notin D$. By Condition (1) above, q beats all its successors in \triangleright . According to the definition of the full-amendment procedure, if q is not eliminated during the winner determination process, q will remain the winner of $(C \setminus D, V)$, which contradicts p being the winner of $(C \setminus D, V)$. Thus, q must be eliminated when some candidate $a \in C \setminus D$ with $a \triangleright q$ is considered during the winner determination procedure. By Condition (2) above, it follows that $a \triangleright p$. Note that at each round of the full-amendment procedure, either the considered candidate is eliminated, or all its successors are eliminated (in which case the considered candidate is declared the winner). Therefore, if q is eliminated when a is considered, p must also be eliminated, which contradicts p 's status as the winner.

We can now conclude that the case where $q \notin D$ is impossible. \square

With the two reduction rules at hand, we are ready to present the following result.

Theorem 19 *CCDC-Full-Amendment is polynomial-time solvable.*

Proof Let $I = (C, p, V, \triangleright, k)$ be an instance of the CCDC-Full-Amendment problem. We exhaustively apply Reduction Rule 1 and Reduction Rule 2. The order of application does not matter, as the soundness of each rule is independent of the other. As previously discussed, the exhaustive application of these rules can be performed in polynomial time.

After the reduction rules have been applied exhaustively, candidate p becomes the full-amendment winner. We conclude that I is a Yes-instance if and only if $k \geq 0$. \square

Since DCDC-Full-Amendment is polynomial-time Turing reducible to CCDC-Full-Amendment, we obtain the following corollary from Theorem 19.

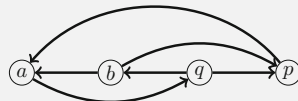
Corollary 6 *DCDC-Full-Amendment is polynomial-time solvable.*

It is important to note that the soundness of Reduction Rule 2 fails for the CCDC- $(m-h)$ -Amendment problem, even when $h = 2$, as demonstrated in Example 2.

Example 2: Reduction Rule 2 is not sound for CCDC- $(m - 2)$ -Amendment

Consider an instance $(C, p, V, \triangleright, k)$, where the election (C, V) and the majority graph of (C, V) along with the agenda \triangleright are depicted below. The election is illustrated on the left, while the majority graph appears on the right, with the agenda represented by a left-to-right ordering of the vertices. Let $k = 1$. The $(m - 2)$ -amendment winner of (C, V) is q . Consequently, if Reduction Rule 2 is applied, q will be removed, and k will be reduced to zero. However, it is straightforward to verify that p is not the $(m - 2)$ -amendment winner of the remaining election. (Note that the number of candidates in the remaining election is three, not four. In this case, the $(m - 2)$ -amendment procedure becomes equivalent to the amendment procedure.) Nevertheless, the instance is a Yes-instance; specifically, the unique feasible solution is to delete candidate b .

vote 1: $a \ q \ b \ p$
 vote 2: $q \ b \ p \ a$
 vote 3: $p \ b \ a \ q$



Let us now consider the problem of destructive control by adding candidates. We first show that, for the full-amendment procedure, it suffices to consider adding at most one candidate. This result is formalized in the following lemma.

Lemma 4 *Let C and D be two disjoint sets of candidates, V a multiset of votes over $C \cup D$, \triangleright an agenda on $C \cup D$, and $p \in C$ a candidate. If there exists $D' \subseteq D$ such that p is not the full-amendment winner of $(C \cup D', V)$ with respect to \triangleright , then there exists a subset $D'' \subseteq D'$ such that $|D''| \leq 1$ and p is not the full-amendment winner of $(C \cup D'', V)$ with respect to \triangleright .*

Proof Assume that there exists a subset $D' \subseteq D$ such that p is not the full-amendment winner of $(C \cup D', V)$ with respect to \triangleright .

If p has a successor $d \in C \cup D'$ that is unbeaten by p with respect to V , we are done. In this case, defining $D'' = D' \cap \{d\}$, p cannot be the full-amendment winner of $(C \cup D'', V)$ due to the presence of d .

Otherwise, p is eliminated when some candidate $c \in C \cup D'$, preceding p in the agenda \triangleright , is considered during the winner determination process applied to $(C \cup D', V)$. In this scenario, c is declared the winner of $(C \cup D', V)$. Let $D'' = D' \cap \{c\}$. Clearly, $|D''| \leq 1$. Now, consider the election $E = (C \cup D'', V)$. If c is eliminated during the winner determination process when one of c 's predecessors in E is considered, then, by the definition of the full-amendment procedure, p is also eliminated along with c , thereby preventing p from winning. Otherwise, as c wins $(C \cup D', V)$ and $D'' \subseteq D'$, the candidate c beats all of its successors in E . In this case, c also wins $(C \cup D'', V)$. \square

Lemma 4 implies that DCAC-Full-Amendment is solvable in polynomial time. We generalize this result to all $(m - h)$ -amendment procedures, where h is a constant, as stated in the following theorem.

To facilitate a better understanding of the algorithm presented below, let us first establish some convenient terminology. Consider an election $E = (C, V)$ and an agenda \triangleright . Recall that candidates are considered in the order specified by \triangleright . Assume that some candidate $c \in C$ is currently being considered. If c beats all of its next $m - h$ successors with respect to V , we say that c eliminates each of these $m - h$ candidates (or equivalently, that each of these candidates is eliminated by c). Otherwise, there exists a candidate c' among these $m - h$ successors who is unbeaten by c . In this case, we say that c is eliminated by c' .

Note that a candidate c may be considered multiple times. For example, if c eliminates its next $m - h$ successors, it will remain the next candidate to be considered. We say that c is directly eliminated if it is eliminated by one of its successors during its first consideration. Nevertheless, when $m \geq 2h - 1$, each candidate can be considered at most twice. Specifically, if C' denotes the set of the first $h - 1$ candidates in \triangleright , then every candidate in C' can be considered at most twice, while every other candidate can be considered at most once. However, this is not the case when $m < 2h - 1$. For instance, consider $m = 5$ and $h = 4$. In this scenario, $m - h = 1$, and the first candidate in the agenda can be considered up to four times.

Theorem 20 *For every constant h , DCAC- $(m - h)$ -Amendment is polynomial-time solvable.*

Proof Let $I = (C, p, D, V, \triangleright, k)$ be an instance of the DCAC- $(m - h)$ -Amendment problem. Our algorithm first addresses the case where I admits a constant-sized feasible solution, followed by the case where no such feasible solution exists.

Step 1: Handling instances with a constant-sized feasible solution.

In this step, we determine whether there exists a subset $D' \subseteq D$ with $|D'| \leq \min\{2h - 1, k\}$ such that p is not the $(m - h)$ -amendment winner of $(C \cup D', V)$ with respect to \triangleright . Since h is a constant, this can be achieved in polynomial time.

If such a subset exists, we conclude that I is a Yes-instance. Otherwise, we consider the following two cases:

- If $|D| \leq 2h - 1$ or $k \leq 2h - 1$, we directly conclude that I is a No-instance.
- If $|D| \geq 2h$ and $k \geq 2h$, the algorithm proceeds to the next step.

Step 2: Handling instances without a constant-sized feasible solution.

At this point, we know that I does not admit a feasible solution $D' \subseteq D$ with at most $2h - 1$ candidates. Our algorithm proceeds by enumerating all tuples (S^L, S^R) of subsets of D such that $S^L \cap S^R = \emptyset$ and $|S^L| = |S^R| = h - 1$. (For clarity and ease of reference, we note that the superscripts L and R denote “left” and “right”, respectively.) The candidates in S^L (respectively, S^R) are intended to represent the first (respectively, last) $h - 1$ candidates among all added candidates in the agenda \triangleright .

Let $k' = k - |S^L| - |S^R| = k - 2h + 2$. As $k \geq 2h$, k' is positive. For each enumerated (S^L, S^R) , we define $S^{L \cup R} = S^L \cup S^R$, and construct a subinstance to determine whether there exists a subset $S \subseteq D \setminus S^{L \cup R}$ such that:

- $|S| \leq k'$,
- p is not the $(m - h)$ -amendment winner of $(C \cup S^{L \cup R} \cup S, V)$ with respect to \triangleright ,

- $S^L \triangleright S \triangleright S^R$, meaning no candidate from S^{LUR} appears between candidates from S in the agenda \triangleright .

Since h is a constant, the number of subinstances is at most polynomial in the size of I . Furthermore, I is a Yes-instance if and only if at least one of the subinstances is a Yes-instance.

Next, we present a polynomial-time algorithm for solving a subinstance specified by an enumerated tuple (S^L, S^R) .

Let d^{LR} be the rightmost candidate from S^L in \triangleright , and let d^{RL} be the leftmost candidate from S^R in \triangleright . To aid our exposition, we define two disjoint subsets X and Y of $D \setminus S^{LUR}$:

- $X = \{d \in D : d^{LR} \triangleright d \triangleright d^{RL}, d \triangleright p\}$, which consists of all predecessors of p from D that lie between d^{LR} and d^{RL} in the agenda \triangleright .
- $Y = \{d \in D : d^{LR} \triangleright d \triangleright d^{RL}, p \triangleright d\}$, which consists of all successors of p from D that lie between d^{LR} and d^{RL} in the agenda \triangleright .

We observe that p beats all its successors in $C \cup S^{LUR} \cup Y$ with respect to V . The reasoning is as follows: if p has a successor $d \in C \cup S^{LUR} \cup Y$ that is not beaten by p , then p is not the $(m - h)$ -amendment winner of $(C \cup \{d\}, V)$. In this case, the instance would already have been resolved in Step 1.

Let ℓ denote the number of successors of p from $C \cup S^{LUR}$ with respect to the agenda \triangleright . We analyze the problem by considering the following two cases:

Case 1: $\ell \geq h - 1$.

In this case, using reasoning similar to the proof of Lemma 4, we can show that if the subinstance is a Yes-instance, there must exist a candidate $d \in X$ such that p does not win $(C \cup S^{LUR} \cup \{d\}, V)$. Since $|S^{LUR} \cup \{d\}| = 2h - 1$, this implies that the instance would already have been resolved in Step 1.

Case 2: $\ell < h - 1$.

Let $k'' = \min\{h - \ell, k'\}$. In this case, we determine whether there exists a subset $Z \subseteq (X \cup Y)$ such that $|Z| \leq k''$ and p is not the $(m - h)$ -amendment winner of $(C \cup S^{LUR} \cup Z, V)$ with respect to \triangleright . Since h is a constant, this can be done in polynomial time. If such a subset Z exists, we conclude that I is a Yes-instance; otherwise, we discard the enumerated (S^L, S^R) .

Regarding the correctness of this step, the result is evident when $h - \ell > k'$.

For the case where $h - \ell \leq k'$, we claim that if the subinstance is a Yes-instance, then adding a set Z of at most $k'' = h - \ell$ candidates from $X \cup Y$ to the election $(C \cup S^{LUR}, V)$ is sufficient to prevent p from winning. We prove this claim as follows. Assume that there exists a subset $Z' \subseteq (X \cup Y)$ such that p is not the $(m - h)$ -amendment winner of $(C \cup S^{LUR} \cup Z', V)$. Consequently, p is eliminated when one of its predecessors, say c^* , is considered during the winner determination process applied to $(C \cup S^{LUR} \cup Z', V)$. Let \triangleright' denote the agenda \triangleright restricted to $C \cup S^{LUR} \cup Z'$.

We construct the desired set Z as follows: For each candidate $c \in \triangleright'[1, h - 1 - \ell]$ —that is, each candidate within the first $h - 1 - \ell$ positions of \triangleright' —which does not beat at least one of its successors in \triangleright' , we identify the leftmost successor of c in \triangleright' that is not beaten by c with respect to V . Let C' denote the set of these

“leftmost” candidates. We then define

$$Z = (C' \cup \{c^*\}) \cap Z'.$$

Clearly, $|Z| \leq |C'| + |\{c^*\}| \leq (h-1-\ell)+1 = h-\ell$. Let $E = (C \cup S^{\text{LUR}} \cup Z, V)$, and let \triangleright'' be \triangleright restricted to $C \cup S^{\text{LUR}} \cup Z$. Note that due to the presence of candidates in S^{L} , the first $h-1$ candidates in \triangleright' are identical to those in \triangleright'' . Similarly, the presence of candidates in S^{R} ensures that the last $h-1$ candidates in \triangleright' are also the same as those in \triangleright'' . Using similar reasoning as in the proof of Lemma 4, we prove below that p does not win E . Our proof proceeds by considering the following three cases.

Case 1: c^* is within the first $h-1-\ell$ positions of \triangleright' .

In this case, c^* is not eliminated in $(C \cup S^{\text{LUR}} \cup Z', V)$ by any of its predecessors from $\triangleright'[1, h-1-\ell]$. Consider now the winner determination process applied to E and \triangleright'' . By the definition of C' , all predecessors of c^* will be directly eliminated. Since c^* eliminates p in $(C \cup S^{\text{LUR}} \cup Z', V)$ and $Z \subseteq Z'$, it follows that c^* will also eliminate p .

Case 2: c^* is within neither the first $h-1-\ell$ positions nor the last $h-1$ positions of \triangleright' .

As c^* is not eliminated in $(C \cup S^{\text{LUR}} \cup Z', V)$ by any of its predecessors from $\triangleright'[1, h-1-\ell]$ in the election $(C \cup S^{\text{LUR}} \cup Z', V)$, by the definition of C' , c^* is not eliminated by any of its predecessors from $\triangleright''[1, h-1-\ell] = \triangleright'[1, h-1-\ell]$ in E . It follows that all the first $h-1-\ell$ candidates in \triangleright'' are directly eliminated in E . Consequently, if c^* is eliminated when one of its predecessors, say a , is considered in E , then a must not belong to the first $h-1-\ell$ candidates in \triangleright'' . In this case, all the next $(m-h)$ successors of a , including both c^* and p , are eliminated. Otherwise, p will be eliminated by c^* in E .

Case 3: c^* is within the last $(h-1)$ positions of \triangleright' .

Consider the winner determination process applied to E and \triangleright'' . If all the first $h-1-\ell$ candidates in \triangleright'' are directly eliminated, the proof proceeds similarly to that of Case 2. Otherwise, assume that some candidate $c \in \triangleright''[h-1-\ell]$ is not directly eliminated. In this case, after c is considered for the first time, at most h candidates remain. It follows that either p is eliminated together with c^* by some predecessors of c^* , or p is eliminated by c^* .

Finally, if all enumerated tuples are discarded, we conclude that I is a No-instance. \square

The proof of Theorem 20 in fact implies that if an instance of DCAC- $(m-h)$ -Amendment is a Yes-instance, it must have a feasible solution of size at most $\min\{k, 3h-2\}$.

3.3 The Successive Procedure

In this section, we study the successive procedure. We first consider constructive control by adding/deleting voters. A slight modification of the reduction presented in the proof of Theorem 11 leads to the following result.

Theorem 21 *CCAV-Successive is W[1]-hard with respect to the number of added votes plus the number of registered votes. This holds even when the distinguished candidate is the last one in the agenda.*

Proof We show that the reduction established in the proof of Theorem 11 can be used to prove the theorem. Recall that our reduction is from the PERFECT CODE problem, and we have the following $3\kappa + 2$ registered votes in V :

- a multiset V_1 of $\kappa + 2$ votes with the preference $p \vec{X} \vec{Y}$;
- a multiset V_2 of $\kappa - 2$ votes with the preference $\vec{X} p \vec{Y}$; and
- a multiset V_3 of $\kappa + 2$ votes with the preference $\vec{X} \vec{Y} p$.

In addition, for each vertex u_i in G , we have an unregistered vote \succ_i with the preference

$$(\vec{X}[N_X[u_i]])(\vec{Y} \setminus N_Y[u_i]) p (\vec{X} \setminus N_X[u_i])(\vec{Y}[N_Y[u_i]]).$$

The agenda is $\triangleright = (\vec{X}, \vec{Y}, p)$. The proof for the correctness is similar.

(\Rightarrow) In the proof of Theorem 11, we showed that if there is a perfect code, then after adding the unregistered votes corresponding to the perfect code, p ties with everyone else. This means that none of the predecessors of p is the winner after the deletion of the votes. As p is the last one in the agenda, p wins the resulting election.

(\Leftarrow) Suppose that there is a $W' \subseteq W$ such that $|W'| \leq \kappa$ and p is the successive winner of $E = (C, V \cup W')$ with respect to \triangleright . Let $U' = \{u_i \in U : \succ_i \in W'\}$ be the set of vertices corresponding to W' . We claim that U' is a perfect code. Assume, for the sake of contradiction, that this is not the case. Then, one of the following cases occurs.

Case 1: $\exists u_i \in U$ such that $u_i \notin N_G[U']$.

In this case, all unregistered votes in W' rank y_i before all her successors. Thus, in E , there are a total of $|V_3| + \kappa = \kappa + 2 + \kappa = 2\kappa + 2$ votes ranking y_i before the set of all her successors. However, in this case, either some predecessor of y_i or y_i herself wins E , contradicting the assumption that p wins E .

Case 2: $\exists u_i \in U$ and $u_j, u_{j'} \in U'$ such that $u_i \in N_G[u_j] \cap N_G[u_{j'}]$.

By to the construction of the unregistered votes, the two votes \succ_j and $\succ_{j'}$ in W' both rank x_i before all the successors of x_i . So, in E there are at least $|V_2 \cup V_3| + 2 = 2\kappa + 2$ votes ranking x_i before all the successors of x_i . However, in this case, either some predecessor of x_i or x_i herself wins E , contradicting that p wins E .

Since both cases lead to contradictions, we conclude that U' is a perfect code of G .

□

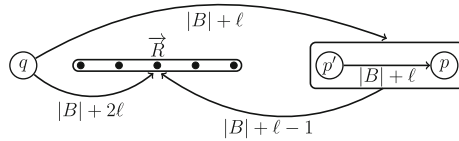


Fig. 9 An illustration of the weighted majority graph of (C, V) and the agenda as used in the proof of Theorem 22. All arcs between the vertices in R are forward arcs with a weight of at least $2|B| - \kappa + \ell$. The agenda is depicted by the left-to-right ordering of the vertices

For constructive control by deleting voters, we have two $W[2]$ -hardness results.

Theorem 22 *CCDV-Successive is $W[2]$ -hard with respect to the number of deleted votes. This holds even when the distinguished candidate is the last one in the agenda.*

Proof We prove the theorem by a reduction from the RBDS problem. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . Without loss of generality, we assume that $|B| \geq \kappa$. Similar to the proof of Theorem 2, we assume that every red vertex has the same degree, which we denote by ℓ . We create an instance of CCDV-Successive as follows. For each red vertex we create one candidate denoted still by the same symbol for simplicity. In addition, we create three candidates q , p , and p' , where p is the distinguished candidate. Let $C = R \cup \{q, p, p'\}$. The agenda is $\triangleright = (q, \vec{R}, p', p)$. We create the following $2|B| + 2\ell - \kappa$ votes:

- a multiset V_1 of $|B| - \kappa$ votes with the preference $\vec{R} \ p \ p' \ q$;
- a multiset V_2 of $\ell - 1$ votes with the preference $q \ p \ p' \ \vec{R}$;
- a singleton V_3 of one vote with the preference $q \ \vec{R} \ p \ p'$;
- a multiset V_4 of ℓ votes with the preference $p' \ p \ q \ \vec{R}$; and
- for each blue vertex $b \in B$, one vote \succ_b with the preference

$$q \ (\vec{R}[N_G(b)]) \ p' \ p \ (\vec{R} \setminus N_G(b)).$$

Let V_B be the multiset of votes corresponding to the blue vertices in B . Let $V = V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_B$. Let $k = \kappa$. The CCDV-Successive instance is $(C, p, V, \triangleright, k)$.

The weighted majority graph of (C, V) and the agenda \triangleright are illustrated in Fig. 9. We prove the correctness of the reduction as follows.

(\Rightarrow) Suppose that there is a $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . Let $V' = \{\succ_b : b \in B'\}$ be the set of votes corresponding to B' . Clearly, $|V'| = \kappa = k$. Let $E = (C, V \setminus V')$. We shall show that p is the successive winner of E with respect to \triangleright . First, as there are exactly $(\ell - 1) + 1 + (|B| - \kappa) = |B| + \ell - \kappa$ votes in $V \setminus V'$ ranking q in the top, $|V \setminus V'| = 2|B| + 2\ell - 2\kappa$, and q is the first one in the agenda, we know that q is not the successive winner of E . Let r be any candidate from R . Let $S(r)$ be the set of r 's successors in \triangleright . Clearly, all votes in $V_1 \cup V_3$ rank r before $S(r)$. Let $b \in B'$ be a blue vertex dominating r . It holds that $r \succ_b p$. As exactly ℓ votes in V_B rank r before p , this implies that $V_B \setminus V'$ contains at most $\ell - 1$ votes ranking r before p . To summarize, there can be at most $|V_1| + |V_3| + \ell - 1 = |B| - \kappa + \ell$ votes in $V \setminus V'$ ranking r before $S(r)$, excluding the winning chance of r . Observe that p

ties with p' in E , which excludes the winning chance of r . We now know that none of the predecessors of p wins E , and that p is the successive winner of E with respect to the agenda \triangleright , as it is the last candidate in \triangleright .

(\Leftarrow) Suppose that there exists $V' \subseteq V$ such that $|V'| \leq k$ and p is the successive winner of $E = (C, V \setminus V')$ with respect to \triangleright . Observe that all votes in V' must rank q in the first place and $|V'| = \kappa$, since otherwise q majority-dominates the set of all her successors in E , which contradicts that p wins E . There are three groups of votes ranking q in the first place: those in V_2 , V_3 , and V_B . However, observe that it must hold that $V' \subseteq V_B$, since otherwise p' would beat p in E . Given that p is the only successor of p' , this contradicts the fact that p is the successive winner of E . Let $B' = \{b \in B : \succ_b \in V'\}$ be the set of blue vertices corresponding to V' . We claim that B' dominates R . Assume, for the sake of contradiction, that this is not the case. Let r be the leftmost red vertex in the agenda \triangleright not dominated by any vertex from B' . Then, all votes in V' rank p before r . This implies that in $V \setminus V'$ there are at least $|V_1| + |V_3| + \ell = (|B| - \kappa) + 1 + \ell$ votes ranking r before all successors of r (the last term ℓ corresponds to the ℓ votes in $\{\succ_b : b \in B, r \in N_G(b)\}$ which is a subset of $V \setminus V'$). However, as $|V \setminus V'| = 2|B| - 2\kappa + 2\ell$, this implies that r is the successive winner of E , contradicting the winning of p . So, we know that B' dominates R . This implies that the RBDS instance is a Yes-instance. \square

We can show the W[2]-hardness of CCDV-Successive with respect to the dual parameter of the number of deleted votes.

Theorem 23 *CCDV-Successive is W[2]-hard with respect to the number of remaining votes. Moreover, this holds even when the distinguished candidate is the last one in the agenda.*

Proof The reduction for the W[2]-hardness of CCDC-Full-Amendment in the proof of Theorem 13 directly applies here. The correctness argument is also the same (one needs only to replace “full-amendment” with “successive”). \square

However, if the distinguished candidate has only a few predecessors, the problem becomes tractable from the parameterized complexity point of view. As a matter of fact, we have a result for a constructive control problem to which both CCAV- τ and CCDV- τ are polynomial-time Turing reducible.

EXACT CONSTRUCTIVE CONTROL BY EDITING VOTERS for τ (E-CCEV- τ)

- Given:** A set C of candidates, a distinguished candidate $p \in C$, a multiset V of registered votes over C , a multiset W of unregistered votes over C , an agenda \triangleright on C , and two nonnegative integers k and k' .
- Question:** Are there $V' \subseteq V$ and $W' \subseteq W$ such that $|V'| = k$, $|W'| = k'$, and p is the τ winner of $(C, V' \cup W')$ with respect to \triangleright ?
-

Theorem 24 *E-CCEV-Successive is FPT with respect to the number of predecessors of the distinguished candidate in the agenda.*

Proof Let $(C, p, V, W, \triangleright, k, k')$ be an instance of E-CCEV-Successive. Let $m = |C|$. Let $\triangleright[\ell] = p$ for some $\ell \in [m]$. Therefore, the distinguished candidate p has $\ell - 1$ predecessors. We present a natural ILP formulation for the given instance as follows.

We partition V (respectively, W) into 2^ℓ submultisets, each identified by an ℓ -length 1-0 vector \mathbf{s} and denoted by $V_{\mathbf{s}}$ (respectively, $W_{\mathbf{s}}$). Precisely, a vote \succ from V (respectively, W) belongs to $V_{\mathbf{s}}$ (respectively, $W_{\mathbf{s}}$) if and only if the following condition holds:

- For each $i \in [\ell]$, $\mathbf{s}[i] = 1$ if and only if there exists some $j \in [m]$ with $j > i$ such that $\triangleright[j] \succ \triangleright[i]$, i.e., at least one of $\triangleright[i]$'s successors is ranked before $\triangleright[i]$ in the vote \succ .

For each submultiset $V_{\mathbf{s}}$, we introduce a nonnegative integer variable $x_{\mathbf{s}}$, and for each submultiset $W_{\mathbf{s}}$, we introduce a nonnegative integer variable $y_{\mathbf{s}}$. The variables $x_{\mathbf{s}}$ and $y_{\mathbf{s}}$ indicate, respectively, the number of votes from $V_{\mathbf{s}}$ and $W_{\mathbf{s}}$ that are contained in a certain feasible solution. Let \mathbf{S} be the set of all ℓ -length 1-0 vectors. The constraints are as follows.

- As we aim to find two submultisets $V' \subseteq V$ and $W' \subseteq W$ of cardinalities respectively k and k' , we have that $\sum_{\mathbf{s} \in \mathbf{S}} x_{\mathbf{s}} = k$ and $\sum_{\mathbf{s} \in \mathbf{S}} y_{\mathbf{s}} = k'$.
- For each $\mathbf{s} \in \mathbf{S}$ we have that $0 \leq x_{\mathbf{s}} \leq |V_{\mathbf{s}}|$ and $0 \leq y_{\mathbf{s}} \leq |W_{\mathbf{s}}|$.
- For each $i \in [\ell - 1]$, in order to ensure that $\triangleright[i]$ is not the successive winner of $(C, V' \cup W')$ with respect to \triangleright , we have that

$$\sum_{\mathbf{s} \in \mathbf{S}} \mathbf{s}[i] \cdot (x_{\mathbf{s}} + y_{\mathbf{s}}) \geq \frac{k + k'}{2}.$$

- To ensure that p is the successive winner of $(C, V' \cup W')$ with respect to \triangleright , we introduce

$$\sum_{\mathbf{s} \in \mathbf{S}} \mathbf{s}[\ell] \cdot (x_{\mathbf{s}} + y_{\mathbf{s}}) < \frac{k + k'}{2}.$$

As we have $2^{\ell+1}$ variables, by Lemma 2, the above ILP can be solved in FPT-time in ℓ . \square

From Theorem 24, we have the following corollary.

Corollary 7 CCAV-Successive and CCDV-Successive are FPT with respect to the number of predecessors of the distinguished candidate in the agenda.

Now we move on to destructive control by adding/deleting voters. We show that both problems become polynomial-time solvable for the successive procedure.

Theorem 25 E-DCEV-Successive is polynomial-time solvable.

Proof Let $I = (C, p, V, W, \triangleright, k, k')$ be an instance of E-DCEV-Successive. Let $m = |C|$ be the number of candidates. Let $p = \triangleright[\ell]$ for some $\ell \in [m]$. Moreover, let S be the set of the successors of p in the agenda \triangleright . We solve I as follows.

We first check whether there exist $V' \subseteq V$ and $W' \subseteq W$ of cardinalities k and k' , respectively, such that p does not majority-dominate S with respect to $V' \cup W'$. This can be achieved using the following procedure. Let V_p (respectively, W_p) denote the submultiset of V (respectively, W) that ranks p before all candidates in S , i.e., $V_p = \{> \in V : \{p\} > S\}$ (respectively, $W_p = \{> \in W : \{p\} > S\}$). Furthermore, let $V_{\bar{p}} = V \setminus V_p$ and let $W_{\bar{p}} = W \setminus W_p$. If $|V_{\bar{p}}| \geq k$, we let V' be any arbitrary submultiset of $V_{\bar{p}}$ of cardinality k . Otherwise, we let V' be the union of $V_{\bar{p}}$ and any submultiset of exactly $k - |V_{\bar{p}}|$ votes from V_p . The submultiset W' is defined analogously.

Now, if p does not majority-dominate S with respect to $V' \cup W'$, we conclude that the given instance I is a Yes-instance.

Otherwise, to prevent p from winning, some predecessor of p in the agenda \triangleright must win. In this case, we determine, for each predecessor $c = \triangleright[i]$ of p where $i < \ell$, whether there exist $V' \subseteq V$ and $W' \subseteq W$ with $|V'| = k$ and $|W'| = k'$ such that c majority-dominates $\triangleright[i + 1, m]$ with respect to $V' \cup W'$. This can be accomplished in polynomial time using a greedy algorithm. Specifically, let V_c denote the submultiset of votes in V that rank c before $\triangleright[i + 1, m]$, and let W_c denote the submultiset of votes in W that rank c before $\triangleright[i + 1, m]$. Then, we construct V' as a submultiset of V with cardinality k , including as many votes from V_c as possible. Similarly, we construct W' as a submultiset of W with cardinality k' , including as many votes from W_c as possible. If c majority-dominates $\triangleright[i + 1, m]$ in $(C, V' \cup W')$, we conclude that I is a Yes-instance.

If none of the predecessors of p provides a Yes-answer, we conclude that I is a No-instance. \square

From Theorem 25, we obtain the following corollary.

Corollary 8 *DCAV-Successive and DCDV-Successive are polynomial-time solvable.*

Now we study control by adding/deleting candidates for the successive procedure. The following result is easy to see.

Corollary 9 *Successive is immune to CCAC if the distinguished candidate is the first one in the agenda.*

However, as long as the distinguished candidate is not in the first place of the agenda, the problem becomes intractable from the parameterized complexity perspective, and this holds even when there are only two registered candidates.

Theorem 26 *CCAC-Successive is W[2]-hard with respect to the number of added candidates, as long as the distinguished candidate is not in the first place of the agenda. Moreover, this holds even when there are only two registered candidates.*

Proof We prove the theorem by a reduction from RBDS. Let (G, κ) be an instance of RBDS, where G is a bipartite graph with the bipartition (R, B) . We construct an instance $(C, p, D, V, \triangleright, k)$ of CCAC-Successive as follows. We create only two

registered candidates, denoted by p and q , where p is the distinguished candidate. Let $C = \{p, q\}$. Then, for each blue vertex $b \in B$ we create one unregistered candidate denoted still by b for simplicity. Let $D = B$. Let \triangleright be any agenda on $C \cup D$ such that q is in the first place (the relative order of other candidates can be set arbitrarily). We create the following votes:

- one vote with the preference $p \xrightarrow{\triangleright} q$;
- $|R|$ votes with the preference $q \xrightarrow{\triangleright} p$; and
- for each red vertex $r \in R$, one vote \succ_r with the preference

$$\left(\overrightarrow{B}[N_G(r)]\right) \quad q \quad p \quad \left(\overrightarrow{B} \setminus N_G(r)\right).$$

Let $V_R = \{\succ_r : r \in R\}$ denote the multiset of votes corresponding to the red vertices, and let V denote the multiset of all the $2|R| + 1$ votes created above. Lastly, let $k = \kappa$.

The construction of $(C, p, D, V, \triangleright, k)$ can clearly be completed in polynomial time. We now proceed to prove the correctness of the reduction.

(\Rightarrow) Suppose there exists a subset $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . Let $E = (B' \cup \{p, q\}, V)$. We will show that p is the successive winner of E with respect to \triangleright . First, observe that there are $|R| + 1$ votes ranking p before all candidates in B . Hence, it suffices to prove that q is not the successive winner of E . Since B' dominates R , for every vote \succ_r corresponding to a red vertex $r \in R$, there exists at least one $b \in B'$ such that b dominates r . Consequently, b is ranked before q in \succ_r . This implies that the $|R|$ votes with the preference $q \xrightarrow{\triangleright} p$ are all votes ranking q before B' . However, since there are a total of $2|R| + 1$ votes, q cannot be the successive winner of E . Thus, we are done.

(\Leftarrow) Suppose there exists a subset $B' \subseteq B$ such that $|B'| \leq k$ and p is the successive winner of the election $E = (B' \cup \{p, q\}, V)$ with respect to the agenda \triangleright . We show below that B' dominates R in G . Assume, for the sake of contradiction, that this is not the case, i.e., there exists a red vertex $r \in R$ such that $r \notin N_G(B')$. Then, the vote \succ_r corresponding to r ranks q before B' . Together with the $|R|$ votes with the preference $q \xrightarrow{\triangleright} p$, there are at least $|R| + 1$ votes ranking q in the first place in the election E , implying that q is the successive winner of E , a contradiction. \square

For constructive control by deleting candidates, we first present a intractability result.

Theorem 27 *CCDC-Successive is W[1]-hard with respect to the number of deleted candidates. This holds even when the distinguished candidate is the first one in the agenda.*

Proof We prove the theorem by a reduction from the CLIQUE problem. Let (G, κ) be an instance of CLIQUE, where $G = (U, F)$ is a graph. Let $n = |U|$ and $m = |F|$ be the number of vertices and the number of edges of G , respectively. Without loss of generality, we assume that $m \geq \kappa \cdot (\kappa - 1)$. For each vertex in G , we create one candidate denoted by the same symbol for simplicity. In addition, we create a distinguished candidate p . Let $C = U \cup \{p\}$. Let \triangleright be an agenda where p is the first candidate. We create the following votes.

- First, we create a multiset of $m - \kappa \cdot (\kappa - 1) + 1$ votes, each with the preference $p \vec{U}$.
- Then, for each edge $e \in F$ between two vertices u and u' , we create one vote, denoted \succ_e , with the preference $\vec{e} p (\vec{U} \setminus e)$. Let V_2 be the set of these m votes.

Let $V = V_1 \cup V_2$ be the multiset of the above $2m - \kappa \cdot (\kappa - 1) + 1$ created votes. Let $k = \kappa$, i.e., we are allowed to delete at most κ candidates. We have completed the construction of an instance $(C, p, V, \triangleright, k)$ of CCDC-Successive, which clearly can be done in polynomial time. In the following, we prove the correctness of the reduction.

(\Rightarrow) Assume that there is a clique $K \subseteq U$ of size κ in the graph G . We prove below that p is the successive winner of $(C \setminus K, V)$ with respect to \triangleright . Since p is the first candidate in the agenda, this amounts to proving that there are at least $m - \frac{\kappa \cdot (\kappa - 1)}{2} + 1$ votes ranking p before $U \setminus K$. Let $F(K)$ be the set of edges whose endpoints are both contained in K . As K is a clique of κ vertices, $F(K)$ consists of exactly $\frac{\kappa \cdot (\kappa - 1)}{2}$ edges. Let \succ_e be a vote corresponding to an edge $e \in F(K)$. From the definition of \succ_e , only the two candidates corresponding to the two endpoints of e are ranked before p in the vote. This implies that all the $\frac{\kappa \cdot (\kappa - 1)}{2}$ votes corresponding to $F(K)$ rank p before $U \setminus K$. Then, as all votes in V_1 rank p in the top, in total there are at least $|V_1| + \frac{\kappa \cdot (\kappa - 1)}{2} = m - \frac{\kappa \cdot (\kappa - 1)}{2} + 1$ votes in V ranking p before $U \setminus K$, we are done.

(\Leftarrow) Let $K \subseteq U$ be an arbitrary subset of at most $k = \kappa$ vertices (candidates). If $|K| < \kappa$ or K is not a clique, then there can be at most $\frac{\kappa \cdot (\kappa - 1)}{2} - 1$ edges with both endpoints in K . Following a similar reasoning as above, one can verify that at most $|V_1| + \frac{\kappa \cdot (\kappa - 1)}{2} - 1 = m - \frac{\kappa \cdot (\kappa - 1)}{2}$ votes rank p before $U \setminus K$. This implies that p cannot be the successive winner of $(C \setminus K, V)$ with respect to the agenda \triangleright . Consequently, if G does not contain a clique of size κ , the instance of CCDC-Successive constructed above is a No-instance. \square

Additionally, we show that the problem remains W[1]-hard with respect to the number of candidates not deleted.

Theorem 28 *CCDC-Successive is W[1]-hard with respect to the number of remaining candidates. This holds even when the distinguished candidate is the first one in the agenda.*

Proof We prove the theorem via a reduction from the BICLIQUE problem. Let (G, κ) be an instance of the BICLIQUE problem, where G is a bipartite graph with the bipartition (X, Y) . Let $m = |X|$, and let $n = |Y|$. We assume that $m > \kappa$ and $n > 2\kappa$, which does not change the W[1]-hardness of the BICLIQUE problem. (Otherwise, the problem can be solved in FPT-time with respect to κ .) We construct a CCDC-Successive instance as follows. For each vertex $x \in X$, we create one candidate denoted still by x . In addition, we create a candidate p which is the distinguished candidate. Let $C = X \cup \{p\}$. Let \triangleright be an agenda on C such that p is the first candidate. We create the following votes.

- For each vertex $y \in Y$, we create one vote \succ_y with the preference

$$(\vec{X} \setminus N_G(y)) p (\vec{X}[N_G(y)]).$$

For a given $Y' \subseteq Y$, let $V_{Y'}$ be the multiset of votes created for vertices in Y' .

- We create a multiset V' of $n - 2\kappa + 1$ votes, each with the preference $p \succ_{\vec{X}}$.

Let $V = V_Y \cup V'$ denote the multiset of the $2n - 2\kappa + 1$ votes constructed above. Let $k = m - \kappa$. The instance of CCDC-Successive is $(C, p, V, \triangleright, k)$.

We prove the correctness of the reduction as follows.

(\Rightarrow) Assume that G contains a biclique (X', Y') such that $|X'| = |Y'| = \kappa$. Let $E = (X' \cup \{p\}, V)$. We show that p is the successive winner of E with respect to \triangleright . First, as (X', Y') is a biclique in G , due to the above construction, every vote \succ_y where $y \in Y'$ ranks p before all candidates in X' . As all votes in V' rank p in the top, there are in total at least $|Y'| + |V'| = n - \kappa + 1$ votes ranking p before X' . As there are in total $2n - 2\kappa + 1$ votes, p is the successive winner of E .

(\Leftarrow) Assume that there is a subset $X' \subseteq X$ such that $|X'| \geq m - k = \kappa$ and p is the successive winner of $(X' \cup \{p\}, V)$ with respect to \triangleright . Observe that, since p is the first candidate in the agenda, for any subset $X'' \subseteq X'$, p remains as the successive winner of $(X'' \cup \{p\}, V)$. This observation allows us to assume that $|X'| = \kappa$. Given this, we know that there are in total at least $n - \kappa + 1$ votes in V ranking p before X' . Furthermore, since $|V'| = n - 2\kappa + 1$ and all votes in V' rank p in the top, it follows that there are at least κ votes in V_Y ranking p before X' . Let $V_{Y'} \subseteq V_Y$, where $Y' \subseteq Y$ is a submultiset of κ votes, each of which ranks p before X' . By the construction, every vertex in Y' is adjacent to all vertices in X' . In other words, (X', Y') forms a biclique of size κ in G . \square

However, if the distinguished candidate has only a constant number of successors, we can solve CCDC-Successive in polynomial time. In particular, we show that the problem is FPT with respect to the number of successors of the distinguished candidate.

Theorem 29 *CCDC-Successive can be solved in time $O^*(2^\ell)$, where ℓ is the number of successors of the distinguished candidate in the agenda.*

Proof Let $I = (C, p, V, \triangleright, k)$ be an instance of CCDC-SUCCESSIVE. Let C' denote the set of successors of p in the agenda \triangleright , and let $\ell = |C'|$. To solve the problem, we enumerate all subsets $S \subseteq C'$ containing at most k candidates. Each enumerated S represents a hypothesis that, in a desired feasible solution, precisely the candidates in S are deleted from C' .

For a fixed subset $S \subseteq C'$, we determine whether it is possible to expand S by including at most $k - |S|$ candidates from $C \setminus (C' \cup \{p\})$ such that p becomes the successive winner of $(C \setminus S, V)$ with respect to the restriction of \triangleright to $C \setminus S$. It follows that the given instance I is a Yes-instance if and only if there exists a subset $S \subseteq C'$ for which the answer to the above question is Yes.

Our algorithm proceeds as follows. For each enumerated subset S , we execute the following steps:

- (1). If p is the successive winner of $(C \setminus S, V)$ with respect to \triangleright , we conclude that I is a YES instance.
- (2). Otherwise, we exhaustively apply the following procedure:

Procedure: If the successive winner of $(C \setminus S, V)$ with respect to \triangleright precedes p in \triangleright , add the successive winner to S .

After the above procedure has been applied exhaustively, if p is the successive winner of $(C \setminus S, V)$ and $|S| \leq k$, we conclude that I is a Yes-instance. Otherwise, we discard the current subset S .

If all subsets $S \subseteq C'$ are discarded, we conclude that the given instance I is a No-instance.

Since there are at most 2^ℓ subsets S to enumerate, and the above procedure for each subset S can be executed in polynomial time, the overall algorithm runs in time $O^*(2^\ell)$. \square

Note that the W[2]-hardness reduction of CCAC-Successive in the proof of Theorem 26 can be adapted to establish the W[2]-hardness of DCAC-Successive by designating q as the distinguished candidate. In the following, we present a simplified variant of this reduction to show a similar result for a more restrictive case.

Theorem 30 *DCAC-Successive is W[2]-hard with respect to the number of added candidates. Moreover, this holds even when the distinguished candidate is the only registered candidate and is the first one in the agenda.*

Proof We prove the theorem by slightly modifying the reduction for CCAC-Successive presented in the proof of Theorem 26. Specifically, given an instance (G, κ) of RBDS, where G is a bipartite graph with the bipartition (R, B) , we construct an instance of DCAC-Successive as follows. We create only one registered candidate p . That is, $C = \{p\}$. Then, for each blue vertex $b \in B$, we create one unregistered candidate denoted still by the same symbol for simplicity. Let $D = B$ be the set of the unregistered candidates. The agenda is $\triangleright = (p, \vec{B})$. We create the following votes:

- $|R| - 1$ votes with the preference $p \vec{B}$; and
- for each $r \in R$, one vote \succ_r with the preference

$$(\vec{B}[N_G(r)]) \ p \ (\vec{B} \setminus N_G(r)).$$

Let $V_R = \{\succ_r : r \in R\}$ denote the multiset of votes corresponding to the red vertices, and let V denote the multiset of the above created $2|R| - 1$ votes. We set $k = \kappa$. The DCAC-Successive instance is $(C, p, D, V, \triangleright, k)$.

The construction can clearly be completed in polynomial time. We prove its correctness as follows.

(\Rightarrow) Suppose that there is a $B' \subseteq B$ such that $|B'| = \kappa$ and B' dominates R in G . We show that p is not the successive winner of $(B' \cup \{p\}, V)$ with respect to \triangleright . In fact, as B' dominates R , for every vote \succ_r corresponding to a red vertex $r \in R$, there exists $b \in B'$ such that b' dominates r in G , and hence b is ranked before p in \succ_r . This implies that the $|R| - 1$ votes in $V \setminus V_R$ are all those who rank p before B' . As we have in total $2|R| - 1$ votes and p is in the first place of the agenda \triangleright , p cannot be the successive winner of $(B' \cup \{p\}, V)$ with respect to \triangleright .

(\Leftarrow) Suppose there exists a subset $B' \subseteq B$ such that $|B'| \leq k$ and p is not the successive winner of $(B' \cup \{p\}, V)$ with respect to \triangleright . We aim to show that B' dominates R in G .

Assume, for the sake of contradiction, that there exists $r \in R$ such that $r \notin N_G(B')$. Observe that the distinguished candidate p is ranked before B' in \succ_r . Together with the votes from $V \setminus V_R$, this results in at least $|R|$ votes ranking p in the first position in the election $(B' \cup \{p\}, V)$. Since p is the first candidate in the agenda \triangleright , it follows that p must be the successive winner of $(B' \cup \{p\}, V)$ —a contradiction. \square

Next, we show that the same problem is tractable if the distinguished candidate has a back seat in the agenda.

Theorem 31 *DCAC-Successive can be solved in time $O^*(2^\ell)$, where ℓ is the number of successors of the distinguished candidate in the agenda.*

Proof Let $I = (C, p, D, V, \triangleright, k)$ be an instance of DCAC-Successive. Define D' as the set of successors of p contained in D with respect to the agenda \triangleright , and let $\ell = |D'|$. We enumerate all $S \subseteq D'$ of up to k candidates and, for each enumerated S , proceed as follows: If p is not the successive winner of $(C \cup S, V)$ with respect to \triangleright , we conclude that I is a Yes-instance. Otherwise, two cases arise:

Case: $|S| = k$.

In this case, we discard S .

Case: $|S| < k$.

In this case, we check if there is a candidate $d \in D \setminus D'$ such that d is the successive winner of $(C \cup S \cup \{d\}, V)$ with respect to \triangleright . If such a candidate exists, we conclude that I is a Yes-instance; otherwise, we discard S .

Since there are at most 2^ℓ possible choices for S , and for each S , the above algorithm can be executed in polynomial time, the overall running time of the algorithm is $O^*(2^\ell)$. \square

Now, we turn to the destructive control by deleting candidates. The following corollary is straightforward.

Corollary 10 *Successive is immune to DCDC if the distinguished candidate is the first in the agenda.*

Corollary 10 indicates that an election controller cannot take any effective action if they are limited to performing the candidate deletion operation and the distinguished candidate they aim to prevent from winning is the first in the agenda. However, if the distinguished candidate has at least one predecessor, the destructive election controller has an opportunity to influence the outcome. Nevertheless, as established in the following two theorems, the controller faces an intractable problem.

Theorem 32 *DCDC-Successive is W[1]-hard with respect to the number of deleted candidates, as long as the distinguished candidate is not the first one in the agenda.*

Proof We prove the theorem via a reduction from the CLIQUE problem. Let (G, κ) be an instance of CLIQUE where $G = (U, F)$. Let $m = |F|$ be the number of edges in G . Without loss of generality, we assume that $m \geq \frac{\kappa \cdot (\kappa - 1)}{2} > 0$. We create an instance $(C, p, V, \triangleright, k)$ of DCDC-Successive as follows. For each vertex in G , we create one candidate denoted by the same symbol for notational brevity. In addition, we create two candidates q and p , where p is the distinguished candidate. Let $C = U \cup \{q, p\}$. Let \triangleright be an agenda on C where q is the first candidate. We create the following votes.

- First, we create a multiset V_1 of $m - \frac{\kappa \cdot (\kappa - 1)}{2} + 1$ votes, each with the preference

$$q \succ p \succ \vec{U}.$$

- Second, we create a multiset V_2 of $\frac{\kappa \cdot (\kappa - 1)}{2}$ votes, each with the preference

$$p \succ \vec{U} \succ q.$$

- Third, for each edge $e \in F$, we create one vote \succ_e with the preference

$$\vec{e} \succ q \succ p \succ (\vec{U} \setminus e).$$

Let $V_F = \{\succ_e : e \in F\}$ be the set of the votes corresponding to the edges in G . We have that $|V_F| = m$.

Let $V = V_1 \cup V_2 \cup V_F$ be the multiset of all the $2m + 1$ votes created above. Let $k = \kappa$. We prove the correctness of the reduction below.

(\Rightarrow) Suppose that there is a clique K of κ vertices in G . Defining $E = (C \setminus K, V)$, we claim that q is the successive winner of E with respect to \triangleright . Let $F(K)$ be the set of edges whose both endpoints are in K , and let $V_{F(K)} = \{\succ_e : e \in F(K)\}$ be the set of votes corresponding to $F(K)$. Clearly, $|V_{F(K)}| = |F(K)| = \frac{\kappa \cdot (\kappa - 1)}{2}$. According to the construction of the votes, after deleting candidates from K , each vote in $V_{F(K)}$ ranks q in the first place. Therefore, in the election E there are at least $|V_{F(K)}| + |V_1| = m + 1$ votes ranking q in the first place, implying that q is the successive winner of E .

(\Leftarrow) Suppose that there is a subset $C' \subseteq C \setminus \{p\}$ of at most k candidates such that p is not the successive winner of $(C \setminus C', V)$ with respect to \triangleright . Let $E = (C \setminus C', V)$. Observe that p majority-dominates U , the set containing all possible successors of p . It follows that q is the successive winner of E , and $C' \subseteq U$. Then, as q is the first candidate in the agenda \triangleright and $|V| = 2m + 1$, there are at least $m + 1$ votes ranking q in the first place in E . This implies that, in the election E , at least $\frac{\kappa \cdot (\kappa - 1)}{2}$ votes in V_F rank q in the top. Let F' be the set of the edges corresponding to any arbitrary $\frac{\kappa \cdot (\kappa - 1)}{2}$ votes in E ranking q in the top. By the construction of the votes, we know that both endpoints of each edge in F' are contained in C' . As $|F'| = \frac{\kappa \cdot (\kappa - 1)}{2}$ and $|C'| \leq \kappa$, this is possible only if C' forms a clique of size κ in G . \square

Theorem 33 *DCDC-Successive is W[1]-hard with respect to the remaining candidates. This result holds as long as the distinguished candidate is not the first in the agenda.*

Proof We prove the theorem by a reduction from BICLIQUE. Let (G, κ) be an instance of BICLIQUE, where G is a bipartite graph with the bipartition (X, Y) . Let $m = |X|$ and let $n = |Y|$. Without loss of generality, we assume that $\min\{m, n\} > \kappa > 1$. We create an instance of DCDC-Successive as follows. First, for each vertex $x \in X$, we create one candidate denoted by the same symbol for notational brevity. Additionally, we introduce two candidates, q and p , where p is the distinguished candidate. Let $C = X \cup \{q, p\}$. There are $m + 2$ candidates in total. Let \triangleright be an agenda on C such that q is the first candidate. We create the following votes.

- First, we create a multiset V_1 of κ votes, each with the preference

$$p \succ \vec{X} \succ q.$$

- Second, we create a multiset V_2 of $n - \kappa + 1$ votes, each with the preference

$$q \succ p \succ \vec{X}.$$

- Third, for each vertex $y \in Y$, we create one vote \succ_y with the preference

$$(\vec{X} \setminus N_G(y)) \succ q \succ p \succ (\vec{X}[N_G(y)]).$$

For a given $Y' \subseteq Y$, we use $V_{Y'} = \{\succ_y : y \in Y'\}$ to denote the multiset of the votes corresponding to Y' . Let $V = V_1 \cup V_2 \cup V_Y$ be the multiset of all the $2n + 1$ votes created above. Let $k = m - \kappa$. The instance of DCDC-Successive is $(C, p, V, \triangleright, k)$.

In the following, we prove the correctness of the reduction, i.e., the BICLIQUE instance is a Yes-instance if and only if there is a subset $C' \subseteq C$ such that $p \in C'$, $|C'| \geq \kappa + 2$, and p is not the successive winner of (C', V) with respect to \triangleright .

(\Rightarrow) Assume that there are $X' \subseteq X$ and $Y' \subseteq Y$ such that $|X'| = |Y'| = \kappa$, and $X' \cup Y'$ induces a complete bipartite subgraph of G . Let $C' = X' \cup \{p, q\}$ and let $E = (C', V)$. We claim that q is the successive winner of E . Let y be a vertex in Y' . Due to the definition of \succ_y , q is ranked before $X' \cup \{p\}$ in \succ_y . Therefore, in total, there are at least $|Y'| + |V_2| = n + 1$ votes ranking q before $X' \cup \{p\}$ in E . Since $|V| = 2n + 1$ and $X' \cup \{p\}$ is the set of successors of q in E , we conclude that q is the successive winner of E .

(\Leftarrow) Observe that p majority-dominates the set of all its successors in (C, V) . Therefore, if p is not the winner after some candidates are deleted, it must be the case that q becomes the winner of the resulting election. As a consequence, let us assume that there is a subset $X' \subseteq X$ of at least κ candidates such that q is the successive winner of $(X' \cup \{q, p\}, V)$ with respect to \triangleright . Similar to the proof for the \Rightarrow direction, we know that there are at least κ votes in V_Y ranking q before $X' \cup \{p\}$. Let $V_{Y'}$, where $Y' \subseteq Y$, be the multiset of such votes. By the construction of the votes, it holds that $X' \subseteq N_G(y)$ for all $y \in Y'$. This implies that $G[X' \cup Y']$ is a complete bipartite graph. As $|X'| \geq \kappa$ and $|Y'| \geq \kappa$, the instance of BICLIQUE is a Yes-instance. \square

We have obtained numerous intractability results (NP-hardness, W[1]-hardness, or W[2]-hardness) for the special cases where the distinguished candidate has only a constant number of predecessors or successors. These hardness results lead to the following corollaries:

Corollary 11 *The following problems are paraNP-hard with respect to the number of predecessors of the distinguished candidate:*

- CCAV- τ and CCDV- τ for each $\tau \in \{\text{Amendment}, \text{Full-Amendment}\}$;
- X-Successive for each $X \in \{\text{CCAC}, \text{DCAC}, \text{CCDC}, \text{DCDC}\}$; and
- X-Full-Amendment for each $X \in \{\text{DCAV}, \text{DCDV}\}$.

Corollary 12 *The following problems are paraNP-hard with respect to the number of successors of the distinguished candidate:*

- CCAV- τ and CCDV- τ for each $\tau \in \{\text{Amendment}, \text{Full-Amendment}, \text{Successive}\}$;
- CCAC- τ for each $\tau \in \{\text{Full} - \text{Amendment}, \text{Successive}\}$;
- DCAV- τ and DCDV- $\tau \in \{\text{Amendment}, \text{Full-Amendment}\}$; and
- DCDC-Successive.

We also note that if a problem is shown to be intractable when the distinguished candidate p has ℓ predecessors (respectively, successors) for some constant ℓ , the same result can be extended to the case where p has $\ell + 1$ predecessors (respectively, successors). This can be achieved by introducing a new candidate who is immediately before (respectively, after) p in the agenda and is ranked last in all votes.

4 Algorithmic Lower Bounds

In this section, we discuss how the reductions established in Sect. 3 offer lower bounds on kernelization algorithms, exact algorithms, and approximation algorithms. The following notations are used in our discussions:

- m_r : number of registered candidates, i.e., $m_r = |C|$.
- m_u : number of unregistered candidates, i.e., $m_u = |D|$.
- m : total number of candidates, i.e., $m = |C| + |D|$.
- n_r : number of registered votes, i.e., $n_r = |V|$.
- n_u : number of unregistered votes, i.e., $n_u = |W|$.
- n : total number of votes, i.e., $n = |V| + |W|$.
- k : a given upper bound on the cardinality of a desired feasible solution.

4.1 Kernelization Lower Bounds

It is known that RBDS does not admit any polynomial kernels with respect to both $|R| + \kappa$ and $|B|$, assuming $\text{PH} \neq \Sigma^3_P$ (see, e.g., the book chapter edited by Cygan et al. [18], and the paper by Dom, Lokshtanov, and Saurabh [20]). In light of this result, many of our reductions imply the nonexistence of polynomial kernels for various parameters, as these reductions are, in fact, polynomial parameter transformations. To clarify our discussion, we reiterate the formal definitions of kernelization and polynomial parameter transformations.

Definition 2 (*Kernelization*) Let P be a parameterized problem. A kernelization algorithm (or kernelization) for P is an algorithm that takes as input an instance (X, κ) of P and outputs an instance (X', κ') of P such that the following four conditions hold simultaneously.

- The algorithm runs in polynomial time in the size of (X, κ) .
- (X, κ) is a Yes-instance of P if and only if (X', κ') is a Yes-instance of P .
- $|X'| \leq f(\kappa)$ for some computable function f depending only on κ .
- $\kappa' \leq g(\kappa)$ for some computable function g depending only on κ .

Table 3 Nonexistence of polynomial kernels with respect to various parameters, assuming $\text{PH} \neq \Sigma_p^3$

	CCAV- τ	CCDV- τ	CCAC- τ	CCDC- τ
Amendment	$n, m + n_{\tau} + k$ (Thm. 1)	$m + k$ (Thm. 2) $n, m + n - k$ (Thm. 3)	\	\
Full-amendment	?	$m + k$ (Thm. 12) $n, m + n - k$ (Thm. 13)	$m_u, m_{\tau} + k$ (Thm. 17)	\
Successive	?	$n, m + k$ (Thm. 22) $m + n - k$ (Thm. 23)	$m, n + k$ (Thm. 26)	?
	DCAV- τ	DCDV- τ	DCAC- τ	DCDC- τ
Amendment	?	$m + k$ (Thm. 7) $n, m + n - k$ (Thm. 8)	\	\
Full-amendment	?	$n, m + k$ (Thm. 15)	\	\
Successive	\	\	$m, n + k$ (Thm. 30)	?

Entries marked with “\” indicate that the corresponding problems are solvable in polynomial time, while entries marked with “?” signify that our reductions do not provide a conclusive result regarding the nonexistence of polynomial kernels for these problems, given the current state of known lower bound techniques

The new instance (X', κ') in the above definition is called a kernel, and the size of X' is referred to as the size of the kernel. If $f(\kappa)$ is a polynomial function of κ , we say that (X', κ') is a polynomial kernel, and the problem P is said to admit a polynomial kernel.

Definition 3 (*Polynomial Parameter Transformation*) Let P and Q be two parameterized problems. A polynomial parameter transformation from P to Q is an algorithm that takes as input an instance (X, κ) of P and outputs an instance (X', κ') of Q such that the following three conditions are satisfied:

- The algorithm runs in polynomial time with respect to the size of (X, κ) .
- (X, κ) is a Yes-instance of P if and only if (X', κ') is a Yes-instance of Q .
- $\kappa' \leq g(\kappa)$, where g is a polynomial function of κ .

Lemma 5 [9, 18, 20] *Let P and Q be two parameterized problems such that the unparameterized version of P is NP-complete and the unparameterized version of Q is in NP. Then, if there exists a polynomial parameter transformation from P to Q , it follows that Q admitting a polynomial kernel implies that P also admits a polynomial kernel.*

With these notions in mind, and recalling that (1) the unparameterized version of RBDS is NP-complete, (2) all election control problems considered in the paper are in NP, (3) RBDS does not admit any polynomial kernels with respect to both $|R| + \kappa$ and $|B|$ assuming $\text{PH} \neq \Sigma_p^3$, and (4) many of our reductions are polynomial parameter transformations, we can summarize the nonexistence of polynomial kernels for various election control problems in Table 3.

4.2 Exact Algorithm Lower Bounds

Assuming the Strong Exponential Time Hypothesis (SETH), it is known that RBDS cannot be solved in time $O^*((2 - \epsilon)^{|B|})$ (see, e.g., the book chapter by Cygan et al. [18]). Moreover, unless the Exponential Time Hypothesis (ETH) fails, CLIQUE cannot be solved in time $f(\kappa) \cdot z^{o(\kappa)}$ (where z denotes the number of vertices), and RBDS cannot be solved in time $f(\kappa) \cdot |B|^{o(\kappa)}$ for any computable function f in κ (see, e.g., the work of Chen et al. [15]). These lower bounds, combined with several of our reductions, suggest that brute-force-based algorithms for many election control problems are essentially optimal. A summary of these results can be found in Table 4.

4.3 Inapproximability Results

This section is dedicated to the inapproximability consequences of the previously established reductions. For each of the eight standard election control problems, we examine its optimal version, where the goal is to add or delete the minimum number of voters or candidates such that the distinguished candidate becomes the winner (in the case of constructive control problems) or is not the winner (in the case of destructive control problems).

Let OPTIMAL RBDS be the optimal version of RBDS, where the objective is to select a minimum number of blue vertices that dominate all red vertices. It has long been known that unless $\text{NP} \subseteq \text{DTIME}(n^{\log \log n})$, OPTIMAL RBDS cannot be approximated in polynomial time within a factor of $(1 - \epsilon) \ln |R|$ (see, e.g., the work of Feige [29]). This result has been recently improved by Dinur and Steurer [19] who showed that the same lower bound holds under the assumption $\text{P} \neq \text{NP}$. Building on these results and our previous reductions, we obtain numerous inapproximability results for the optimal versions of the election control problems, as summarized in Table 5.

5 Conclusion

We have investigated eight standard control problems under the h -amendment procedure and the successive procedure. The 1-amendment procedure and the successive procedure are currently two of the most important sequential voting procedures used in practical parliamentary and legislative decision-making. The full-amendment procedure is a special case of h -amendment where h equals the number of candidates minus one. An advantage of the full-amendment procedure is that for the same election and the same agenda, if the full-amendment winner does not coincide with the amendment (respectively, successive) winner, then the former beats the latter. However, a disadvantage of the full-amendment procedure is that in the worst case it may need the head-to-head comparisons among all candidates to determine the winner, while the amendment procedure only needs voters to compare $m - 1$ pairs of candidates, where m is the number of candidates.

Table 4 Lower bounds for exact algorithms solving election control problems

	CCAV- τ	CCDV- τ	CCAC- τ	CCDC- τ
Amendment	$O^*((2 - \epsilon)^{n_u})$ (Thm. 1)	$f(k) \cdot n^{o(k)}$ (Thm. 2)	\setminus	\setminus
full-	$f(k) \cdot n_u^{o(k)}$ (Thm. 1)	$f(n - k) \cdot n^{o(n-k)}$ (Thm. 3)		
Amendment	?	$f(k) \cdot n^{o(k)}$ (Thm. 12)	$f(k) \cdot m_u^{o(k)}$ (Thm. 17)	\setminus
Successive	?	$f(n - k) \cdot n^{o(n-k)}$ (Thm. 13)	$f(k) \cdot m_u^{o(k)}$ (Thm. 26)	$f(k) \cdot m^{o(k)}$ (Thm. 27)
		$f(k) \cdot n^{o(k)}$ (Thm. 22)	DCAC- τ	DCDC- τ
		$f(n - k) \cdot n^{o(n-k)}$ (Thm. 23)	\setminus	\setminus
Amendment	DCAV- τ	DCDV- τ		
	?	$f(k) \cdot n^{o(k)}$ (Thm. 7)		
		$f(n - k) \cdot n^{o(n-k)}$ (Thm. 8)		
Full-amendment	?	$f(k) \cdot n^{o(k)}$ (Thm. 15)	$f(k) \cdot m^{o(k)}$ (Thm. 30)	$f(k) \cdot m^{o(k)}$ (Thm. 32)
Successive	\setminus	\setminus		

The lower bound $O^*((2 - \epsilon)^{n_u})$ for CCAV-Amendment is based on the SETH, while all other lower bounds are based on the ETH. Entries filled with “ \setminus ” indicate that the corresponding problems are solvable in polynomial time. Entries filled with “?” signify that, based on current lower bound techniques, our reductions do not imply lower bounds matching the running times of brute-force-based algorithms

Table 5 Lower bounds for approximation algorithms based on the assumption $P \neq NP$

	CCAV- τ	CCDV- τ	CCAC- τ	CCDC- τ
Amendment	$(1 - \epsilon) \ln m$ (Thm. 1)	$(1 - \epsilon) \ln m$ (Thm. 2)	\	\
Full-amendment	?	$(1 - \epsilon) \ln m$ (Thm. 12)	$(1 - \epsilon) \ln m_r$ (Thm. 17)	\
Successive	?	$(1 - \epsilon) \ln m$ (Thm. 22)	$(1 - \epsilon) \ln n$ (Thm. 26)	?
	DCAV- τ	DCDV- τ	DCAC- τ	DCDC- τ
Amendment	?	$(1 - \epsilon) \ln m$ (Thm. 7)	\	\
Full-amendment	?	$(1 - \epsilon) \ln m$ (Thm. 15)	\	\
Successive	\	\	?	?

Entries filled with “\” mean the corresponding problems are polynomial-time solvable. Entries filled with “?” mean that our reductions established in the paper do not imply meaningful inapproximability results

Our study offers a comprehensive understanding of the parameterized complexity of these problems. In particular, we obtained both parameterized intractability results ($W[1]$ -hardness results or $W[2]$ -hardness results) for the special cases where the distinguished candidate is the first candidate or the last candidate in the agenda, and many tractability results including some FPT-algorithms and several polynomial-time algorithms. For a summary of our concrete results over the amendment procedure, the full-amendment procedure, and the successive procedure, we refer to Table 1. Our study also yields a lot of algorithmic lower bounds, as summarized in Tables 3–5.

Overall, our investigation conveys the following message.

- The amendment procedure and the successive procedure behave quite differently regarding their resistance to these control types: the amendment procedure demonstrates greater resistance to voter control, whereas the successive procedure proves more resistant to candidate control.
- From a computational complexity perspective, the amendment procedure and the full-amendment procedure exhibit similar behavior concerning their resistance to the eight standard control types: both are resistant to the four voter control types and vulnerable to most candidate control types, with the exception that CCAC-Full-Amendment is computationally hard.
- From the parameterized complexity perspective, the full-amendment procedure outperforms the amendment procedure in the sense that election control problems for the full-amendment procedure are at least as difficult to solve as the same problems for the amendment procedure. For instance, we showed that destructive control by adding/deleting voters for the amendment procedure is FPT with respect to the number of predecessors of the distinguished candidate (Corollary 4). However, the same problems for the full-amendment procedure are NP-hard, even when the distinguished candidate has a constant number of predecessors (Theorem 14 and Theorem 15).
- The position of the distinguished candidate on the agenda has a significant impact on the parameterized complexity of the problems. Specifically, from a complexity-theoretic perspective, most of these problems are more challenging to solve when the distinguished candidate p has a back position on the agenda than when p has a front position.

A summary of our results and those by Black [8], Farquharson [28], and Bredereck et al. [11] reveals that there is no clear-cut conclusion on which procedure is the best one because these procedures all have their own advantages regarding resistance to different strategic behaviors. Nevertheless, in many practical applications, not all types of strategic behavior are likely to occur. If only one or two types of strategic behavior are likely to happen, our study might provide persuasive guidance on the choice of the procedures.

Finally, we touch upon some prominent topics for future research.

- First, as our study solely focuses on worst-case analysis, it is important to conduct experimental investigations to examine if the standard control problems are genuinely difficult to solve in practice.
- Second, beyond the eight control types discussed in this paper, it is worthwhile to investigate additional control problems, such as constructive and destructive control by partitioning voters or candidates. For further insights on these topics, see the book chapter by Baumeister and Rothe [7], the work of Erdélyi, Hemaspaandra, and Hemaspaandra [24], and the book chapter by Faliszewski and Rothe [27].
- Third, exploring the parameterized complexity of the problems with respect to structural parameters, such as the single-peaked width and the single-crossing width, is another avenue of investigation. Definitions of these structural parameters and related results can be found in the papers of Cornaz, Galand, and Spanjaard [16, 17], and Yang and Guo [56]. We note that since all h -amendment procedures are Condorcet-consistent, election control problems under these procedures can be solved in polynomial time when restricted to single-peaked or single-crossing domains, given certain mild restrictions (e.g., assuming an odd number of voters). This result leverages the techniques developed by Brandt et al. [10] and Magiera and Faliszewski [40].
- Another interesting topic is investigating whether the P-results for destructive control problems can be extended to FPT-results for the corresponding resolute control problems proposed first by Yang and Wang [57], where there are multiple distinguished candidates whom the election controller would like to make nonwinners.
- Finally, numerous open problems remain to be addressed following our study. Beyond the unresolved questions concerning the parameterized complexity of various candidate control problems with respect to the number of voters, as discussed in Sect. 2.4, the complexity of the following problems for constant $h \geq 2$ also remains open: X - h -Amendment for $X \in \{\text{CCAC}, \text{CCDC}, \text{DCAC}, \text{DCDC}\}$, and X -($m - h$)-Amendment for $X \in \{\text{CCDC}, \text{DCDC}\}$.

Acknowledgements The author sincerely thanks the anonymous reviewers of AAMAS 2022 for their valuable feedback on an earlier version of this paper. Additionally, the author extends heartfelt gratitude to the anonymous reviewer of Algorithmica for their insightful and constructive comments, which significantly enhanced the quality of this work. In particular, the author acknowledges the anonymous reviewer of Algorithmica for their invaluable feedback on Reduction Rule 2. Their input helped correct a previously flawed statement and led to the development of the counterexample presented in Example 2.

Author Contributions All authors contribute equally. All authors reviewed the manuscript.

Funding Open Access Funding Enabled and organized by Projekt DEAL.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Apesteguia, J., Ballester, M.A., Masatliloglu, Y.: A foundation for strategic agenda voting. *Games Econ. Behav.* **87**, 91–99 (2014)
2. Bang-Jensen J., Gutin, G.Z. (Eds.) *Classes of Directed Graphs*. Springer Monographs in Mathematics. Springer (2018)
3. Barberá, S., Gerber, A.: Sequential voting and agenda manipulation. *Theor. Econ.* **12**, 211–247 (2017)
4. Bartholdi, J.J., III., Tovey, C.A., Trick, M.A.: How hard is it to control an election? *Math. Comput. Model.* **16**(8–9), 27–40 (1992)
5. Baumeister, D., Erdélyi, G., Erdélyi, O.J., Rothe, J.: Control in judgment aggregation. In: STAIRS, pp. 23–34 (2012)
6. Baumeister, D., Erdélyi, G., Erdélyi, O.J., Rothe, J., Selker, A.-K.: Complexity of control in judgment aggregation for uniform premise-based quota rules. *J. Comput. Syst. Sci.* **112**, 13–33 (2020)
7. Baumeister, D., Rothe, J.: Preference aggregation by voting. In: Rothe, J. (Ed.) *Economics and Computation: An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, chapter 4, pp. 197–325. Springer, Heidelberg (2016)
8. Black, D.: *The Theory of Committees and Elections*. Cambridge University Press, Cambridge (1958)
9. Bodlaender, H. L., Thomassé, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. In: *ESA*, pp. 635–646 (2009)
10. Brandt, F., Brill, M., Hemaspaandra, E., Hemaspaandra, L.A.: Bypassing combinatorial protections: polynomial-time algorithms for single-peaked electorates. *J. Artif. Intell. Res.* **53**, 439–496 (2015)
11. Bredereck, R., Chen, J., Niedermeier, R., Walsh, T.: Parliamentary voting procedures: agenda control, manipulation, and uncertainty. *J. Artif. Intell. Res.* **59**, 133–173 (2017)
12. Brill, M., Schmidt-Kraepelin, U., Suksompong, W.: Margin of victory for tournament solutions. *Artif. Intell.* **302**, Nr. 103600 (2022)
13. Cesati, M.: Perfect Code is $W[1]$ -complete. *Inf. Process. Lett.* **81**(3), 163–168 (2002)
14. Chen, J., Faliszewski, P., Niedermeier, R., Talmon, N.: Elections with few voters: candidate control can be easy. *J. Artif. Intell. Res.* **60**, 937–1002 (2017)
15. Chen, J., Huang, X., Kanj, I.A., Xia, G.: Strong computational lower bounds via parameterized complexity. *J. Comput. Syst. Sci.* **72**(8), 1346–1367 (2006)
16. Cornaz, D., Galand, L., Spanjaard, O.: Bounded single-peaked width and proportional representation. In: *ECAI*, pp. 270–275 (2012)
17. Cornaz, D., Galand, L., Spanjaard, O.: Kemeny elections with bounded single-peaked or single-crossing width. In: *IJCAI*, pp. 76–82 (2013)
18. Cygan, M., Fomin, F. V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Lower bounds based on the exponential-time hypothesis. In: *Parameterized Algorithms*, Chapter 14, pp. 467–521. Springer (2015)
19. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: *STOC*, pp. 624–633 (2014)
20. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and IDs. *ACM Trans. Algor.* **11**(2), Nr. 13 (2014)

21. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness II: on completeness for W[1]. *Theor. Comput. Sci.* **141**(1–2), 109–131 (1995)
22. Downey, R.G., Fellows, M.R.: *Fundamentals of Parameterized Complexity*. Texts in Computer Science, Springer, New York (2013)
23. Downey, R. G., Fellows, M. R., Stege, U.: Parameterized complexity: a framework for systematically confronting computational intractability. In: *Contemporary Trends in Discrete Mathematics: From DIMACS and DIMATIA to the Future*, pp. 49–99 (1999)
24. Erdélyi, G., Hemaspaandra, E., Hemaspaandra, L.A.: More natural models of electoral control by partition. In: *ADT*, pp. 396–413 (2015)
25. Erdélyi, G., Neveling, M., Reger, C., Rothe, J., Yang, Y., Zorn, R.: Towards completing the puzzle: Complexity of control by replacing, adding, and deleting candidates or voters. *Auton. Agent Multi-Ag.* **35**(2), Nr. 41 (2021)
26. Erdélyi, G., Reger, C., Yang, Y.: The complexity of bribery and control in group identification. *Auton. Agent Multi-Ag.* **34**(1), Nr. 8 (2020)
27. Faliszewski, P., Rothe, J.: Control and bribery in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (Eds.) *Handbook of Computational Social Choice*, Chapter 7, pp. 146–168. Cambridge University Press (2016)
28. Farquharson, R.: *Theory of Voting*. Yale University Press, New Haven (1969)
29. Feige, U.: A threshold of $\ln n$ for approximating set cover. *J. ACM* **45**(4), 634–652 (1998)
30. Fomin, F.V., Kratsch, D., Woeginger, G.J.: Exact (exponential) algorithms for the dominating set problem. In: *WG*, pp. 245–256 (2004)
31. Frank, A., Tardos, É.: An application of simultaneous Diophantine approximation in combinatorial optimization. *Combinatorica* **7**(1), 49–65 (1987)
32. Garey, M., Johnson, D.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York (1979)
33. Hemaspaandra, E., Hemaspaandra, L.A., Rothe, J.: Anyone but him: the complexity of precluding an alternative. *Artif. Intell.* **171**(5–6), 255–285 (2007)
34. Horan, S.: Agendas in legislative decision-making. *Theor. Econ.* **16**(1), 235–274 (2021)
35. Kannan, R.: Minkowski’s convex body theorem and integer programming. *Math. Oper. Res.* **12**(3), 415–440 (1987)
36. Lenstra, H.W., Jr.: Integer programming with a fixed number of variables. *Math. Oper. Res.* **8**(4), 538–548 (1983)
37. Lin, B.: The parameterized complexity of k -biclique. In: *SODA*, pp. 605–615 (2015)
38. Liu, H., Feng, H., Zhu, D., Luan, J.: Parameterized computational complexity of control problems in voting systems. *Theor. Comput. Sci.* **410**(27–29), 2746–2753 (2009)
39. Liu, H., Zhu, D.: Parameterized complexity of control by voter selection in Maximin, Copeland, Borda, Bucklin, and Approval election systems. *Theor. Comput. Sci.* **498**, 115–123 (2013)
40. Magiera, K., Faliszewski, P.: How hard is control in single-crossing elections? *Auton. Agent Multi-Ag.* **31**(3), 606–627 (2017)
41. McGarvey, D.C.: A theorem on the construction of voting paradoxes. *Econometrica* **21**(4), 608–610 (1953)
42. Meir, R., Procaccia, A.D., Rosenschein, J.S., Zohar, A.: Complexity of strategic behavior in multi-winner elections. *J. Artif. Intell. Res.* **33**, 149–178 (2008)
43. Miller, N.R.: Graph-theoretical approaches to the theory of voting. *AM. J. Polit. Sci.* **21**, 769–803 (1977)
44. Neveling, M., Rothe, J.: Control complexity in Borda elections: solving all open cases of offline control and some cases of online control. *Artif. Intell.* **298**, Nr. 103508 (2021)
45. Niedermeier, R.: *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, Oxford (2006)
46. Rasch, B.E.: Parliamentary floor voting procedures and agenda setting in Europe. *Legis. Stud. Q.* **25**(1), 3–23 (2000)
47. Rasch, B.E.: Insincere voting under the successive procedure. *Public Choice* **158**(3–4), 499–511 (2014)
48. Tovey, C.A.: Tutorial on computational complexity. *Interfaces* **32**(3), 30–61 (2002)
49. West, D.B.: *Introduction to Graph Theory*. Prentice-Hall, London (2000)
50. Yang, Y.: The complexity of control and bribery in majority judgment. In: *AAMAS*, pp. 1169–1177 (2017)
51. Yang, Y.: On the complexity of Borda control in single-peaked elections. In: *AAMAS*, pp. 1178–1186 (2017)

52. Yang, Y.: Complexity of manipulating and controlling approval-based multiwinner voting. In: IJCAI, pp. 637–643 (2019)
53. Yang, Y.: On the complexity of constructive control under nearly single-peaked preferences. In: ECAI, pp. 243–250 (2020)
54. Yang, Y.: On the complexity of controlling amendment and successive winners. In: AAMAS, pp. 1768–1770 (2022)
55. Yang, Y., Dimitrov, D.: How hard is it to control a group? *Auton. Agent Multi-Ag.* **32**(5), 672–692 (2018)
56. Yang, Y., Guo, J.: Controlling elections with bounded single-peaked width. In: AAMAS, pp. 629–636 (2014)
57. Yang, Y., Wang, J.: Anyone but them: the complexity challenge for a resolute election controller. In: AAMAS, pp. 1133–1141 (2017)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.