

Combinatorial voter control in elections

Laurent Bulteau, Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, Nimrod Talmon

► **To cite this version:**

Laurent Bulteau, Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, Nimrod Talmon. Combinatorial voter control in elections. Theoretical Computer Science, Elsevier, 2015, 589, pp.99-120. 10.1016/j.tcs.2015.04.023 . hal-01260623

HAL Id: hal-01260623

<https://hal.archives-ouvertes.fr/hal-01260623>

Submitted on 22 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Combinatorial Voter Control in Elections^{☆,☆☆}

Laurent Bulteau^{b,1}, Jiehua Chen^{b,2}, Piotr Faliszewski^{c,3},
Rolf Niedermeier^b, Nimrod Talmon^{b,4}

^a *Inria Grenoble - Rhône-Alpes / LBBE, Université Lyon 1, CNRS, UMR5558,
F-69622 Villeurbanne, France*

^b *Institut für Softwaretechnik und Theoretische Informatik, Technische Universität
Berlin, D-10587 Berlin, Germany*

^c *AGH University of Science and Technology, P-30-059 Kraków, Poland*

Abstract

Voter control problems model situations such as an external agent trying to affect the result of an election by adding voters, for example by convincing some voters to vote who would otherwise not attend the election. Traditionally, voters are added one at a time, with the goal of making a distinguished alternative win by adding a minimum number of voters. In this paper, we initiate the study of combinatorial variants of control by adding voters. In our setting, when we choose to add a voter v , we also have to add a whole bundle $\kappa(v)$ of voters associated with v . We study the computational complexity of this problem for two of the most basic voting rules, namely the Plurality rule and the Condorcet rule.

Keywords: Voting, NP-hard election control problem, domain restrictions,

[☆] A preliminary short version of this work has been presented at the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS 2014), Budapest, August 2014 [12]. In this long version we provide an additional fixed-parameter tractability result (Theorem 5) and all proofs that were omitted in the conference version.

^{☆☆} This work has been partly supported by COST Action IC1205 on Computational Social Choice.

Email addresses: `l.bulteau@gmail.com` (Laurent Bulteau),
`jiehua.chen@tu-berlin.de` (Jiehua Chen), `faliszew@agh.edu.pl` (Piotr Faliszewski),
`rolf.niedermeier@tu-berlin.de` (Rolf Niedermeier), `nimrodtalmon77@gmail.com`
(Nimrod Talmon)

¹Supported by the Alexander von Humboldt Foundation while visiting TU Berlin.

²Supported by the Studienstiftung des Deutschen Volkes.

³Supported by a DFG Mercator fellowship within project PAWS (NI 369/10) while staying at TU Berlin, and by AGH University grant 11.11.230.124 afterward.

⁴Supported by DFG, RTG “Methods for Discrete Structures” (GRK 1408).

1. Introduction

We study the computational complexity of control by adding voters [2, 31], investigating the case where the sets of voters that we can add have some combinatorial structure. The problem of election control by adding voters models situations where some agent (for example, a campaign manager of one of the alternatives) tries to ensure a given alternative’s victory by convincing some undecided voters to vote. Traditionally, in this problem we are given a description of an election (that is, a set C of alternatives and a set V of voters who already decided to vote), and also a set W of undecided voters. For each voter in $V \cup W$ we assume that we know how this voter intends to vote, which is expressed as a linear order over the set C ; while this assumption is somewhat unrealistic, it is a standard assumption within computational social choice, and we might have a good approximation of this knowledge from preelection polls. Our goal is to ensure that our preferred alternative p becomes a winner by convincing as few voters from W as possible to vote—provided that it is at all possible to ensure p ’s victory in this way.

Control by adding voters corresponds, for example, to situations where supporters of a given alternative make direct appeals to other supporters of the alternative to vote. For example, they may stress the importance of voting or help with the voting process by offering rides to the voting locations. Unfortunately, in its traditional phrasing, control by adding voters does not model larger-scale attempts at convincing people to vote. For example, a campaign manager might be interested in airing a TV advertisement that would motivate supporters of a given alternative to vote (though, of course, it might also motivate some of this alternative’s enemies), or maybe launch viral campaigns, where friends convince their own friends to vote. It is clear that the sets of voters that we can add should have some sort of a combinatorial structure. For instance, a TV advertisement appeals to a particular group of voters and we can add all of them at the unit cost of airing the advertisement. A public speech in a given neighborhood will convince a particular group of people to vote at the unit cost of organizing the meeting. Convincing a person to vote will “for free” also convince her friends to vote.

The goal of our work is to formally define an appropriate computational problem modeling a combinatorial variant of control by adding voters and to study its computational complexity. We focus on the Plurality rule and

the Condorcet rule, and we do so for the following reasons. First, these are the rules originally studied by Bartholdi et al. [2] in the first paper on the complexity of election control. Second, the Plurality rule is the most widely used rule in practice and the Condorcet rule models a large family of Condorcet-consistent rules. Third, the Plurality rule is one of the few rules for which the standard variant of control by adding voters is solvable in polynomial time [2]. For the Condorcet rule the problem is NP-hard in general [2], but becomes polynomial-time solvable if we assume that the elections have a particular structure (for example, if they are either single-peaked [24] or single-crossing [37]). For the case of single-peaked elections, in essence, all our hardness results for the Condorcet rule directly translate to all Condorcet-consistent voting rules, a large and important family of voting rules. We defer the formal details, definitions, and concrete results to the following sections. Instead, we state the high-level main messages of our work. Herein, we assume that adding an unregistered voter means adding a *bundle* (subset) of unregistered voters; in this way, it is easy to see that the standard variant of control by adding voters is a special case of the combinatorial variant (set the bundle of each unregistered voter to be a singleton consisting of this single voter):

1. Many typical variants of combinatorial control by adding voters are intractable, but there is also a rich landscape of tractable cases. For instance, with bundle sizes up to two, the problem is either fixed-parameter tractable with respect to the number k of bundles to add or already polynomial-time solvable when requiring the bundling function to be full- d (see Section 2 for the definition; informally, this means that only voters with roughly the same preference orders can be bundled together).
2. Assuming that voters have single-peaked preferences does not lower the complexity of the problem (even though it does so in many other election problems [7, 14, 24]). On the contrary, assuming single-crossing preferences does lower the complexity of the problem.

We believe that our setting of combinatorial control, and—more generally—of combinatorial problems that model manipulating elections, offers a very fertile ground for future research and we intend the current paper as an initial step.

Related Work. Bartholdi et al. [2] were the first to study the concept of election control by adding/deleting voters/alternatives in a given election.

They considered the constructive variant of the problem, where the goal is to ensure a given alternative’s victory (and we focus on this variant of the problem as well). The destructive variant, where the goal is to prevent someone from winning, was introduced by Hemaspaandra et al. [31]. These papers focused on the Plurality rule and the Condorcet rule (for the destructive case of Hemaspaandra et al. [31], also the Approval rule). Since then, many other researchers extended this study to a number of other rules and models [39, 22, 23, 45, 25, 43].

We study our control problems using the tools and methods of parameterized complexity theory. Most frequently, parameterized complexity of control problems is studied with respect to the number of alternatives as the parameter [22, 23, 32]. The number of voters received far less attention as a parameter (for the case of control, the parameter appears, for example, in the works of Betzler and Uhlmann [3] and, very recently, of Chen et al. [13]; Brandt et al. [8] consider it in the context of winner determination). Several authors have also considered other parameters, such as the solution size (for example, the number of voters one can add). Papers focusing on this parameter include, for example, those of Liu et al. [36], Liu and Zhu [35], and Erdélyi et al. [20].

Some of our results regard the complexity of election control for the case where the voters’ preference orders are either single-peaked [5] or single-crossing [41, 44] (intuitively, both these domain restrictions model cases where there is a linear spectrum of opinions; single-peakedness assumes that it is the alternatives that are ordered from one extreme to the other within this spectrum, for example, the left-to-right political spectrum, while single-crossingness assumes that the order is over the voters and their opinions). For both types of domain restrictions there are algorithms that can recognize elections with a given property (see, for example, the works of Bartholdi and Trick [1] and Escoffier et al. [21] for the single-peaked domain and those of Elkind et al. [18] and Bredereck et al. [9] for the single-crossing domain). The complexity of control for single-peaked elections was studied by Faliszewski et al. [24] and was continued, for example, by Brandt et al. [7] and Faliszewski et al. [26]. The case of control in single-crossing elections was considered by Magiera and Faliszewski [37]. Generally speaking, the complexity of control problems often drops from NP-completeness to being polynomial-time solvable when one of these domain restrictions is assumed. Naturally, single-peakedness and single-crossingness were studied algorithmically in many other contexts as well. Perhaps the first authors which observed that assuming them may lower the complexity of election problems were Walsh [50] and Conitzer [14].

In all previous work on election control, the authors always assumed that one could affect each entity of the election at unit cost only.⁵ For example, one could add a voter at a unit cost and adding two voters always is twice as expensive as adding a single voter. Only the paper of Faliszewski et al. [25], where the authors study control in weighted elections, could be seen as an exception: one could think of adding a voter of weight w as adding a group of w voters, each with unit weight. On the one hand, the weighted election model does not allow one to express rich combinatorial structures as those that we study here, while on the other hand, in our study we consider unweighted elections only (though adding weights to our model would be seamless). Very recently, Chen et al. [13] studied the combinatorial variant of both constructive as well as destructive control by either adding or deleting alternatives. They discovered that with few voters, the complexity of the corresponding control problem for different voting rules ranges from polynomial-time solvable to NP-hard even for a constant number of voters. Erdélyi et al. [19] also studied a variant of combinatorial control by adding, deleting, or partitioning of the voters. They used a slightly different—though also very natural—model of bundling voters, where each voter has a label and each bundle consists exactly of the voters with a given label. Formally, our models are incomparable and, indeed, we show hardness results for the case of combinatorial control by adding voters with bundles of size two, whereas in their model this case is easily seen to be polynomial-time solvable.

The specific combinatorial flavor of our model is inspired by the seminal work of Rothkopf et al. [46]⁶ on *combinatorial auctions* (see, for example, the work of Sandholm [47] for additional information). There, bidders can place bids on combinations of items such that the bid on the combination of a set of items might be less than, equal to, or greater than the sum of the individual bids on each element from the same set of items. While in combinatorial auctions one “bundles” items to bid on, in our scenario one bundles voters.

In the computational social choice literature, combinatorial voting is typically associated with scenarios where voters express opinions over a set of alternatives that themselves have a specific combinatorial structure (typically, one uses CP-nets to model preferences over such sets of alternatives [6]). For example, Conitzer et al. [15] studied a form of control in this setting and

⁵By “previous work,” we mean papers that were published prior to the conference version of our paper.

⁶According to Google Scholar, accessed February 2015, cited more than 1000 times.

Mattei et al. [38] studied bribery problems. In contrast, we use the standard model of elections where all alternatives and preference orders are given explicitly, but we have a combinatorial structure on the sets of voters which can be added.

Paper Outline. In Section 2, we introduce notations used throughout the whole paper and concepts necessary for the parameterized complexity analysis. In Section 3, we formally define our central problem and summarize our contributions. Then, we go on to study the complexity of combinatorial voter control problems where voters may have arbitrary preference orders: in Section 4, we focus on the so-called canonical parameters, specifically, the solution size, the number of alternatives, and the number of unregistered voters, while in Section 5 we focus on parameters arising from the combinatorial structure, that is, the maximum bundle size and the swap distance. In Section 6, we analyze situations where the voters' preference orders are either single-peaked or single-crossing. We conclude in Section 7 with several future research directions.

2. Preliminaries

We assume familiarity with standard notions regarding algorithms and complexity theory. For each nonnegative integer z , we write $[z]$ to mean $\{1, \dots, z\}$.

Elections. An election $E := (C, V)$ consists of a set C of m alternatives and a set V of voters $v_1, v_2, \dots, v_{|V|}$. Each voter v has a linear order \succ_v over the set C , which we call a *preference order* (a *vote*). For example, let $C = \{c_1, c_2, c_3\}$ be a set of alternatives. The preference order $c_1 \succ_v c_2 \succ_v c_3$ of voter v indicates that v likes c_1 best (1st position), then c_2 , and c_3 least (3rd position). We call a voter $v \in V$ a *c-voter* if c is at the first position of v 's preference order. Given a subset $C' \subseteq C$ of alternatives, if not stated explicitly, we write $\langle C' \rangle$ to denote an arbitrary but fixed preference order over C' .

Voting Rules. A voting rule \mathcal{R} is a function that, given an election $E = (C, V)$, outputs a (possibly empty) set $\mathcal{R}(E) \subseteq C$ of the (tied) election winners. We study the Plurality rule and the Condorcet rule. Given an election, the *Plurality score* of an alternative c is the number of voters having c at the first position in their preference orders; an alternative is a Plurality winner if it has the highest Plurality score. An alternative c is a *Condorcet winner* [16] if it beats all other alternatives in head-to-head contests, that

is, if for each alternative $c' \in C \setminus \{c\}$ it holds that $|\{v \in V \mid c \succ_v c'\}| > |\{v \in V \mid c' \succ_v c\}|$. Condorcet's rule elects the (unique) Condorcet winner if it exists, and returns an empty set otherwise. A voting rule is *Condorcet-consistent* if it elects a Condorcet winner when there is one. However, if there is no Condorcet winner, then a Condorcet-consistent rule is free to provide any set of winners.

Domain Restrictions. Intuitively, an election is *single-peaked* [5] if it is possible to order the alternatives on a line in such a way that for each voter v whose preferred alternative is c , the following holds: for each two alternatives c_i and c_j such that both of them are on the same side of c (with respect to the ordering of the alternatives on the line), among c_i and c_j , v prefers the one closer to c . For example, single-peaked elections arise when we view the alternatives as positioned on the standard political left-right spectrum and voters form their preferences based solely on the alternatives' positions on this spectrum. Formally, we have the following definition.

Definition 1 (Single-peaked elections). Let C be a set of alternatives and let L be a linear order over C (referred to as the societal axis). We say that a preference order \succ (over C) is *single-peaked* with respect to L if for each three alternatives $c, c', c'' \in C$ it holds that:

$$((c L c' L c'') \vee (c'' L c' L c)) \implies ((c \succ c') \implies (c' \succ c'')).$$

An election (C, V) is single-peaked with respect to L if the preference order of each voter in V is single-peaked with respect to L . An election is single-peaked if there is a societal axis with respect to which it is single-peaked.

There are polynomial-time algorithms that decide whether a given election is single-peaked and, if so, provide a societal axis for it [1, 21].

Single-crossing elections, introduced by Mirrlees [41] and Roberts [44], are based on similar idea as single-peaked ones, but from a different perspective. Here, it is assumed that it is possible to order the voters such that for each two alternatives x and y either all voters rank x and y identically, or there is a single point along this order where voters switch from preferring one of the alternatives to preferring the other one (see the work of Saporiti and Tohmé [48] for a number of real-life examples where single-crossing elections may appear and for further references regarding this domain restriction). Formally, we have the following definition.

Definition 2 (Single-crossing elections). An election (C, V) is *single-crossing* if there is an order L over V such that for each two alternatives c

and c' and each three voters x, y, z with $x L y L z$ it holds that:

$$(c \succ_x c' \wedge c \succ_z c') \implies c \succ_y c'.$$

Just as for the case of single-peakedness, there are polynomial-time algorithms that decide whether an election is single-crossing and, if so, produce the voter order witnessing this fact [18, 9].

Combinatorial Bundling Functions. Given a voter set W , a combinatorial bundling function $\kappa : W \rightarrow 2^W$ (abbreviated as *bundling function*) is a function assigning to each voter a subset of voters. For convenience, for each subset $W' \subseteq W$, we let $\kappa(W') = \bigcup_{x \in W'} \kappa(x)$. For $x \in W$, $\kappa(x)$ is called x 's *bundle* and x is called the *leader* of this bundle. We assume that $x \in \kappa(x)$ so that $\kappa(x)$ is never empty. We typically write b to denote the maximum bundle size under a given κ (which will always be clear from context). Intuitively, we use combinatorial bundling functions to describe the sets of voters that we can add to an election at a unit cost. For example, one can think of $\kappa(x)$ as the group of voters that join the election under x 's influence. Bundling functions can be represented explicitly: for each voter x , simply list the voters in $\kappa(x)$.

It is worthwhile to shortly discuss the model here. We could have defined our problem differently, by having sets of voters as the bundles, without a distinguished leader (somewhat similarly to the model of Erdélyi et al. [19], but with each voter having possibly many labels). We chose our approach based on the idea that upon convincing a single voter to attend the election, his or her friends would likely follow. We mention that most of our results transfer to this other model as well.

We are interested in various special cases of bundling functions. We say that κ is *leader-anonymous* if for each two voters x and y with the same preference orders, $\kappa(x) = \kappa(y)$ holds. Furthermore, κ is *follower-anonymous* if for each two voters x and y with the same preference orders, and each voter z , it holds that $x \in \kappa(z)$ if and only if $y \in \kappa(z)$. We call κ *anonymous* if it is both leader-anonymous and follower-anonymous. One possible way of thinking about an anonymous bundling function is that it is a function assigning to each preference order appearing in the input a subset of the preference orders appearing in the input. For example, anonymous bundling functions naturally model scenarios such as airing TV advertisements that appeal to particular groups of voters.

The *swap distance* between two voters v_i and v_j is the minimum number of swaps of consecutive alternatives that transform v_i 's preference order into that of v_j . Given a number $d \in \mathbb{N}$, we call κ a *full- d bundling function* if

for each $x \in W$, $\kappa(x)$ is exactly the set of all $y \in W$ such that the swap distance between the preference orders of x and y is at most d . The idea is that here, only voters whose preference orders are roughly the same can be bundled together.

We introduce the concept of the *bundling graph* of an election, which, roughly speaking, models how the bundles of two voters interact with each other.

Definition 3 (Bundling graphs). Given a bundling function κ (over the set W of voters), the *bundling graph* is a simple and directed graph $G = (V(G), E(G))$. For each voter x there is a vertex $u_x \in V(G)$, and for each two distinct voters z and y such that $y \in \kappa(z)$, there is an arc $(u_z, u_y) \in E(G)$.

We use the classic definition of *connectivity* and *connected components* for directed graphs. That is, G is called (*weakly*) *connected* if the underlying undirected graph is connected. Accordingly, a *connected component* of G is a maximal connected subgraph of G .

Given a bundling graph and an alternative c , we say that a vertex is a *c-vertex* if the corresponding voter is a c -voter; otherwise we call it a *non-c-vertex*. Accordingly, we say that an arc is a (c_1, c_2) -arc if the source of this arc is a c_1 -vertex and the target is a c_2 -vertex.

For a non-negative integer x , an *x-star* is a (directed) graph consisting of $(x + 1)$ vertices and x arcs such that there is a vertex with x (in- or out-) neighbors.

For arbitrary bundling functions, the bundling graph is a directed graph. However, if κ is a *full-d bundling function*, that is, for each voter v , $\kappa(v)$ contains all the voters with swap distance at most d , then we can consider it as an undirected one. The reason is that in this case for every two unregistered voters x and y we have that $y \in \kappa(x)$ if and only if $x \in \kappa(y)$. In consequence, for each arc (u_x, u_y) in the bundling graph, the reverse arc (u_y, u_x) is also present.

Observation 1. *If κ is a full-d bundling function, then for any unregistered voter x and any $y \in \kappa(x)$, it holds that $x \in \kappa(y)$.*

Proof. Note that for any two voters x and y if $y \in \kappa(x)$, then the swap distance between x and y is at most d . Therefore, since κ is a full-d bundling function and the swap distance is clearly symmetric, x must be in $\kappa(y)$. \square

Note that the “mutual containment” property, as stated above, does not hold for every bundling function. For example, κ with $\kappa(x) = \{x, y\}$ and

$\kappa(y) = \{y\}$ is a valid bundling function. The following is easy to observe, as full- d bundling functions depend only on the preference orders and not on the specific voters:

Observation 2. *If κ is a full- d bundling function, then κ is also anonymous.*

Proof. To show that κ is anonymous we need to show both leader-anonymity and follower-anonymity. Suppose that κ is a full- d bundling function. Let x, y, z be three voters such that x and y have the same preference order. If $z \in \kappa(x)$ (respectively, if $x \in \kappa(z)$), then z has a swap distance of at most d to x , and hence, to y . By the definition of full- d bundling functions, $z \in \kappa(y)$ (respectively, $y \in \kappa(z)$). This shows the leader-anonymity (respectively, follower-anonymity) of κ . \square

Parameterized Complexity. An instance (I, r) of a *parameterized problem* consists of the actual instance I and of an integer r referred to as the *parameter* [17, 27, 42]. A parameterized problem is called *fixed-parameter tractable* (is in FPT) if there is an algorithm solving it in time $f(r) \cdot |I|^{O(1)}$, where f is a computable function depending on the parameter r only. An algorithm with running-time $|I|^{f(r)}$ shows membership in the class XP (clearly, $\text{FPT} \subseteq \text{XP}$).

If a parameterized problem is fixed-parameter tractable due to a formulation as an integer linear program (ILP), then we say that this problem is in ILP-FPT. Of course, ILP-FPT is not a separate complexity class; we use this notation because ILP-FPT problems are solved using the famous algorithm of Lenstra [34], which means that while they are in FPT, the algorithms for them might be not very practical and might not reveal any structural properties of the problems.

One can show that a parameterized problem L is (presumably) not fixed-parameter tractable by devising a *parameterized reduction* from a W[1]-hard or a W[2]-hard problem to L . A parameterized reduction from a parameterized problem L to another parameterized problem L' is a function that acts as follows, for some two computable functions f and g : given an instance (I, r) , it computes in $f(r) \cdot |I|^{O(1)}$ time an instance (I', r') such that $r' \leq g(r)$ and $(I, r) \in L \Leftrightarrow (I', r') \in L'$. Betzler et al. [4] survey parameterized complexity investigations in voting.

In this paper, we use the following three problems parameterized by the “solution size” to show parameterized intractability results: CLIQUE, PARTIAL VERTEX COVER, and SET COVER. The first two problems are W[1]-complete while the last one is W[2]-complete [17]. We give the formal

definition of CLIQUE in the proof of [Theorem 2](#), PARTIAL VERTEX COVER in the proof of [Theorem 9](#), and SET COVER in the proof of [Theorem 1](#).

3. Central Problem

For a given voting rule \mathcal{R} , we define our central problem of combinatorial constructive control by adding voters as follows:

\mathcal{R} COMBINATORIAL CONSTRUCTIVE CONTROL BY ADDING VOTERS (\mathcal{R} -C-CC-AV)

Input: An election $E = (C, V)$, a set W of (unregistered) voters with $V \cap W = \emptyset$, a bundling function $\kappa : W \rightarrow 2^W$, a preferred alternative $p \in C$, and a non-negative integer bound $k \in \mathbb{N}$.

Question: Is there a subset of voters $W' \subseteq W$ of size at most k such that $p \in \mathcal{R}(C, V \cup \kappa(W'))$?

We use the so-called nonunique-winner model. That is, for a control action to be successful it suffices for p to be one of the tied winners. Throughout this work, we refer to each subset $W' \subseteq W$ of voters such that p wins the election $(C, V \cup \kappa(W'))$ and $|W'| \leq k$ as a *solution*, and we refer to k as the *solution size* (formally, k is a bound on the allowed solution size, but our notation makes the discussion a bit simpler). For the Plurality rule, we also assume that the score difference between the current winner and p does not exceed the total number of p -voters in W .

\mathcal{R} -C-CC-AV is a generalization of the well-studied problem \mathcal{R} CONSTRUCTIVE CONTROL BY ADDING VOTERS (\mathcal{R} -CC-AV) (in which, effectively, κ is fixed so that for each $w \in W$ we have $\kappa(w) = \{w\}$). The non-combinatorial problem CC-AV is linear-time solvable for the Plurality rule by a simple calculation [2] (see also the remark at the beginning of [Section 5](#)), but is NP-complete for the Condorcet rule [36]. Therefore we have the following observations:

Observation 3. *If the maximum bundle size b is one, then PLURALITY-C-CC-AV is solvable in linear time.*

Observation 4. *CONDORCET-C-CC-AV is NP-hard even if the maximum bundle size b is one.*

Our Contributions. We introduce a new model for combinatorial control in voting. Our results show that C-CC-AV is NP-hard even for the Plurality rule. For this reason, we complement our study by focusing on a number of

Bundling function κ	Unrestricted domain	Restricted domain
Arbitrary	$\text{XP} : O(n^k) \cdot (m + n')^{O(1)}$ [Prop. 1] W[2]-hard wrt. k [Thm. 1] (already when $m = 2$) FPT wrt. k when $b \leq 2$ [Thm. 5]	
Anonymous	ILP-FPT wrt. m [Thm. 3]	
	NP-hard [Thm. 4] (already when $b = 2$) W[1]-hard wrt. k [Thm. 2] (already when $b = 3$)	?
Full- d (in effect, anonymous [Obs. 2])	P when $b \leq 2$ [Thm. 6] NP-hard [Thm. 7] (already when $b = 3$ and $d = 3$) NP-hard [Thm. 8] (already when $b = 4$ and $d = 1$)	Single-peaked: W[1]-hard wrt. k [Thm. 9] Single-crossing: P [Thm. 10]

Table 1: Computational complexity classification of PLURALITY-C-CC-AV (since the non-combinatorial problem CC-AV is already NP-hard for Condorcet’s rule, we concentrate on the Plurality rule here). The parameters that we study here are “the number m of alternatives”, “the number n' of registered voters”, “the number n of unregistered voters”, “the solution size k ”, “the maximum bundle size b ”, and “the maximum swap distance d between the leader and its followers in a bundle”. We distinguish between unrestricted and restricted domains (the left and the right column), and between arbitrary, anonymous, and full- d bundling functions (respectively, the first, the second, and the third row). ILP-FPT means FPT based on a formulation as an integer linear program. The question mark (?) means that it is open whether the tractability result for full- d bundling functions also holds for non-full bundling functions and whether the hardness results for unrestricted domains transfer to single-peaked or single-crossing domains.

different parameters, showing both fixed-parameter tractability results and parameterized hardness results. We almost completely resolve the complexity of C-CC-AV for the Plurality rule and the Condorcet rule as a function of the maximum bundle size b and the maximum distance d from a voter v to the farthest element of v ’s bundle. For example, for Plurality voting, the

complexity of the problem depends on b in the following way:

- (1) If $b = 1$, then the problem is polynomial-time solvable (this is due to Bartholdi et al. [2]; see [Observation 3](#)).
- (2) If $b = 2$, then the complexity of the problem depends on the bundling function. If the bundling function is full- d , then the problem is polynomial-time solvable ([Theorem 6](#)). Otherwise, the problem is NP-hard ([Theorem 4](#)), but is in FPT with respect to the solution size ([Theorem 5](#)).
- (3) If $b = 3$, then the problem is W[1]-hard even for anonymous bundling functions ([Theorem 2](#)), and is NP-hard for full- d bundling functions, even if $d \leq 3$ ([Theorem 7](#)).
- (4) For any constant $b \geq 4$, the problem is NP-hard already for full- d bundling functions with $d = 1$ ([Theorem 8](#)); for $d = 0$, which in essence means looking at the weighted control case for the special case of unary-encoded weights, the problem is polynomial-time solvable [25].

For the Condorcet rule, we obtain NP-hardness even when the input has only two alternatives (of course, this result applies to the Plurality rule as well; for two alternatives the two rules are identical).

Furthermore, we show that for both the Plurality rule and the Condorcet rule C-CC-AV remains hard even when restricting the elections to be single-peaked, but that it is polynomial-time solvable when we focus on single-crossing elections. Our results for Plurality elections are summarized in [Table 1](#).

We make a final remark that the combinatorial variants of voter control problems that we study are clearly contained in NP. Thus, our NP-hardness results in fact imply NP-completeness results.

4. Canonical Parameterizations

In this section we provide our results for unrestricted elections, i.e., for the case where voters may have arbitrary preference orders. Later, in [Section 6](#), we will consider single-peaked and single-crossing elections that only allow “reasonably restricted” preference orders.

4.1. Parameterization by the Solution Size and by the Number of Unregistered Voters

We start our discussion by considering the parameters “number $|W|$ of unregistered voters” and “solution size k ”. A simple brute-force algorithm, checking all possible combinations of k bundles, proves that both PLURALITY-C-CC-AV and CONDORCET-C-CC-AV are in XP for parameter k , and in FPT for parameter $|W|$ (the latter holds because $k \leq |W|$). Indeed, the same result holds for all voting rules with polynomial-time winner-determination procedures.

Proposition 1. PLURALITY-C-CC-AV is solvable in time $O(|W|^k \cdot (|V| + |W|))$ and CONDORCET-C-CC-AV is solvable in time $O(|W|^k \cdot (|V| + |W|) \cdot |C|^2)$, implying that both for PLURALITY and CONDORCET, CC-AV is in XP for parameter k and in FPT for parameter $|W|$.

Proof. We can solve C-CC-AV by considering all elections resulting from adding one of the $\sum_{j=0}^k \binom{|W|}{j} \leq O(|W|^k)$ possible combinations of (up to) k bundles of unregistered voters. For each combination of (up to) k bundles of voters, we use the standard winner determination algorithm for the given voting rule. For PLURALITY, the winner can be computed in time $O(|V| + |W|)$; for CONDORCET, the winner can be computed in time $O((|V| + |W|) \cdot |C|^2)$: for each pair of alternatives c and c' , we compute whether a strict majority of voters prefers c to c' or c' to c (we can do so on a voter-by-voter basis by storing the results for each pair of alternatives). \square

The XP result for PLURALITY-C-CC-AV parameterized by the solution size k probably cannot be improved to fixed-parameter tractability. Indeed, for parameter k we show that the problem is W[2]-hard, even for elections with only two alternatives. This is quite remarkable because typically election problems with a small number of alternatives are easy (they can be solved either by brute-force or by integer linear programming employing the famous FPT algorithm of Lenstra [34]; see the survey of Betzler et al. [4] for examples). Nonetheless, there are other known examples of problems where a small number of alternatives does not seem to lower the complexity of a given election problem [10]. Furthermore, since our proof uses only two alternatives, it applies to almost all natural voting rules: for two alternatives almost all of them (including the Condorcet rule) are equivalent to the Plurality rule. Also, every election with two alternatives is trivially single-peaked and single-crossing, thus the next result extends to these domain restrictions as well.

Theorem 1. PLURALITY-C-CC-AV is NP and W[2]-hard when parameterized by the solution size k are W[2]-hard, even for two alternatives.

Proof. We first show the W[2]-hardness result by providing a parameterized reduction from the W[2]-complete problem DOMINATING SET [17], defined as follows (we take h to be the parameter):

DOMINATING SET

Input: An undirected graph $G = (V(G), E(G))$ and a non-negative integer $h \in \mathbb{N}$.

Question: Does G admit a *dominating set* of size at most h , that is, a vertex subset $U \subseteq V(G)$ with $|U| \leq h$ such that each vertex from $V(G) \setminus U$ is adjacent to at least one vertex from U ?

Let (G, h) be a DOMINATING SET instance. We construct an election (C, V) as follows. We let the set of alternatives be $C = \{p, g\}$, where p is our preferred alternative. Since our election has only two alternatives, when we speak of a p -voter (a g -voter), we mean a voter with preference order $p \succ g$ (respectively, preference order $g \succ p$). The registered voter set V consists of $|V(G)|$ g -voters (and no p -voters). The unregistered voter set W consists of one p -voter w_i for each vertex $u_i \in V(G)$. We let the bundle $\kappa(w_i)$ of w_i consist of the p -voters corresponding to the closed neighborhood of u_i . Formally, we define $W := \{w_i \mid u_i \in V(G)\}$ and $\kappa(w_i) = \{w_i\} \cup \{w_j \mid \{u_i, u_j\} \in E(G)\}$. Observe that all unregistered voters are p -voters. In order to let p win, we have to add bundles to the election that correspond to the whole vertex set. Finally, we set $k := h$.

It is clear that our construction is a polynomial reduction and hence, a parameterized reduction. It remains to show that there is a dominating set of size at most h if and only if there is a subset W' of unregistered voters of size at most k , such that if their respective bundles are added to the election, then p becomes a Plurality winner of the election.

For the “if” part, suppose that there is a subset W' of size at most k such that p is a winner of the Plurality election $(C, V \cup \kappa(W'))$. Define U to be the set of vertices corresponding to the voters from W' , that is, $U := \{u_i \mid w_i \in W'\}$. Then, it is easy to verify that $|U| \leq k = h$ and for each vertex $u_i \in V(G) \setminus U$ there must be a vertex $u_j \in U$ which is adjacent to u_i , since otherwise p will not obtain enough points to become a winner.

For the “only if” part, given a dominating set U of size at most h , we define W' to be the corresponding voter set, that is, $W' := \{w_i \mid u_i \in U\}$. It is easy to verify that $|W'| \leq h = k$ and p as well as g win with n points both.

As already mentioned, the parameterized reduction we presented is indeed a polynomial-time reduction. Since the problem we reduce from is NP-hard we can conclude that our problem is NP-hard even for two alternatives. \square

Since with two alternatives the Condorcet rule is equivalent to the strict majority rule (that is, one needs to add enough p -voters such that p is preferred to the second alternative by more than half of the voters) we can adapt the above reduction for the Plurality rule to also work for the Condorcet rule. Thus, the following holds.

Proposition 2. *CONDORCET-C-CC-AV is $W[2]$ -hard when parameterized by the solution size k , even for two alternatives.*

Proof. We use the same unregistered voters as described in the proof for [Theorem 1](#), and we construct the original election with $(n - 1)$ g -voters. The correctness proof follows in an analogous way. \square

If we require the bundling functions to be anonymous and the maximum bundle size to be three, then PLURALITY-C-CC-AV turns out to be $W[1]$ -hard (regarding the same parameter k).

Theorem 2. *PLURALITY-C-CC-AV is $W[1]$ -hard when parameterized by the solution size, even when the maximum bundle size b is three and the bundling function is anonymous.*

Proof. To show the $W[1]$ -hardness result, we provide a parameterized reduction from the $W[1]$ -hard problem CLIQUE [17], defined as follows (we take h as the parameter):

CLIQUE

Input: An undirected graph $G = (V(G), E(G))$ and $h \in \mathbb{N}$.

Question: Does G admit a size- h clique, that is, a size- h vertex subset $U \subseteq V(G)$ such that $G[U]$ is complete?

Let (G, h) be a CLIQUE instance. Without loss of generality, we assume that G is connected, $h \geq 3$, and each vertex in G has degree at least $h - 1$. We construct an election $E = (C, V)$ with $C := \{p, f, g\} \cup \{c_e \mid e \in E(G)\}$ with p as our preferred alternative, and the intention of f being the current winner, and g having sufficiently many unregistered supporters to ensure that we indeed add a “clique solution” to the election. We use the edge alternatives from $\{c_e \mid e \in E(G)\}$ to ensure that all the unregistered voters have different preference orders.

We introduce registered voters such that initially, f wins with $\binom{h}{2} + h$ points, g has $\binom{h}{2}$ points, our preferred alternative p has h points, and all edge alternatives have zero points. Formally, the registered voter set V consists of the following groups of voters:

- (1) h voters, each with preference order $p \succ \langle C \setminus \{p\} \rangle$,
- (2) $\binom{h}{2} + h$ voters, each with preference order $f \succ \langle C \setminus \{f\} \rangle$, and
- (3) $\binom{h}{2}$ voters, each with preference order $g \succ \langle C \setminus \{g\} \rangle$.

In this way, we enforce that p needs at least $\binom{h}{2}$ points to become a winner. By carefully constructing the preference orders of the unregistered voters, we can enforce that the added voters correspond to a clique of size h . To this end, for each vertex $u \in V(G)$, we define $C(u) := \{c_e \mid e \in E(G) \wedge u \in e\}$, that is, $C(u)$ contains all edge alternatives that correspond to the incident edges of u . Now, we construct the set W of unregistered voters as follows:

- (1) For each vertex $u \in V(G)$, we add an unregistered g -voter w_u with preference order

$$g \succ \langle C(u) \rangle \succ p \succ \langle C \setminus (\{g, p\} \cup C(u)) \rangle.$$

We call these unregistered voters *vertex voters*. We set $\kappa(w_u) = \{w_u\}$.

- (2) For each edge $e = \{u, u'\} \in E(G)$, we add an unregistered p -voter w_e with preference order

$$p \succ c_e \succ g \succ \langle C \setminus \{p, g, c_e\} \rangle.$$

We call these unregistered voters *edge voters*. We set $\kappa(w_e) = \{w_u, w_{u'}, w_e\}$.

Note that all unregistered voters have different preference orders. This implies that our bundling function κ is anonymous (when all the unregistered voters have different preference orders, then every bundling function is anonymous). To complete our construction, we set $k := \binom{h}{2}$.

We show that G has a size- h clique if and only if $((C, V), W, \kappa, p, k)$ is a yes-instance for PLURALITY-C-CC-AV. For the “if” part, suppose that there is a subset W' of at most k voters such that p wins the election $(C, V \cup \kappa(W'))$. We show that the vertex set $U' := \{u \in V(G) \mid w_e \in W' \wedge u \in e\}$ is a size- h clique for G . First, we observe that p needs at least $\binom{h}{2}$ points to become a winner because of the difference in scores between the initial

winner f and p . By our construction, only bundles which include an edge voter can increase the score of p by adding one p -voter, while adding another two g -voters. Since we can add at most $k = \binom{h}{2}$ bundles, we must add exactly k bundles of the edge voters. This means that $E(G[U'])$ contains k edges. However, in order to ensure p 's victory, $\kappa(W')$ may only give at most h additional points to g . By the construction of the bundles of the edge voters, this means that U' contains at most h vertices. With $|E(G[U'])| \geq k$, we conclude that U' is of size h and, hence, is a size- h clique for G .

For the “only if” part, suppose that $U' \subseteq V(G)$ is a size- h clique for G . We construct the subset W' by adding to it any edge voter w_e with $e \in E(G[U'])$. Obviously, $|W'| = k$. Now it is easy to check that p co-wins with both f and g the election $(C, V \cup \kappa(W'))$ with score $\binom{h}{2} + h$. \square

4.2. Parameterization by the Number of Alternatives

From [Theorem 1](#), we know that our central problem for both the Plurality rule and the Condorcet rule is NP-hard already when the input election has only two alternatives. The corresponding proof uses the non-anonymity of the bundling function in a crucial way. Indeed, if we require the bundling function to be anonymous, then C-CC-AV can be formulated as an integer linear program (ILP) where the number of variables and the number of constraints are bounded by some function dependent only on the number m of alternatives. Finding feasible solutions for such integer linear programs is in FPT, parameterized by the number of variables, due to the famous result of Lenstra [\[34\]](#).

Theorem 3. *For Plurality and Condorcet, when parameterized by the number m of alternatives, C-CC-AV is NP-hard already for two alternatives, while it is fixed-parameter tractable for anonymous bundling functions.*

Proof. The NP-hardness result with two alternatives follows from [Theorem 1](#). Thus, we only need to show the fixed-parameter tractability result for anonymous bundling functions. We prove this by describing an integer linear program (ILP) with at most $m!$ variables and at most $3m! + m$ constraints which solves PLURALITY-C-CC-AV, where m denotes the number of alternatives. Fixed-parameter tractability then follows because every ILP with ρ variables and L input bits is solvable in time $O(\rho^{2.5\rho + o(\rho)} L)$ [\[34, 33\]](#). The case of CONDORCET-C-CC-AV follows by a nearly identical argument; we mention the necessary modifications at the end of the proof.

In a given election with m alternatives, there are at most $m!$ voters with pairwise different preference orders. Since the bundling function is

anonymous and, hence, follower-anonymous, there are at most $m!$ different bundles. Furthermore, we can assume that all voters in a solution W' have pairwise different preference orders (this is because, due to (leader) anonymity, there is no additional gain in adding two voters with the same preference order).

We introduce some notation for the description of our ILP: Let $\succ_1, \succ_2, \dots, \succ_{m!}$ be a sequence of all the possible preference orders over m alternatives. For $i \in [m!]$, let N_i be the number of voters with preference order \succ_i in W . For each alternative $a \in C$, we write $F(a)$ to denote the set of preference orders in which a is ranked first, and write $s(a)$ to denote a 's initial score.

For each preference order \succ_i , $i \in [m!]$, we introduce two boolean variables, x_i and y_i . The intended meaning of $x_i = 1$ is that the sought solution W' contains a voter with preference order \succ_i . The intended meaning of $y_i = 1$ is that $\kappa(W')$ contains voters with preference order \succ_i . In our ILP, we use the values of the variables x_i to enforce the correct values of the variables y_i . We abuse our notation slightly and for each preference order \succ_i we write $\kappa(\succ_i)$ to denote the set of preference orders of the voters included in the bundle of the voters with preference order \succ_i . For each preference order \succ_j we define $\kappa^{-1}(\succ_j) = \{\succ_i \mid \succ_j \in \kappa(\succ_i)\}$ to be the set of preference orders that include \succ_j in their bundles.

Now we are ready to state our integer linear program (note that it suffices to find a feasible solution and, thus, we do not specify any function to minimize):

$$\sum_{i \in [m!]} x_i \leq k, \quad (1)$$

$$x_i \leq N_i \quad \forall i \in [m!], \quad (2)$$

$$\sum_{\succ_i \in \kappa^{-1}(\succ_j)} x_i \leq m! \cdot y_j \quad \forall j \in [m!], \quad (3)$$

$$\sum_{\succ_i \in \kappa^{-1}(\succ_j)} x_i \geq y_j \quad \forall j \in [m!], \quad (4)$$

$$s(p) + \sum_{\succ_j \in F(p)} N_j \cdot y_j \geq s(a) + \sum_{\succ_j \in F(a)} N_j \cdot y_j \quad \forall a \in C \setminus \{p\}. \quad (5)$$

This ILP requires some comments. First, constraint (1) ensures that at most k voters are added to W' and constraint (2) ensures that the voters added to W' are indeed present in W . Constraints (3) and (4) ensure that

variables y_j , $1 \leq j \leq m!$, have correct values. Indeed, if for some preference order \succ_i we have $x_i = 1$ and $\succ_j \in \kappa(\succ_i)$, then constraint (3) ensures that $y_j = 1$. On the other hand, if for some preference order \succ_j we have that for each preference order \succ_i with $\succ_j \in \kappa(\succ_i)$ it holds that $x_i = 0$, then constraint (4) ensures that $y_j = 0$. Finally, constraint (5) ensures that p has Plurality score at least as high as every other alternative (and, thus, is a winner). Clearly, there is a solution for this integer linear program if and only if there is a solution for the input instance.

For the case of the Condorcet rule, we modify constraint (5). Instead of comparing alternatives' Plurality scores, we formulate it to compare how many voters prefer p to each given alternative a . Designing such a constraint is an easy exercise. \square

5. Parameterization by the Maximum Bundle Size and by the Swap Distance

We now focus on the complexity of PLURALITY-C-CC-AV as a function of two combinatorial parameters: (a) the maximum size b of each voter's bundle, and (b) the maximum swap distance d between the leader and her followers in one bundle. (By Observation 4, for the Condorcet rule the problem is NP-hard already for $b = 1$, so we do not consider the Condorcet rule in this section.)

Specifically, we show that Plurality-C-CC-AV is polynomial-time solvable if the maximum bundle size is one (that is, if we are in the non-combinatorial setting already studied by Bartholdi et al. [2]), but it is NP-hard already when the maximum bundle size is two and the bundling function is anonymous. We also show that when the maximum bundle size is two, the problem for arbitrary bundling functions, parameterized by the solution size, is in FPT. In contrast, if κ is a full- d bundling function (that is, if each bundle contains all the voters at swap distance at most d from the leader), then Plurality-C-CC-AV is polynomial-time solvable if the maximum bundle size is two, but is NP-hard already when the maximum bundle size is three.

5.1. Bundle Size At Most Two—The Intractability Result

First, if $b = 1$, as mentioned in Observation 3, then C-CC-AV reduces to CC-AV and, thus, can be solved in linear time [2]. Indeed, one only needs to calculate the score difference of the preferred alternative and the current winner and check whether k , the maximum number of voters one

may add, is at least as large as this difference and whether there are enough p -voters from the unregistered voter set to add to the election.

PLURALITY-C-CC-AV becomes intractable as soon as the maximum bundle size b is two, even for anonymous bundling functions. To show this we reduce from the following restricted variant of 3SAT.

(2-2)-3SAT

Input: A collection \mathcal{C} of size-two-or-three clauses over the variable set $\mathcal{X} = \{x_1, \dots, x_n\}$, such that each clause has either two or three literals, and each variable appears exactly four times, twice as a positive literal and twice as a negative literal.

Question: Is there a truth assignment that satisfies all the clauses in \mathcal{C} ?

This variant remains NP-hard.

Lemma 1. (2-2)-3SAT is NP-complete.

Proof. Clearly, the problem belongs to NP. We provide a reduction from the NP-complete 3SAT, where each clause has either *two* or *three* literals, each variable occurs either two or three times, and at most one time as a negative literal [49, Theorem 2.1].

First, we assume that no variable appears only positively: if this were the case for some variable, then we could set it to true and simplify the formula. For each variable x_i that appears three times (two times positively and one time negatively), we add one new variable y_i , and two new clauses $\{\neg x_i, \neg y_i, y_i\}$ and $\{\neg y_i, y_i\}$. For each variable x_i that appears two times (one time positively and one time negatively), we add one new clause $\{\neg x_i, x_i\}$. It is easy to see that the original instance is a yes-instance if and only if the newly constructed instance is a yes-instance for (2-2)-3SAT. \square

Theorem 4. PLURALITY-C-CC-AV is NP-hard even if the bundling function is anonymous and the maximum bundle size b is two.

Proof. We reduce from the NP-complete problem (2-2)-3SAT (Lemma 1). Given a (2-2)-3SAT instance $(\mathcal{C}, \mathcal{X})$, where \mathcal{C} is the set of clauses over the set of variables \mathcal{X} , we construct an election (C, V) . We set $k := 4|\mathcal{X}|$, and define the set C of alternatives to be $C := \{p, g\} \cup \{c_i \mid C_i \in \mathcal{C}\} \cup \{d_j^{(1)}, d_j^{(2)}, d_j^{(3)}, d_j^{(4)} \mid x_j \in \mathcal{X}\}$. We refer to the alternatives c_i as the *clause alternatives*, and to the alternatives $d_j^{(z)}$ as the *dummy alternatives*. We use the clause alternatives to make sure that the solution to PLURALITY-C-CC-AV encodes a satisfying truth assignment and we use the dummy

alternatives to ensure that all unregistered voters have distinct preference orders (this ensures that our bundling function is anonymous).

We construct the set V of registered voters so that the initial score of g is $4|\mathcal{X}|$, the initial score of each clause alternative c_i is $4|\mathcal{X}| - |C_i| + 1$ (where $|C_i|$ is the number of literals that clause C_i contains), and the initial score of p is zero. We assume without loss of generality that no clause contains the same literal more than once.

We construct the set W of unregistered voters as follows (throughout the rest of the proof, we will often write ℓ_j to refer to a literal that contains variable x_j ; depending on the context, ℓ_j will mean either x_j or $\neg x_j$ and the exact meaning will always be clear). For each variable $x_j \in \mathcal{X}$ that occurs as a negative literal ($\neg x_j$) in clauses C_i and C_s , $i < s$, and as a positive literal (x_j) in clauses C_r and C_t , $r < t$, we construct:

- (1) Four p -voters, denoted by $p_j^{(1)}, p_j^{(2)}, p_j^{(3)}, p_j^{(4)}$, with the following preference orders:

$$\begin{aligned} p_j^{(1)} &: p \succ d_j^{(1)} \succ \langle C \setminus \{p, d_j^{(2)}\} \rangle, \\ p_j^{(2)} &: p \succ d_j^{(2)} \succ \langle C \setminus \{p, d_j^{(3)}\} \rangle, \\ p_j^{(3)} &: p \succ d_j^{(3)} \succ \langle C \setminus \{p, d_j^{(4)}\} \rangle, \\ p_j^{(4)} &: p \succ d_j^{(4)} \succ \langle C \setminus \{p, d_j^{(1)}\} \rangle; \end{aligned}$$

we call these voters *variable voters*.

- (2) Four *clause voters*, denoted by $c_i^{\neg x_j}, c_r^{x_j}, c_s^{\neg x_j}, c_t^{x_j}$, with the following preference orders:

$$\begin{aligned} c_i^{\neg x_j} &: c_i \succ d_j^{(1)} \succ \langle C \setminus \{c_i, d_j^{(1)}\} \rangle, \\ c_r^{x_j} &: c_r \succ d_j^{(2)} \succ \langle C \setminus \{c_s, d_j^{(2)}\} \rangle, \\ c_s^{\neg x_j} &: c_s \succ d_j^{(3)} \succ \langle C \setminus \{c_r, d_j^{(3)}\} \rangle, \\ c_t^{x_j} &: c_t \succ d_j^{(4)} \succ \langle C \setminus \{c_t, d_j^{(4)}\} \rangle. \end{aligned}$$

Note that each clause C_i has exactly $|C_i|$ corresponding clause voters.

We now describe our bundling function κ . Intuitively, κ is such that the ensuing bundling graph (see [Definition 3](#)) contains a cycle for each variable

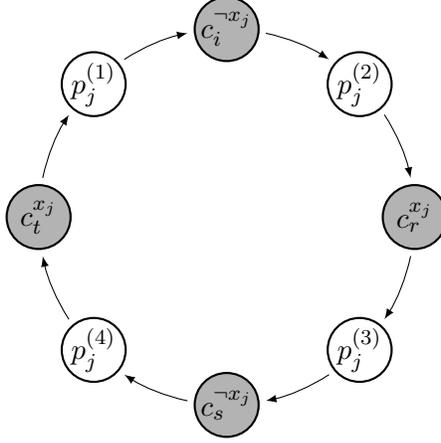


Figure 1: Part of the construction used in [Theorem 4](#). Specifically, we show the cycle corresponding to variable x_j which occurs as a negative literal in clauses C_i and C_s , and as a positive literal in clauses C_r and C_t .

(see [Figure 1](#) for an illustration). Formally, we define κ as follows:

$$\begin{aligned}
\kappa(p_j^{(1)}) &:= \{p_j^{(1)}, c_i^{\neg x_j}\}, & \kappa(c_i^{\neg x_j}) &:= \{c_i^{\neg x_j}, p_j^{(2)}\}, \\
\kappa(p_j^{(2)}) &:= \{p_j^{(2)}, c_r^{x_j}\}, & \kappa(c_r^{x_j}) &:= \{c_r^{x_j}, p_j^{(3)}\}, \\
\kappa(p_j^{(3)}) &:= \{p_j^{(3)}, c_s^{\neg x_j}\}, & \kappa(c_s^{\neg x_j}) &:= \{c_s^{\neg x_j}, p_j^{(4)}\}, \\
\kappa(p_j^{(4)}) &:= \{p_j^{(4)}, c_t^{x_j}\}, & \kappa(c_t^{x_j}) &:= \{c_t^{x_j}, p_j^{(1)}\}.
\end{aligned}$$

This completes the construction. It runs in polynomial time.

To show the correctness of the construction, the general idea is that in order to let p win, all p -voters must be in $\kappa(W')$ and no clause alternative c_i should gain more than $(|C_i| - 1)$ points. More formally, we now show that $(\mathcal{C}, \mathcal{X})$ has a satisfying truth assignment if and only if there is a size- k voter subset $W' \subseteq W$ such that p wins the election $(\mathcal{C}, V \cup \kappa(W'))$ (recall that $k = 4|\mathcal{X}|$).

For the “if” direction, let $\beta : \mathcal{X} \rightarrow \{T, F\}$ be a satisfying truth assignment function for $(\mathcal{C}, \mathcal{X})$. Intuitively, β will guide us through constructing the voter set W' in the following way: First, for each variable x_j , we put into W' those clause voters $c_i^{\ell_j}$ for whom β sets ℓ_j to false. This way in $\kappa(W')$ we include $2|\mathcal{X}|$ p -voters and, for each clause c_i , at most $(|C_i| - 1)$ c_i -voters. The former is true because exactly $|\mathcal{X}|$ literals are set to false by β , each literal is included in exactly two clauses, and adding each $c_i^{\ell_j}$ into W' also includes a unique p -voter into $\kappa(W')$; the latter is true because if β is a

satisfying truth assignment then each clause C_i contains at most $(|C_i| - 1)$ literals set to false. Then, for each clause voter $c_i^{\ell_j}$ already in W' , we also add the voter $p_j^{(z)}$, $1 \leq z \leq 4$, that contains $c_i^{\ell_j}$ in her bundle. This way we include in $\kappa(W')$ $2|\mathcal{X}|$ additional p -voters without increasing the number of clause voters included. Formally, we define W' as follows:

$$W' := \{c_i^{\neg x_j}, p_j^{(z)} \mid \beta(x_j) = T \wedge \neg x_j \in C_i \wedge c_i^{\neg x_j} \in \kappa(p_j^{(z)})\} \cup \\ \{c_i^{x_j}, p_j^{(z)} \mid \beta(x_j) = F \wedge x_j \in C_i \wedge c_i^{x_j} \in \kappa(p_j^{(z)})\}.$$

As per our intuitive argument, one can verify that all p -voters are contained in $\kappa(W')$ and each clause alternative c_i gains at most $(|C_i| - 1)$ points.

For the “only if” part, let W' be a subset of voters such that p wins the election $(C, V \cup \kappa(W'))$.

We say that a literal ℓ_j is *selected* if at least one voter $c_i^{\ell_j}$ is *not* in $\kappa(W')$. If for some variable x_j all four voters $c_i^{\ell_j}$ are in $\kappa(W')$, then we arbitrarily set literal x_j to be selected. Intuitively, selecting a literal means that it should be set to true to make the formula satisfied. More precisely, in the same way as all voters $c_i^{\ell_j}$ cannot be together in $\kappa(W')$ for any alternative c_i , all literals in clause C_i cannot be set to false together.

Before we continue with defining the truth assignment, we first prove that for each variable x_j , literals x_j and $\neg x_j$ cannot be both selected. This is clear in the special case where all four $c_i^{\ell_j}$ are in $\kappa(W')$. Now we observe that for each two clauses that contain the same variable but not the same literal, at least one corresponding clause voter must be added to the election (otherwise $\kappa(W')$ would not contain all unregistered p -voters). Thus, if one clause voter is not contained in $\kappa(W')$, then both of her “neighboring” (in the sense of being adjacent in the bundling graph, depicted in [Figure 1](#)) clause voters must be included in $\kappa(W')$. This means that for each variable x_j , $\kappa(W')$ must contain at least the two voters of the form c^{x_j} or the two voters of the form $c^{\neg x_j}$, which means in turn that only one of x_j and $\neg x_j$ is selected. Although this is not required for our proof, we further remark that the solution could be edited so that *only* voters corresponding to the unselected literals are in $\kappa(W')$.

We now define the truth assignment $\beta : \mathcal{X} \rightarrow \{T, F\}$ such that $\beta(x_j) := T$ if literal x_j is selected and $\beta(x_j) := F$ if $\neg x_j$ is selected. Following the previous arguments, function β is well-defined. It is a satisfying truth assignment function for (C, \mathcal{X}) because for each clause C_i , by the fact that p is a winner in election $(C, V \cup \kappa(W'))$, we have that $\kappa(W')$ contains at most $(|C_i| - 1)$ c_i -voters for each clause alternative c_i . There must be some

$\ell_j \in C_i$ such that $c_i^{\ell_j} \notin \kappa(W')$, hence ℓ_j is selected. Thus, this literal is set to true via β , and clause C_i is satisfied.

Overall, the formula is satisfiable if and only if it is possible to make p win by adding $2|\mathcal{X}|$ bundles, which completes the reduction. \square

5.2. Bundle Size At Most Two—The Tractability Results

While we have just shown that PLURALITY-C-CC-AV is NP-hard even if each bundle has at most two voters, intuitively it is plausible that some tractability should stand out for this case, as this setting is very restrictive. We justify this intuition by showing the following results.

- (1) We give an FPT algorithm for the problem, parameterized by the solution size k (Theorem 5).
- (2) We give a polynomial-time algorithm for the case where we restrict the bundling function to be full- d (Theorem 6).

Both results rely on the fact that, for $b = 2$, we can work with the corresponding bundling graph in the following way. Intuitively, we should select arcs containing as many p -vertices as possible. Hence, we first find a maximum matching among arcs whose both endpoints are p -vertices. We then update the bundles (and the corresponding bundling graph) and add to the solution all p -voters whose bundles are singletons. Finally, in the non-full- d case, we brute-force search through the bundling graph structure corresponding to the remaining part of the solution in FPT-time. In the full- d case, we can solve the remaining problem greedily in polynomial time.

To implement our ideas, we first need some notions and observations concerning the structure of the bundling graphs in our instances. Throughout the remainder of the discussion of the $b = 2$ case, let $I = ((C, V), W, \kappa, p, k)$ be a PLURALITY-C-CC-AV instance, and let $G = (V(G), E(G))$ be the bundling graph for I . We assume that this is a yes-instance and we let W' be a solution of size up to k for I . We say that a solution is minimal if it has the smallest size among all solutions, and we focus on such solutions.

Since $b = 2$, each bundle corresponds to one of the following four different bundle types:

- (1) bundles consisting of two p -voters, the corresponding bundling graph notion for this type is a (p, p) -arc;
- (2) bundles consisting of exactly one p -voter, the corresponding bundling graph notion for this type is a p -vertex with no outgoing arcs;

- (3) bundles consisting of one p -voter and one non- p -voter, the corresponding bundling graph notion for this type is either a $(p, \text{non-}p)$ -arc or a $(\text{non-}p, p)$ -arc; and
- (4) bundles not containing p -voters.

Clearly, we never need to include bundles of the last type into our solution. Thus, we only need to take care of the bundles of the first three types and, without loss of generality, we assume that G does not contain any arc between two non- p -voters. Further, we can consider the first three bundle types independently, in the same order as they are listed above. The next two lemmas formalize this observation.

Lemma 2. *If W' contains a c -voter w' with $c \in C$ (possibly $p = c$) whose bundle $\kappa(w')$ is not of type (1) but there is at least one p -voter $w \in W \setminus \kappa(W')$ such that $\kappa(w)$ is of type (1), then $W'' := (W' \setminus \{w'\}) \cup \{w\}$ is also a solution of the same size.*

Proof. Let $E' := (C, V \cup \kappa(W'))$ and let $E'' := (C, V \cup \kappa(W''))$. Clearly, $|W'| = |W''|$. Furthermore, p gets at least as many points in election E'' as in election E' and every other alternative gets at most the same score in E'' as in E' . \square

Lemma 3. *If W' contains a p -voter w' such that $\kappa(w')$ is of type (3) but there is at least one p -voter $w \in W \setminus \kappa(W')$ such that $\kappa(w)$ is of type (2), then $W'' := (W' \setminus \{w'\}) \cup \{w\}$ is also a solution of the same size.*

Proof. Let $E' := (C, V \cup \kappa(W'))$ and let $E'' := (C, V \cup \kappa(W''))$. Observe that $|W'| = |W''|$ and p gets the same score in both election E' and election E'' . Furthermore, every other alternative gets at most the same score in E' as in E'' . \square

We need the following lemma to preprocess trivial yes-instances and to upper-bound (by k) the number of p -vertices adjacent to a given vertex.

Lemma 4. *Suppose that all the bundles are of type (3). If G contains (as a subgraph) a k -star whose center is a c -vertex for some $c \in C \setminus \{p\}$ and whose leaves are p -vertices, then the following holds.*

1. *If c has at least k points more than p , then no solution can contain the leader of any bundle corresponding to an arc of the star.*
2. *Otherwise, provided that p can become a winner at all, this star corresponds to a solution of size k .*

Proof. Suppose that c has at least k points more than p . Since all bundles are of type (3), adding at most k bundles increases the score of p by at most k points. Adding even one bundle corresponding to an arc from the star increases the score of c by one, making the score difference between p and c to be at least k . Thus, if it is only allowed to add at most k bundles, then adding a bundle corresponding to an arc from the star makes it impossible for p to be a Plurality winner.

In contrast, if c has at most $k - 1$ points more than p , then the best we can do is to add all leaders of the k bundles corresponding to the k arcs of the star. This increases the score of p by k (the highest possible value given that all bundles are of type (3)) and keep the score of c not greater than that of p . If it is possible to ensure p 's victory, then adding these k bundles ensures p 's victory. \square

Now, we are ready to state and prove the two theorems relying on these observations.

Theorem 5. *If the maximum bundle size b is two, then PLURALITY-C-CC-AV parameterized by the solution size k can be solved in $O(m \cdot n^2 + k^{4k} \cdot m \cdot n)$ time.*

Proof. Following the discussion preceding the proof, our algorithm starts by picking as many disjoint bundles of types (1) and (2) as possible. To do so, we first find a maximum matching for the corresponding bundling graph restricted to the (p, p) -arcs, and we add the bundles corresponding to this matching (we also update the bundling function to take this into account; indeed, some bundles may change type). Then, we add as many bundles with exactly one p -voter as possible (that is, bundles of type (2)). By Lemmas 2 and 3, this greedy approach is correct. From now on, we assume that our bundles are only of type (3) (and we assume that our value k is modified, taking into account all the bundles we have added so far).

We now make several observations regarding our bundling graph G . We assume that it is possible to ensure p 's victory, and we consider a specific minimum-size solution, denoted by W' . We let G' be the subgraph of G that corresponds to the solution W' . Formally, $G' = (V(G'), E(G'))$, where $V(G') = V(G)$, and

$$E(G') := \{(u, v) \in E(G) \mid u \text{ corresponds to a leader } x \text{ in } W' \text{ such that } v \text{ corresponds to a follower in } \kappa(x)\}.$$

Claim 1. *For each p -vertex in G' , it holds that the sum of its in-degree and its out-degree is at most one.*

Proof of Claim 1. Towards a contradiction, assume that there is a p -vertex u for which the sum of its in-degree and its out-degree is at least two. Let u_1 and u_2 be two neighbors of u (note that at least one of them is an in-neighbor). Since all the bundles are of type (3), both u_1 and u_2 are non- p -vertices. Let $v_1, v_2 \in W$ be the voters to which u_1 and u_2 correspond. At least one of them must be a leader (that is, must belong to W'). It is easy to verify that removing this voter from W' results in a correct solution of smaller size, which contradicts the assumption that W' is minimal. (of Claim 1) \diamond

By the above claim, we conclude that each (weakly) connected component of G' is a star.

Claim 2. *Each (weakly) connected component of G' has the following properties:*

1. *It is a star.*
2. *The center of the star corresponds to a non- p -voter and all leaves to p -voters.*
3. *The center of the star has at most one out-arc and at most k in-arcs.*

Proof of Claim 2. Let F be a (weakly) connected component of G' . We first show that F contains at most one non- p -vertex. Towards a contradiction, suppose that F contains two non- p -vertices, u and u' . These two vertices cannot be adjacent because we only have bundles of type (3). Thus, there is a path between u and u' that contains at least one p -vertex (we ignore the direction of the edges on this path). However, the sum of the in-degree and the out-degree of a p -vertex on such a path would be two, which by Claim 1 is impossible. Thus, F contains at most one non- p -vertex. Since for every p -vertex the sum of its in-degree and its out-degree is at most one (Claim 1), F is a star and we can take the p -vertices to be the leaves. Hence, we take the non- p -vertex as the center (if we only had two vertices, one p -vertex and one non- p -vertex, it would be possible to consider the p -vertex as the center as well, but we do not do so). The last part of our claim follows by Lemma 4 and because every bundle has size two. (of Claim 2) \diamond

Based on the above claims, we derive a search-tree algorithm for the case where we are left with bundles of type (3) only. That is, the bundling graph corresponding to a minimum solution consists only of stars where the centers are non- p -voters. We start with an empty graph and we keep adding

to it “good” stars, one by one. To formalize this idea, we need one more notion: Given two graphs, G_1 and G_2 , such that G_2 is a subgraph of G_1 , by $G_1 \setminus G_2$ we mean the directed graph obtained from G_1 by deleting all arcs from G_2 followed by deleting all isolated vertices.

Now, we are ready to describe our algorithm. First, we guess the size k' ($0 \leq k' \leq k$) of a minimum-size solution (this value equals the additional score that p will gain). Then, we begin with an empty graph M and a budget r , initialized to k' . We will repeatedly add “good” stars to the graph M . Thus, when we speak of the j th star, we mean the j th star that is, or will be, added to M . During the algorithm, we occasionally mark some of the p -vertices as *reserved* for some of the stars that are yet to be added to M . Similarly, we occasionally mark some alternatives as *needed* by some of the to-be-added stars. Intuitively, if a p -vertex is reserved for a star, it means that this star uses this vertex as a leaf. When an alternative is needed by a star, it means that this star’s center must correspond to this alternative.

The main part of our algorithm is to execute the following two steps in a loop, until the whole budget is used up. Initially M is empty and the budget r is set to k' . Let i be the number of stars in M added so far, plus one (so, initially, $i = 1$).

- (1) Find a star S from $G \setminus M$ that satisfies the following constraints (let the center of S be a c -vertex, $c \neq p$):
 - (a) S contains all the p -vertices that have been reserved for the i th star but no p -vertices that have been reserved for any i' th star with $i' > i$. If some alternative c' is marked as needed for the i th star, then the constraint $c' = c$ must hold.
 - (b) The original score of c plus the number of occurrences of the c -vertices in M plus the number of times that c is needed for some stars i' , $i' > i$, is less than the original score of p plus k' , and
 - (c) S has the largest—but not larger than $(r - t)$ —number of leaves among all stars $G \setminus G'$ fulfilling the first two conditions, where t is the number of p -vertices that are reserved for a star j , $j > i$.

If such a star does not exist, then we backtrack to the last step where the $(i - 1)$ th star has not yet been added to M (because the graph constructed so far cannot be extended to a graph corresponding to a size- k' solution).

- (2) There are three possibilities regarding the relation between S and the bundling graph G'' of a size- k solution, where G'' contains M and respects the current reservation and the current center requirements; we guess which one of them actually applies:
- (A) G'' contains S . Let S be the i th star and add it to M , decrease the budget r by the number of arcs in S .
 - (B) There is a value j , $j > i$, such that so far we did not mark any alternative as needed by the j th star. Guess the value of j and mark alternative c as needed by the j th star. If the number of stars that need an alternative exceeds the budget, then we backtrack to the beginning of Step (2) and branch into other possibilities.
 - (C) There is a value j , $j > i$, such that the j th star will use some *non-reserved* vertex v from S . Guess the value of j and guess the vertex v from S . Reserve v for the j th star. If all p -vertices from S have been reserved, then backtrack to the beginning of Step (2) and branch into other possibilities.

We show the correctness of this algorithm by the following inductive argument. Suppose that the graph M computed so far is correct, that is, there is a bundling graph G' that satisfies the following.

- (a) G' contains M ,
- (b) G' corresponds to a size- k' solution,
- (c) G' respects the current reservation requirements, that is, every reserved p -vertex is contained in G' , no star in G' contains two vertices that are reserved for different stars, and if two p -vertices are reserved for the same star j , then they are also contained in the same star of G' , and
- (d) G' respects the current center requirements, that is, for each alternative $c \neq p$, it holds that the number of times that c is needed is at most the number of c -vertices in G' .

Now, we show that there is a bundling graph G''' satisfying the four conditions above such that one of the three branchings in Step (2) applies to G''' . Obviously, if G' contains the center of S , then we can verify that there is a bundling graph G''' satisfying the conditions (a)-(d) and containing S (Branching (2A)). Now assume that G' does not contain the center of S .

There are three cases regarding the intersection between G' and S (we use the following notation: if a star has some c -vertex as the center, $c \neq p$, we say that this is a c -centered star):

1. Suppose that the bundling graph G' does not contain any vertex from the star S . Let S_1 be a c -centered star in $G' \setminus M$ if $G' \setminus M$ contains one. Otherwise, let S_1 be an empty graph.
 - (a) If no p -vertex in S_1 is reserved or if S_1 is empty, then let S_2 be a supergraph of S_1 and a subgraph of $G' \setminus M$ such that S_2 consists of exactly s p -vertices that are not reserved for any star j with $j > i$ (note that such subgraph exists because of the constraint (1c)). One can verify that the bundling graph $G'' := (G' \setminus S_2) \cup S$ satisfies the conditions (a)-(d) and contains S . Thus, branching (2A) applies to G'' .
 - (b) Otherwise, S_1 contains a reserved p -vertex u . By assumption, G' does not contain any vertex of S . Therefore, from constraint (1a), we know that no p -vertex in G' is reserved for the i th star. This implies that u is reserved for a star j with $j > i$ and that we can guess one value $j > i$ and mark the alternative c as needed for the star j . Thus, branching (2B) applies to G' .
2. Suppose that the graph $G' \setminus M$ contains two stars S_1 and S_2 , $S_1 \neq S_2$, such that each star S_ℓ , $\ell \in \{1, 2\}$, contains a vertex v_ℓ from S .
 - (a) If v_1 (resp. if v_2) is reserved, then v_1 (resp. if v_2) is reserved for the i th star (constraint (1a)). This implies that v_2 (resp. v_1) is not reserved. Then, we can guess one value j , $j > i$, and guess one not yet reserved vertex v from S , and mark v as reserved for the j th star.
 - (b) Otherwise, both v_1 as well as v_2 are not reserved. Since S_1 and S_2 cannot be the i th star at the same time, we can guess one value j , $j > i$ and guess one not yet reserved vertex v from S , and mark v as reserved for the j th star.

In both cases, branching (2C) applies to G' .

3. Suppose that the graph $G' \setminus M$ contains exactly one star S_1 which contains a p -vertex u from S . Let the center of S_1 be a c_1 -vertex.
 - (a) If $c = c_1$ and if no p -vertex in S_1 is reserved for a star j with $j > i$, then let S_2 be a supergraph of S_1 and a subgraph of $G' \setminus M$

such that S_2 consists of exactly s p -vertices that are not reserved for star j with $j > i$ (note that such subgraph exists because of constraint (1c)). One can verify that the bundling graph $G'' := (G' \setminus S_2) \cup S$ satisfies conditions (a)-(d) and contains S . Thus, branching (2A) applies to G'' .

- (b) If S_1 has a p -vertex which is reserved for some star j' , $j' > i$, then this vertex cannot be u because of constraint (1a). Thus, we can guess one value j , $j > i$, and guess one not yet reserved vertex v from S , and mark v as reserved for the j th star. This implies that branching (2C) applies to G' .
- (c) If $c \neq c'$ and if no p -vertex in S_1 is reserved, then by constraint (1c) we know that S_1 has at most s arcs.
 - i. If the original score of c plus the number of occurrences of the c -vertices in G' is less than the original score of p plus k' , then let S_2 be a supergraph of S_1 and a subgraph of $G' \setminus M$ such S_2 consists of exactly s p -vertices that are not reserved for star j with $j > i$ (note that such subgraph exists because S satisfies constraint (1c)). One can verify that the bundling graph $G'' := (G' \setminus S_2) \cup S$ satisfies the conditions (a)-(d) and contains S ; note that G'' and G' have the same number of p -vertices because S_1 is the only star in G' that has a p -vertex from S , and that the score of c will not exceed the final score of p because of constraint (1b). Thus, branching (2A) applies to G'' .
 - ii. Otherwise, $G' \setminus M$ contains at least one c -centered star S_2 due to constraint (1b). Since S_1 and S_2 cannot be the i th star simultaneously, we can guess one value j with $j > i$ and guess one not yet reserved vertex v from S , and mark v as reserved for the j th star, or guess one value j with $j > i$ and mark c as needed for the j th star. Thus, either branching (2C) or branching (2B) hold for G' .

We have shown the correctness of our algorithm. To see the running time, note that constructing the bundling graph for a given instance runs in time $O(m \cdot n^2)$. The constructed graph has at most n vertices and at most n arcs. Preprocessing bundles of type (1) by finding a maximum matching runs in time $O(n^{3/2})$ [40]. Preprocessing bundles of type (2) runs in $O(n)$ time. For the running time of the search-tree algorithm, note that S has at most k arcs and there are at most $k - 1$ additional stars that need an

alternative. This leads to a total of $1 + k \cdot (k - 1) + (k - 1) = k^2$ possible guesses in Step (2). Moreover, after guessing the minimum solution size k' , we build a search-tree algorithm which has depth at most $2k$ (the number of stars in M plus the number of stars that need an alternative is at most k while the number of reserved p -vertices is at most k) and has branching factor k^2 . This means that our search tree has size at most $(k^2)^{2k}$. In each branching node we can find in $O(m \cdot n)$ time an appropriate star S fulfilling the constraints given in step (1). Thus, the combined running time is $O(m \cdot n^2 + k^{4k} \cdot m \cdot n)$. \square

If we require the bundling function to be full- d , then we obtain a polynomial-time algorithm by extending the greedy algorithm of Bartholdi et al. [2].

Theorem 6. *If κ is a full- d bundling function and the maximum bundle size b is two, then PLURALITY-C-CC-AV is solvable in $O((|V| + |W|) \cdot |C|)$ time.*

Proof. As in Theorem 5, we first select as many p -vertices as possible without any non- p vertex. Since in our PLURALITY-C-CC-AV instance $((C, V), W, \kappa, p, k)$ every bundle has at most two voters and κ is a full- d bundling function, any two bundles are either equal or disjoint. Thus, by Lemmas 2 and 3, we can greedily select all (disjoint) bundles of type (1) and then all (disjoint) bundles of type (2) (note that by selecting a bundle we mean to add the leader of the bundle to the solution; since both voters corresponding to a bundle are leaders of the same bundle, we can choose between them arbitrarily).

By Lemma 2 and Lemma 3 and due to all bundles being either equal or disjoint, this greedy approach is correct. Thus, from now on, we assume that our bundles are only of type (3).

The algorithm continues by sorting (in ascending order) the remaining bundles by the score of the non- p -voter in each bundle (remember that we have only bundles of type (3)) and by adding these bundles with respect to this ordering.

The correctness of the algorithm is easy to verify.

For the running time, note that calculating the scores of all alternatives costs $O(|V| \cdot |C|)$ time. Adding bundles with two p -voters or with one p -voter and throwing away all irrelevant bundles costs $O(|W| \cdot |C|)$. Sorting the bundles can be done in $O(|W| \cdot |C|)$ time as the values are upper-bounded in the instance size. Then, looping over the sorted alternatives, for each bundle possibly shifting the current alternative higher in the sorted sequence, can

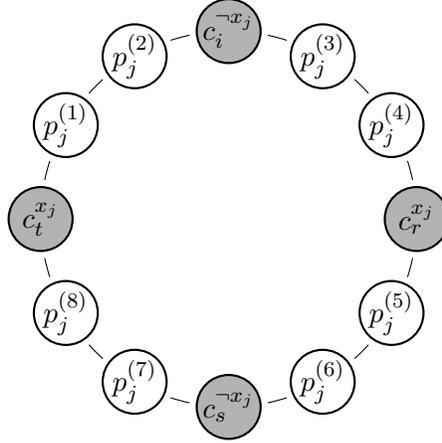


Figure 2: Part of the construction used in [Theorem 7](#). Specifically, we show the cycle corresponding to variable x_j which occurs as a negative literal in clauses C_i and C_s , and as a positive literal in clauses C_r and C_t .

be done in $O(|W| \cdot |C|)$ time. Thus, the total running time is $O((|V| + |W|) \cdot |C|)$. \square

5.3. Bundle Size Three or More

For $b = 3$, we obtain NP-hardness even for full- d bundling functions with $d \leq 3$. The proof is similar to the one used in the proof of [Theorem 4](#).

Theorem 7. PLURALITY-C-CC-AV is NP-hard even for full- d bundling functions with constant value $d \geq 3$ and maximum bundle size $b \leq 3$.

Proof. We give a reduction from (2-2)-3SAT. This reduction is almost the same as the one given in the proof of [Theorem 4](#). The main difference is that we carefully construct the voters' preference orders so that the bundling function is full-3 and each bundle consists of at most three voters. This leads to some technical differences, but the main idea of the construction remains the same.

Let $(\mathcal{C}, \mathcal{X})$ be our input instance of (2-2)-3SAT, where \mathcal{C} is the set of clauses over the variables from the set \mathcal{X} . We form the same set of alternatives as in the proof of [Theorem 4](#), i.e., we set $C := \{p, g\} \cup \{c_i \mid C_i \in \mathcal{C}\} \cup \{d_j^{(1)}, d_j^{(2)}, d_j^{(3)}, d_j^{(4)} \mid x_j \in \mathcal{X}\}$. As in the proof of [Theorem 4](#), we construct the set of registered voters so that the initial score of alternative g is $8|\mathcal{X}|$, the initial score of each clause alternative c_i is $8|\mathcal{X}| - |C_i| + 1$ (where $|C_i|$ is the number of literals in C_i), and the initial scores of all the other alternatives are zero.

We construct the set of unregistered voters as follows. For each variable $x_j \in \mathcal{X}$ that occurs as a negative literal ($\neg x_j$) in some clauses C_i and C_s , $i < s$, and as a positive literal (x_j) in some clauses C_r and C_t , $r < t$, we introduce eight unregistered p -voters with preference orders:⁷

$$\begin{aligned}
p_j^{(1)} &: p \succ c_t \succ c_i \succ c_r \succ c_s \succ d_j^{(1)} \succ d_j^{(2)} \succ d_j^{(3)} \succ d_j^{(4)} \succ \dots, \\
p_j^{(2)} &: p \succ c_i \succ c_r \succ c_s \succ c_t \succ d_j^{(1)} \succ d_j^{(2)} \succ d_j^{(3)} \succ d_j^{(4)} \succ \dots, \\
p_j^{(3)} &: p \succ c_i \succ c_r \succ c_s \succ c_t \succ d_j^{(2)} \succ d_j^{(4)} \succ d_j^{(3)} \succ d_j^{(1)} \succ \dots, \\
p_j^{(4)} &: p \succ c_r \succ c_s \succ c_t \succ c_i \succ d_j^{(2)} \succ d_j^{(4)} \succ d_j^{(3)} \succ d_j^{(1)} \succ \dots, \\
p_j^{(5)} &: p \succ c_r \succ c_s \succ c_t \succ c_i \succ d_j^{(3)} \succ d_j^{(4)} \succ d_j^{(1)} \succ d_j^{(2)} \succ \dots, \\
p_j^{(6)} &: p \succ c_s \succ c_t \succ c_i \succ c_r \succ d_j^{(3)} \succ d_j^{(4)} \succ d_j^{(1)} \succ d_j^{(2)} \succ \dots, \\
p_j^{(7)} &: p \succ c_s \succ c_t \succ c_i \succ c_r \succ d_j^{(4)} \succ d_j^{(2)} \succ d_j^{(1)} \succ d_j^{(3)} \succ \dots, \\
p_j^{(8)} &: p \succ c_t \succ c_i \succ c_r \succ c_s \succ d_j^{(4)} \succ d_j^{(2)} \succ d_j^{(1)} \succ d_j^{(3)} \succ \dots.
\end{aligned}$$

One can verify that, for each integer $z \in [4]$, the following holds.

- (1) The swap distance between the two p -voters $p_j^{(2z-1)}$ and $p_j^{(2z)}$ is exactly three.
- (2) For each $z' \in [8] \setminus \{2z-1, 2z\}$, the swap distances between $p_j^{(2z-1)}$ and $p_j^{(z')}$, and between $p_j^{(2z)}$ and $p_j^{(z')}$, are at least four.

For each variable x_j , we also introduce four unregistered clause voters with the following preference orders:

$$\begin{aligned}
c_i^{\neg x_j} &: c_i \succ p \succ c_r \succ c_s \succ c_t \succ d_j^{(2)} \succ d_j^{(1)} \succ d_j^{(4)} \succ d_j^{(3)} \succ \dots, \\
c_r^{x_j} &: c_r \succ p \succ c_s \succ c_t \succ c_i \succ d_j^{(4)} \succ d_j^{(3)} \succ d_j^{(2)} \succ d_j^{(1)} \succ \dots, \\
c_s^{\neg x_j} &: c_s \succ p \succ c_t \succ c_i \succ c_r \succ d_j^{(4)} \succ d_j^{(1)} \succ d_j^{(3)} \succ d_j^{(2)} \succ \dots, \\
c_t^{x_j} &: c_t \succ p \succ c_i \succ c_r \succ c_s \succ d_j^{(1)} \succ d_j^{(4)} \succ d_j^{(2)} \succ d_j^{(3)} \succ \dots.
\end{aligned}$$

One can verify that, for each c_y -voter $c_y^{\ell_j}$, it holds that:

⁷We assume that those alternatives that we do not list explicitly in the preference orders are ranked identically by these voters.

- (A) The swap distance between $c_y^{\ell_j}$ and the p -voter $p_j^{(z)}$ that ranks c_y at the second place is exactly three (one swap between p and c_i and two swaps among alternatives $d_j^{(1)}, d_j^{(2)}, d_j^{(3)}$, and $d_j^{(4)}$).
- (B) The swap distance between $c_y^{\ell_j}$ and another clause voter is at least four (because of the clause alternatives c_i, c_r, c_s , and c_t , and p).
- (C) The swap distance between $c_y^{\ell_j}$ and every p -voter $p_j^{(z)}$, $z \in [8]$ that ranks c_y below the second place is at least four (because of the clause alternatives c_i, c_r, c_s , and c_t , and p).

Note that the swap distance between two voters that correspond to two different variables is much larger than three. We define our bundling function to be a full-3 bundling function. In effect, for variable x_j , we obtain the following values of the bundling function (also depicted in [Figure 2](#)):

$$\begin{aligned}
\kappa(p_j^{(8)}) &= \{p_j^{(7)}, p_j^{(8)}, c_t^{x_j}\}, & \kappa(p_j^{(2)}) &= \{p_j^{(1)}, p_j^{(2)}, c_i^{-x_j}\}, \\
\kappa(c_t^{x_j}) &= \{p_j^{(8)}, c_t^{x_j}, p_j^{(1)}\}, & \kappa(c_i^{-x_j}) &= \{p_j^{(2)}, c_i^{-x_j}, p_j^{(3)}\}, \\
\kappa(p_j^{(1)}) &= \{c_t^{x_j}, p_j^{(1)}, p_j^{(2)}\}, & \kappa(p_j^{(3)}) &= \{c_i^{-x_j}, p_j^{(3)}, p_j^{(4)}\}, \\
\kappa(p_j^{(4)}) &= \{p_j^{(3)}, p_j^{(4)}, c_r^{x_j}\}, & \kappa(p_j^{(6)}) &= \{p_j^{(5)}, p_j^{(6)}, c_s^{-x_j}\}, \\
\kappa(c_r^{x_j}) &= \{p_j^{(4)}, c_r^{x_j}, p_j^{(5)}\}, & \kappa(c_s^{-x_j}) &= \{p_j^{(6)}, c_s^{-x_j}, p_j^{(7)}\}, \\
\kappa(p_j^{(5)}) &= \{c_r^{x_j}, p_j^{(5)}, p_j^{(6)}\}, & \kappa(p_j^{(7)}) &= \{c_s^{-x_j}, p_j^{(7)}, p_j^{(8)}\}.
\end{aligned}$$

Finally, we set $k := 4|\mathcal{X}|$.

The proof of the correctness is, in essence, the same as for the case of the proof of [Theorem 4](#). If there is a satisfying truth assignment $\beta: \mathcal{X} \rightarrow \{T, F\}$ for our input instance, then we can derive a solution to our problem as follows. We say that a clause is *failed* by literal ℓ_j if either this clause contains ℓ_j and $\beta(\ell_j) = F$, or this clause contains $\neg\ell_j$ and $\beta(\ell_j) = T$. For each literal ℓ_j , we include in our solution these two clause voters that correspond to the clauses failed by ℓ_j . Further, we include those p -voters who have in their bundles these two clause voters. Since each literal is contained in exactly two clauses, one can easily verify that a thus defined solution contains exactly $4|\mathcal{X}|$ voters and that it gives p additional $8|\mathcal{X}|$ points. Since every clause C_i is failed by at most $|C_i| - 1$ literals, each clause alternative obtains at most $|C_i| - 1$ additional points. Thus, altogether, p , g , and all clause alternatives tie for victory.

The proof for the other direction is the same as in the case of [Theorem 4](#) (in particular, given a solution W , we declare a literal ℓ_j to be selected if at least one voter $c_i^{\ell_j}$ is not included in $\kappa(W)$; the proof that, if W is a correct solution for our control problem then setting the selected literals to true leads to a satisfying truth assignment, proceeds as the one of [Theorem 4](#)). \square

Taking also the swap distance d into account, we find that PLURALITY-C-CC-AV is NP-hard even if $d = 1$ and $b = 4$. This stands in contrast to the case where $d = 0$, where \mathcal{R} -C-CC-AV reduces to the CC-AV problem for weighted elections [\[25\]](#), which, for Plurality voting, is polynomial-time solvable by a simple greedy algorithm.

Theorem 8. *PLURALITY-C-CC-AV is NP-hard even for full- d bundling functions with constant value $d \geq 1$, and even if the maximum bundle size b is four.*

Proof. We can apply the same construction in W[1]-hardness proof for [Theorem 9](#) (see the next section) to provide a polynomial-time reduction from the VERTEX COVER problem instead of from the PARTIAL VERTEX COVER problem. The maximum swap distance remains the same, that is, it is one. Since the maximum bundle size is exactly one more than the maximum vertex degree of the VERTEX COVER instance, and since the VERTEX COVER problem is NP-hard [\[28\]](#) already when the maximum vertex degree is three, the statement of the theorem follows. \square

6. Single-Peaked and Single-Crossing Elections

In this section we consider the computational complexity of PLURALITY-C-CC-AV and CONDORCET-C-CC-AV for the restricted case where the elections are required to be single-peaked or single-crossing. Further, we focus on instances with full- d bundling functions; our hardness results extend to anonymous (via [Observation 2](#) in [Section 2](#)) or arbitrary bundling functions. Moreover, the W[2]-hardness result for non-anonymous bundling functions ([Theorem 1](#)) also extends to these restricted domains.

We find that the results for the combinatorial variant of control under our domain restrictions are quite different than those for the non-combinatorial case. Indeed, both for Plurality and for Condorcet, the voter control problems for single-peaked elections and for single-crossing elections are solvable in polynomial time for the non-combinatorial case [\[7, 24, 37\]](#) (for the case of Plurality, this is true even in the unrestricted case). For the combinatorial

case, we show hardness for both PLURALITY-C-CC-AV and CONDORCET-C-CC-AV for single-peaked elections, but we give polynomial-time algorithms for single-crossing elections. We begin with the case of single-peaked elections.

Theorem 9. *Both PLURALITY-C-CC-AV and CONDORCET-C-CC-AV parameterized by the solution size k are $W[1]$ -hard for single-peaked elections, even for full- d bundling functions and for any constant $d \geq 1$.*

Proof. We consider the Plurality rule first. We provide a parameterized reduction which is indeed a polynomial-time reduction from the problem PARTIAL VERTEX COVER (PVC), which is $W[1]$ -hard with respect to the parameter solution size h [30] and is defined as follows:

PARTIAL VERTEX COVER (PVC)

Input: An undirected graph $G = (V(G), E(G))$ and two non-negative integers $h, \ell \in \mathbb{N}$.

Question: Does G admit a size- h vertex subset $U \subseteq V(G)$ which intersects at least ℓ edges in G ?

Before describing the reduction itself, we first define the following *canonical preference order*:

$$p \succ g \succ a_1 \succ \bar{a}_1 \succ \dots \succ a_{|V(G)|} \succ \bar{a}_{|V(G)|}.$$

Moreover, for each set P of disjoint pairs of alternatives which are neighboring with respect to the canonical preference order, we define the preference order $\text{diff-order}(P)$ to be identical to the canonical preference order except that for each pair in P we swap the order of the alternatives in this pair.

We are now ready to describe the reduction itself. Let (G, h, ℓ) be our input instance for PVC. We form an instance for PLURALITY-C-CC-AV. First, we set the parameter $k := h$. Second, we construct an election (C, V) , where:

- (a) The set of alternatives is $C := \{p, g\} \cup \{a_i, \bar{a}_i \mid u_i \in V(G)\}$.
- (b) The set of registered voters is such that the initial score of g is $h + \ell$ and the initial scores of all the other alternatives are zero. We achieve this by forming $h + \ell$ voters, each with the canonical preference order as its preference order.

We define the set W of unregistered voters as follows:

- (1) For each edge $e = \{u_i, u_j\} \in E(G)$, we create an *edge voter* w_e with preference order $\text{diff-order}(\{\{a_i, \bar{a}_i\}, \{a_j, \bar{a}_j\}\})$ (we say that w_e corresponds to edge e). All edge voters are p -voters.
- (2) For each edge $e = \{u_i, u_j\} \in E(G)$, we create a *dummy voter* d_e with preference order $\text{diff-order}(\{\{p, g\}, \{a_i, \bar{a}_i\}, \{a_j, \bar{a}_j\}\})$ (we say that d_e corresponds to edge e). All dummy voters are g -voters.
- (3) For each vertex $u_i \in V(G)$, we create a *vertex voter* w_i^u with preference order $\text{diff-order}(\{\{a_i, \bar{a}_i\}\})$ (we say that w_i^u corresponds to u_i). All vertex voters are p -voters.

The preference orders of the voters in $V \cup W$ are single-peaked with respect to the axis:

$$\bar{a}_{|V(G)|} \succ \bar{a}_{|V(G)|-1} \succ \dots \succ \bar{a}_1 \succ p \succ g \succ a_1 \succ a_2 \succ \dots \succ a_{|V(G)|}.$$

Finally, we define the function κ to be a full-1 bundling function. To understand how κ works, below we calculate the swap distances between the preference orders of all possible pairs of voters in W . We see the following.

- (A) The distance between each two edge voters is at least two.
- (B) The distance between each edge voter and each dummy voter is exactly one if they correspond to the same edge, and is at least three otherwise.
- (C) The distance between each edge voter w_e and each vertex voter w_i^u is one if $u_i \in e$ and otherwise is three.
- (D) The distance between each two dummy voters is at least two.
- (E) The distance between each dummy voter and each vertex voter is at least two.
- (F) The distance between each two vertex voters is two.

For each edge $e = \{u_i, u_j\} \in E(G)$, we have $\kappa(w_e) := \{w_e, w_i^u, w_j^u, d_e\}$ and $\kappa(d_e) := \{w_e, d_e\}$. For each vertex $u_i \in V(G)$, we have $\kappa(w_i^u) := \{w_i^u\} \cup \{w_e \mid u_i \in e \in E(G)\}$. In this way, adding a dummy voter is never better than adding her corresponding edge voter.

We show that (G, h, ℓ) is a yes-instance for PVC if and only if there is a size- k subset $W' \subseteq W$ such that p is a Plurality winner of the election $(C, V \cup \kappa(W'))$. Recall that all unregistered voters except the dummy voters are p -voters and that p needs at least $h + \ell$ points in order to win.

For the “only if” part, suppose that $X \subseteq V(G)$ is a size- h vertex set and $Y \subseteq E(G)$ is a size- ℓ edge set such that for every edge $e \in Y$ it holds that $e \cap X \neq \emptyset$. We set $W' := \{w_i^u \mid u_i \in X\}$. It is easy to verify that $\kappa(W')$ consists of h vertex voters and at least ℓ edge voters. Each of them gives p one point if added to the election. This results in p being a winner of the election with score at least $h + \ell$.

For the “if” part, suppose that there is a size- k subset $W' \subseteq W$ such that p is a Plurality winner of the election $(C, V \cup \kappa(W'))$. Observe that if W' contains some dummy voter d_e , then we can replace her with w_e . If w_e is already in W' , then we can simply remove d_e from W' . Thus we can assume that W' does not contain any dummy voters. Now, assume that W' contains some edge voter w_e , where $e = \{u_i, u_j\}$. Since, by the previous argument, W' does not contain d_e , we have that d_e is not a member of $\kappa(W' \setminus \{w_e\})$. This means that if both w_i^u and w_j^u belong to $\kappa(W' \setminus \{w_e\})$, then we can safely remove w_e from W' ; p will still be a winner of the election $(C, V \cup \kappa(W' \setminus \{w_e\}))$. On the contrary, assume that exactly one of w_i^u and w_j^u does not belong to $\kappa(W' \setminus \{w_e\})$ and let w_i^u be this voter. It is easy to see that p is a winner of election $(C, V \cup \kappa((W' \setminus \{w_e\}) \cup \{w_i^u\}))$ (the net effect of including the bundle of w_e is that p 's score increases by at most one, whereas the net effect of including the bundle of w_i^u is that p 's score increases by at least one). Similarly, if neither w_i^u nor w_j^u with $i < j$ belong to $\kappa(W' \setminus \{w_e\})$, then it is easy to verify that p is a winner of the election $(C, V \cup \kappa((W' \setminus \{w_e\}) \cup \{w_i^u\}))$. All in all, we can assume that W' contains vertex voters only. Since all vertex voters are p -voters, without loss of generality we can assume that W' contains exactly $k = h$ of them.

We define $X := \{u_i \mid w_i^u \in W'\}$ such that $|X| = k$, and $Y := \{e \in E(G) \mid e \cap \{u_i\} \neq \emptyset\}$. By the construction of the edge voters' preference orders, $\kappa(W')$ consists of k vertex voters and $|Y|$ edge voters. This must add up to at least $h + \ell$ voters. Therefore, $|Y| \geq \ell$, implying that at least ℓ edges are covered by X . This completes the proof for the case of Plurality.

Let us now move on to the case of Condorcet rule. We use the same unregistered voters as defined above and construct the original election with $h + \ell - 1$ registered voters whose preference orders are $\text{diff-order}(\{g, p\})$. We set $k := h$. Since all voters rank either p or g at the first position, the Condorcet rule equals the Plurality rule for the unique-winner model. Thus, using the same reasoning as used for the Plurality rule, one can verify that (G, h, ℓ) is a yes-instance for PVC if and only if there is a size- k subset $W' \subseteq W$ such that p is a Condorcet winner of the election $(C, V \cup \kappa(W'))$. \square

We now present our tractability results for single-crossing elections. Con-

sider an \mathcal{R} -C-CC-AV instance $((C, V), W, d, \kappa, p, k)$, containing an election (C, V) and an unregistered voter set W such that $(C, V \cup W)$ is single-crossing (thus, both (C, V) and (C, W) are single-crossing as well). This has a crucial consequence for full- d bundling functions: for each unregistered voter $w \in W$, the voters in the bundle $\kappa(w)$ appear consecutively along the single-crossing order restricted to only the voters in W .⁸ Using the following two lemmas, we can show that PLURALITY-C-CC-AV and CONDORCET-C-CC-AV are polynomial-time solvable for full- d bundling functions.

Lemma 5. *Let $I = ((C, V), W, d, \kappa, p, k)$ be a PLURALITY-C-CC-AV instance such that $(C, V \cup W)$ is single-crossing and κ is a full- d bundling function. Then, the following statements hold:*

- (i) *The p -voters are ordered consecutively along the single-crossing order.*
- (ii) *If I is a yes-instance, then there is a subset $W' \subseteq W$ of size at most k such that (a) p is a winner of election $(C, V \cup \kappa(W'))$, and (b) all bundles of voters $w \in W'$ contain only p -voters, except at most two bundles which may contain some non- p -voters.*

Proof. Let $n := |W|$ and let $\alpha := \langle w_1, w_2, \dots, w_n \rangle$ be a single-crossing order of the voters in W . Statement (i) follows directly from the definition of the single-crossing property.

As for Statement (ii), let $W' \subseteq W$ be a size- k subset of the unregistered voters such that p is a Plurality winner in the election $(C, V \cup \kappa(W'))$. Without loss of generality, we assume that W' does not contain voters w_i whose bundles do not contain any p -voters. For each subset $S \subseteq W$ of voters, we use $\text{1st}(S)$ (respectively, $\text{2nd}(S)$) to denote the index j (respectively, j') of the first voter $w_j \in S$ (respectively, the last voter $w_{j'} \in S$) along the single-crossing order. Suppose that there are two bundles, $\kappa(w_i)$ and $\kappa(w_j)$, with $\text{1st}(\kappa(w_i)) \leq \text{1st}(\kappa(w_j))$ such that both contain non- p -voters and the first p -voter along α . If $\text{2nd}(\kappa(w_i)) \leq \text{2nd}(\kappa(w_j))$, then $\kappa(w_i)$ does not contain more p -voters than $\kappa(w_j)$ does, while containing at least as many non- p -voters as $\kappa(w_j)$. Thus, we can remove w_i from W' . Otherwise, $\text{2nd}(\kappa(w_i)) > \text{2nd}(\kappa(w_j))$, which means that $\kappa(w_j) \subset \kappa(w_i)$. Thus, we can remove w_j from W' . In any case, we conclude that W' contains at most

⁸Note that for each single-crossing election, the order of the voters possessing the single-crossing property is, in essence, unique (modulo voters with the same preference orders and modulo the fact that if an order witnesses the single-crossing property of an election, then its reverse does so as well).

one voter w whose bundle $\kappa(w)$ contains a non- p -voter and the first p -voter (along the single-crossing order).

Analogously, we can show that W' contains at most one voter w whose bundle $\kappa(w)$ contains a non- p -voter and the last p -voter (along the single-crossing order). To complete the proof, it remains to observe that for each $w \in W$ it holds that if $\kappa(w)$ contains both some p -voters and some non- p -voters, then it also must contain either the first p -voter or the last p -voter (along the single-crossing order). Altogether, this implies that W' contains at most two voters whose bundles contain non- p -voters. \square

For Condorcet voting, we use the well-known median-voter theorem [5] (we provide the proof for the sake of completeness).

Lemma 6. *Let $(C, V \cup \kappa(W'))$ be a single-crossing election with single-crossing voter order $\langle x_1, x_2, \dots, x_z \rangle$ and set $X_{\text{median}} := \{x_{\lceil z/2 \rceil}\} \cup \{x_{z/2+1} \text{ if } z \text{ is even}\}$, where $z = |V| + |\kappa(W')|$. Alternative p is a (unique) Condorcet winner in $(C, V \cup \kappa(W'))$ if and only if every voter in X_{median} is a p -voter.*

Proof. Let X_1 be the set of voters $x_1, x_2, \dots, x_{\lceil z/2 \rceil - 1}$ and let X_2 be the set of voters $V \cup \kappa(W') \setminus (X_1 \cup X_{\text{median}})$.

For the “if” part, let c be an arbitrary alternative from $C \setminus \{p\}$. Then, if there is some voter in X_1 which prefers c over p , then by the assumption that every voter in X_{median} is a p -voter it follows that all voters in $X_{\text{median}} \cup X_2$ are p -voters. Analogously, if there is some voter in X_2 which prefers c over p , then all voters in $X_1 \cup X_{\text{median}}$ are p -voters. In any case, a strict majority of voters are p -voters. Thus, p is the (unique) Condorcet winner.

For the “only if” part, suppose for the sake of contradiction that p is a Condorcet winner while there is a voter in X_{median} which is not a p -voter but a c -voter with $c \in C \setminus \{p\}$. Then, analogously to the reasoning above, at least half of the voters prefer c over p —a contradiction. \square

With these two lemmas available, we describe next polynomial-time algorithms for both PLURALITY-C-CC-AV and CONDORCET-C-CC-AV, for the case of single-crossing elections and full- d bundling functions.

Theorem 10. *Both PLURALITY-C-CC-AV and CONDORCET-C-CC-AV are polynomial-time solvable for the single-crossing case with full- d bundling functions.*

Proof. First, we find a (unique) single-crossing voter order for $(C, V \cup W)$ in quadratic time [18, 9]. Due to Lemma 5 and Lemma 6, we only need

to store the most preferred alternative of each voter in order to find the solution set W' . Thus, the running time from now on only depends on the number of voters. We start with the Plurality rule and let $\alpha := \langle w_1, w_2, \dots, w_{|W|} \rangle$ be a single-crossing voter order.

Due to [Lemma 5 \(ii\)](#), the two bundles in $\kappa(W')$ which may contain non- p -voters appear at the beginning and at the end of the p -voter block, along the single-crossing order. We first guess these two bundles, and after this initial guess, all remaining bundles in the solution contain only p -voters ([Lemma 5 \(i\)](#)). Thus, the remaining task is to find the maximum score that p can gain by selecting k' bundles containing only p -voters. This problem is equivalent to the MAXIMUM INTERVAL COVER problem, which is solvable in $O(|W|^2)$ time, by using dynamic programming, as described by Golab et al. [[29](#), Section 3.2].

For the Condorcet rule, we propose a slightly different algorithm. The goal is to find a subset $W' \subseteq W$ of minimum size such that p is the (unique) Condorcet winner in $(C, V \cup \kappa(W'))$. Let $\beta := \langle x_1, x_2, \dots, x_z \rangle$ be a single-crossing voter order for $(C, V \cup W)$. Considering [Lemma 6](#), we begin by guessing at most two voters in $V \cup W$ whose bundles may contain the median p -voter (or, possibly, several p -voters) along the single-crossing order of voters restricted to the final election (for simplicity, we define the bundle of each registered voter to be her singleton). Let these two bundles be $A_1 := \{x_i, x_{i+1}, \dots, x_{i+t'}\}$ and $A_2 := \{x_{i+t''}, x_{i+t''+1}, \dots, x_{i+j}\}$ for some $i \geq 1$, $t', t'', j \geq 0$. Let $W_1 := \{x_s \in W \mid s < i\}$, and let $W_2 := \{x_s \in W \mid s > i+j\}$. We guess two integers $z_1 \leq |W_1|$ and $z_2 \leq |W_2|$ with the property that there are two subsets $B_1 \subseteq W_1$ and $B_2 \subseteq W_2$ with $|B_1| = z_1$ and $|B_2| = z_2$ such that the median voter(s) in $V \cup B_1 \cup A_1 \cup A_2 \cup B_2$ are indeed p -voters (for now, only the sizes z_1 and z_2 matter, not the actual sets). These four guesses require time $O(|V \cup W|^2 \cdot |W|^2)$. The remaining task is to find two subsets W'_1 and W'_2 of minimum size such that $\kappa(W'_1) \subseteq W_1$, $\kappa(W'_2) \subseteq W_2$, $|\kappa(W'_1)| = z_1$, and $|\kappa(W'_2)| = z_2$. As discussed for the case of the Plurality rule, this can be done in time $O(|W|^2)$ using the algorithm of Golab et al. [[29](#), Section 3.2]. We conclude that one can find a minimum-size subset $W' \subseteq W$ such that p is the (unique) Condorcet winner in $(C, V \cup \kappa(W'))$ in time $O(|V \cup W|^2 \cdot |W|^4)$. \square

7. Conclusion

Our work provides several opportunities for future research. First, we did not discuss destructive control and the related problem of combinatorial deletion of voters. For Plurality, we conjecture that combinatorial addition

of voters for destructive control, and combinatorial deletion of voters for either constructive or destructive control behave similarly to combinatorial addition of voters for constructive control.

Another field of future research is to study other combinatorial voting models—this may include controlling the swap distance, “probabilistic bundling”, “reverse bundling”, or using other distance measures than the swap distance. Naturally, it would also be interesting to consider other problems than election control (with bribery indeed, in a follow-up to this work some of the co-authors consider a combinatorial variant of SHIFT BRIBERY [11]).

Finally, instead of studying a “leader-follower model” as we did, one might also be interested in an “enemy model” referring to control by adding alternatives: the alternatives of an election “hate” each other so that if one alternative is added to the election, then all of its enemies are also added to the election. This scenario of combinatorial candidate control deserves future investigation, already partially conducted by Chen et al. [13].

References

- [1] J. J. Bartholdi, III and M. Trick. Stable matching with preferences derived from a psychological model. *Operations Research Letters*, 5(4): 165–169, 1986.
- [2] J. J. Bartholdi, III, C. A. Tovey, and M. A. Trick. How hard is it to control an election. *Mathematical and Computer Modelling*, 16(8–9): 27–40, 1992.
- [3] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52):5425–5442, 2009.
- [4] N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting—a parameterized complexity perspective. In *The Multivariate Algorithmic Revolution and Beyond*, volume 7370 of *Lecture Notes in Computer Science*, pages 318–363. Springer, 2012.
- [5] D. Black. On the rationale of group decision making. *Journal of Political Economy*, 56(1):23–34, 1948.
- [6] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ce-

- teris paribus* preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.
- [7] F. Brandt, M. Brill, E. Hemaspaandra, and L. A. Hemaspaandra. Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI '10)*, pages 715–722. AAAI Press, 2010.
 - [8] F. Brandt, P. Harrenstein, K. Kardel, and H. G. Seedig. It only takes a few: On the hardness of voting with a constant number of agents. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '13)*, pages 375–382. IFAAMAS, 2013.
 - [9] R. Bredereck, J. Chen, and G. J. Woeginger. A characterization of the single-crossing domain. *Social Choice and Welfare*, 41(4):989–998, 2013.
 - [10] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI '14)*, pages 1398–1404. AAAI Press, 2014.
 - [11] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Large-scale election campaigns: Combinatorial shift bribery. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*. IFAAMAS, 2015.
 - [12] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Combinatorial voter control in elections. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14)*, volume 8635 of *Lecture Notes in Computer Science*, pages 153–164. Springer, 2014.
 - [13] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Elections with few voters: Candidate control can be easy. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*, pages 2045–2051. AAAI Press, 2015.
 - [14] V. Conitzer. Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, 35:161–191, 2009.

- [15] V. Conitzer, J. Lang, and L. Xia. How hard is it to control sequential elections via the agenda? In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI '09)*, pages 103–108. AAAI Press, 2009.
- [16] M. J. A. N. C. de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: L'Imprimerie Royale, 1785.
- [17] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
- [18] E. Elkind, P. Faliszewski, and A. Slinko. Clone structures in voters' preferences. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC '12)*, pages 496–513. ACM Press, 2012.
- [19] G. Erdélyi, E. Hemaspaandra, and L. A. Hemaspaandra. More natural models of electoral control by partition. Technical report arXiv:1410.2652, Oct. 2014.
- [20] G. Erdélyi, M. R. Fellows, J. Rothe, and L. Schend. Control complexity in Bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences*, 81(4):632–660, 2015.
- [21] B. Escoffier, J. Lang, and M. Öztürk. Single-peaked consistency and its complexity. In *Proceedings of the 18th European Conference on Artificial Intelligence (ECAI '08)*, pages 366–370. IOS Press, 2008.
- [22] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland Voting Computationally Resist Bribery and Constructive Control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.
- [23] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.
- [24] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Information and Computation*, 209(2):89–107, 2011.

- [25] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Weighted electoral control. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '13)*, pages 367–374. IFAAMAS, 2013.
- [26] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The complexity of manipulative attacks in nearly single-peaked electorates. *Artificial Intelligence*, 207:69–99, 2014.
- [27] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
- [28] M. R. Garey, D. S. Johnson, and L. J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.
- [29] L. Golab, H. Karloff, F. Korn, A. Saha, and D. Srivastava. Sequential dependencies. In *35th International Conference on Very Large Data Bases (PVLDB '09)*, volume 2(1), pages 574–585. VLDB Endowment, 2009.
- [30] J. Guo, R. Niedermeier, and S. Wernicke. Parameterized complexity of Vertex Cover variants. *Theory of Computing Systems*, 41(3):501–520, 2007.
- [31] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.
- [32] L. A. Hemaspaandra, R. Lavaee, and C. Menton. Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '13)*, pages 1345–1346. IFAAMAS, 2013.
- [33] R. Kannan. Minkowski’s convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.
- [34] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.
- [35] H. Liu and D. Zhu. Parameterized complexity of control problems in Maximin election. *Information Processing Letters*, 110(10):383–388, 2010.

- [36] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, 410(27-29):2746–2753, 2009.
- [37] K. Magiera and P. Faliszewski. How hard is control in single-crossing elections? In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI '14)*, pages 579–584. IOS Press, 2014.
- [38] N. Mattei, M. S. Pini, F. Rossi, and K. B. Venable. Bribery in voting over combinatorial domains is easy. In *Proceedings of the 12th International Symposium on Artificial Intelligence and Mathematics (ISAIM '12)*, 2012.
- [39] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.
- [40] S. Micali and V. V. Vazirani. An $O(\sqrt{|V|} \cdot |E|)$ algorithm for finding maximum matching in general graphs. In *Proceedings of the 21st Annual IEEE Symposium on Foundations of Computer Science (FOCS '80)*, volume 21, pages 17–27. IEEE Computer Society Press, 1980.
- [41] J. A. Mirrlees. An exploration in the theory of optimal income taxation. *Review of Economic Studies*, 38(2):175–208, 1971.
- [42] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
- [43] D. Parkes and L. Xia. A complexity-of-strategic-behavior comparison between Schulze’s rule and ranked pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI '12)*, pages 1429–1435. AAAI Press, 2012.
- [44] K. W. Roberts. Voting over income tax schedules. *Journal of Public Economics*, 8(3):329–340, 1977.
- [45] J. Rothe and L. Schend. Control complexity in Bucklin, fallback, and plurality voting: An experimental approach. In *Proceedings of the 11th International Symposium on Experimental Algorithms (SEA '12)*, volume 7276 of *Lecture Notes in Computer Science*, pages 356–368. Springer, June 2012.

- [46] M. H. Rothkopf, A. Pekeč, and R. M. Harstad. Computationally manageable combinatorial auctions. *Management Science*, 44(8):1131–1147, 1998.
- [47] T. Sandholm. Optimal winner determination algorithms. In P. Cramton, Y. Shoham, and R. Steinberg, editors, *Combinatorial Auctions*, chapter 14. MIT Press, 2006.
- [48] A. Saporiti and F. Tohmé. Single-crossing, strategic voting and the median choice rule. *Social Choice and Welfare*, 26(2):363–383, 2006.
- [49] C. A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984.
- [50] T. Walsh. Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd Conference on Artificial Intelligence (AAAI '07)*, pages 3–8. AAAI Press, 2007.