# On the Computational Complexity of Variants of Combinatorial Voter Control in Elections

Leon Kellerhals, Viatcheslav Korenwein, Philipp Zschoche*, Robert Bredereck**, and Jiehua Chen

TU Berlin, Germany
{leon.kellerhals, viatcheslav.korenwein, zschoche}@campus.tu-berlin.de
{robert.bredereck, jiehua.chen}@tu-berlin.de

**Abstract.** Voter control problems model situations in which an external agent tries to affect the result of an election by adding or deleting the fewest number of voters. The goal of the agent is to make a specific candidate either win (*constructive* control) or lose (*destructive* control) the election. We study the constructive and destructive voter control problems when adding and deleting voters have a *combinatorial flavor*: If we add (resp. delete) a voter $v$, we also add (resp. delete) a bundle $\kappa(v)$ of voters that are associated with $v$. While the bundle $\kappa(v)$ may have more than one voter, a voter may also be associated with more than one voter. We analyze the computational complexity of the four voter control problems for the Plurality rule.

We obtain that, in general, making a candidate lose is computationally easier than making her win. In particular, if the bundling relation is symmetric (i.e. $\forall w\colon w \in \kappa(v) \Leftrightarrow v \in \kappa(w)$), and if each voter has at most two voters associated with him, then destructive control is polynomial-time solvable while the constructive variant remains NP-hard. Even if the bundles are disjoint (i.e. $\forall w\colon w \in \kappa(v) \Leftrightarrow \kappa(v) = \kappa(w)$), the constructive problem variants remain intractable. Finally, the minimization variant of constructive control by adding voters does not admit an efficient approximation algorithm, unless P = NP.

## 1 Introduction

Since the seminal paper by Bartholdi III et al. [3] on controlling an election by adding or deleting the fewest number of voters or candidates with the goal of making a specific candidate to win (*constructive control*), a lot of research has been devoted to the study of control for different voting rules [4, 13, 15, 20, 22, 23], on different control modes [16, 17], or even on other controlling goals (e.g. aiming at several candidates' victory or a specific candidate's defeat) [19, 25]. Recently, Bulteau et al. [8] introduced combinatorial structures to constructive control by adding voters: When a voter is added, a bundle of other voters is added as well.

* PZ was supported by the Stiftung Begabtenförderung berufliche Bildung (SBB).
** RB was from September 2016 to September 2017 on postdoctoral leave at the University of Oxford (GB), supported by the DFG fellowship BR 5207/2.

A combinatorial structure of the voter set allows us to model situations where an external agent hires speakers to convince whole groups of people to participate in (or abstain from) an election. In such a scenario, convincing a whole group of voters comes at the fixed cost of paying a speaker. Bulteau et al. [8] model this by defining a bundle of associated voters for each voter which will be convinced to vote "for free" when this voter is added or deleted. Moreover, the bundles of different voters could overlap. For instance, convincing two bundles of two voters each to participate in the election could result in adding a total of four, three or even just two voters.

We extend the work of Bulteau et al. [8] and investigate the cases where the agent wants to make a specific candidate win or lose by adding (resp. deleting) the fewest number of bundles. We study one of the simplest voting rules, the Plurality rule, where each voter gives one point to his favorite candidate, and the candidate with most points becomes a winner. Accordingly, an election consists of a set $C$ of candidates and a set $V$ of voters who each have a favorite candidate. Since real world elections typically contain only a small number of candidates, and a bundle of voters may correspond to a family with just a few members, we are especially interested in situations where the election has only few candidates and the bundle of each voter is small. Our goal is to ensure that a specific candidate $p$ becomes a winner (or a loser) of a given election, by convincing as few voters from an unregistered voter set $W$ as possible (or as few voters from $V$ as possible), together with the voters in their bundles, to participate (or not to participate) in the election. We study the combinatorial voter control problems from both the classical and the parameterized complexity point of view. We confirm Bulteau et al.'s conjecture [8] that for the Plurality rule, the three problem variants: combinatorial constructive control by deleting voters and combinatorial destructive control by adding (resp. deleting) voters, behave similarly in complexity to the results of combinatorial constructive control by adding voters: They are NP-hard and intractable even for very restricted cases. We can also identify several special cases, where the complexity of the four problems behave differently. For instance, we find that constructive control tends to be computationally harder than destructive control. We summarize our results in Table 1.

**Related Work.** Bartholdi III et al. [3] introduced the complexity study of election control problems and showed that for the Plurality rule, the non-combinatorial variant of the voter control problems can be solved in linear time by a using simple greedy strategy. We refer the readers to the work by Faliszewski and Rothe [14], Rothe and Schend [26] for general expositions on election control problems.

In the original election control setting, a unit modification of the election concerns usually a single voter or candidate. The idea of adding combinatorial structure to election voter control was initiated by Bulteau et al. [8]: Instead of adding a voter at each time, one adds a "bundle" of voters to the election, and the bundles added to the election could intersect with each other. They showed that combinatorial constructive control by adding the fewest number of

bundles becomes notorious hard, even for the Plurality rule and for only two candidates. Chen [9] mentioned that even if each bundle has only two voters and the underlying bundling graph is acyclic, the problem still remains NP-hard. Bulteau et al. [8] and Chen [9] conjectured that

> "*the 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.*"

The combinatorial structure notion for voter control has also been extended to candidate control [10] and electoral shift bribery [7].

**Paper Outline.** In Section 2, we introduce the notation used throughout the paper. In Section 3 we formally define the four problem variants, summarize our contributions, present results in which the four problem variants (constructive or destructive, adding voters or deleting voters) behave similarly, and provide reductions between the problem variants. Sections 4 to 6 present our main results on three special cases

(1) when the bundles and the number of candidates are small,
(2) when the bundles are disjoint, and
(3) when the solution size could be unlimited.

We conclude in Section 7 with several future research directions. Due to space restrictions, some proofs are deferred to our technical report [21].

## 2 Preliminaries

The notation we use in this paper is based on Bulteau et al. [8]. We assume familiarity with standard notions regarding algorithms and complexity theory. For each $z \in \mathbb{N}$ we denote by $[z]$ the set $\{1, \ldots, z\}$.

**Elections.** An *election* $E = (C, V)$ consists of a set $C$ of $m$ *candidates* and a set $V$ of *voters*. Each voter $v \in V$ has a favorite candidate $c$ and we call voter $v$ a $c$-voter. Note that since we focus on the Plurality rule, we simplify the notion of the preferences of voters in an election to the favorite candidate of each voter. For each candidate $c \in C$ and each subset $V' \subseteq V$ of voters, the *(Plurality) score* $s_c(V')$ of candidate $c$ with respect to the voter set $V'$ is defined as the number of voters from $V'$ that have her as favorite candidate. We say that a candidate $c$ is a *winner* of election $(C, V)$ if $c$ has the highest score $s_c(V)$. For the sake of convenience, for each $C' \subseteq C$, a $C'$-voter denotes a voter whose favorite candidate belongs to $C'$.

**Combinatorial Bundling Functions.** Given a voter set $X$, a *combinatorial bundling function* $\kappa\colon X \to 2^X$ (abbreviated as *bundling function*) is a function that assigns a set of voters to each voter; we require that $x \in \kappa(x)$. For the sake of convenience, for each subset $X' \subseteq X$, we define $\kappa(X') = \bigcup_{x \in X'} \kappa(x)$. For a

voter $x \in X$, $\kappa(x)$ is named *$x$'s bundle*; $x$ is called the *leader* of the bundle. We let $b$ denote the *maximum bundle size* of a given $\kappa$. Formally, $b = \max_{x \in X} |\kappa(x)|$. One can think of the bundling function as subsets of voters that can be added at a unit cost (e.g. $\kappa(x)$ is a group of voters influenced by $x$).

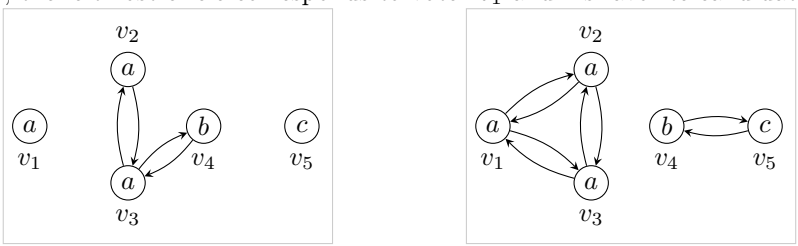
**Bundling graphs.** The *bundling graph* of an election is a model of how the voters' bundles interact with each other. Let $\kappa \colon X \to 2^X$ be a bundling function. The *bundling graph* $G_\kappa = (V(G_\kappa), E(G_\kappa))$ of $\kappa$ is a simple, directed graph, where for each voter $x \in X$ there is a vertex $x \in V(G_\kappa)$ with the same name, and for each two distinct voters $y, z \in X$ with $z \in \kappa(y)$, there is an arc $(y, z) \in E(G_\kappa)$.

We consider three special cases of bundling functions/graphs which we think are natural in real world. We say that a bundling function $\kappa$ is *symmetric* if for each two distinct voters $x, y \in X$, it holds that $y \in \kappa(x)$ if and only if $x \in \kappa(y)$. The bundling graph for a symmetric bundling function always has two directed arcs connecting each two vertices. Thus, we can assume the graph to be undirected.

We say that $\kappa$ is *disjoint* if for each two distinct voters $x, y \in X$, it holds that either $\kappa(x) = \kappa(y)$ or $\kappa(x) \cap \kappa(y) = \emptyset$. It is an easy exercise to verify that disjoint bundling functions are symmetric and the corresponding undirected bundling graphs consist only of disjoint complete subgraphs.

We say that $\kappa$ is *anonymous* if for each two distinct voters $x$ and $y$ with the same favorite candidate, it holds that $\kappa(x) = \kappa(y)$, and that for all other voters $z$ we have $x \in \kappa(z)$ if and only if $y \in \kappa(z)$.

*Example 1.* For an illustration, consider the following election $E \coloneqq (C = \{a, b, c\}$, $V = \{v_1, v_2, \ldots, v_5\})$ in which the favorite candidate of voters $v_1, v_2, v_3$ is $a$, the favorite candidate of $v_4$ is $b$, and the favorite candidate of $v_5$ is $c$. The bundling graph corresponding to the bundling function of this election could be either the left or the right figure as depicted below. Note that the label above or below the circle (which represents the vertex) denotes the name of the voter and the label inside the circle indicates his favorite candidate. For instance, in the left figure below, the leftmost circle corresponds to voter $v_1$ and his favorite candidate is $a$.




The bundling function corresponding to the left bundling graph is symmetric, but neither disjoint nor anonymous. The bundling function corresponding to the right bundling graph is symmetric, disjoint, and anonymous.

**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* [12, 18, 24]. A parameterized problem is called *fixed-parameter tractable* (in FPT) if

there is an algorithm that solves each instance $(I, r)$ in $f(r) \cdot |I|^{O(1)}$ time, where $f$ is a computable function depending only on the parameter $r$.

There is also a hierarchy of hardness classes for parameterized problems, of which the most important ones are W[1] and W[2]. One can show that a parameterized problem $L$ is (presumably) not in FPT 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 two computable functions $f$ and $g$, given an instance $(I, r)$ of problem $L$, it computes in $f(r) \cdot |I|^{O(1)}$ time an instance $(I', r')$ of problem $L'$ so that $r' \leq g(r)$ and that $(I, r) \in L$ if and only if $(I', r') \in L'$. For a survey of research on parameterized complexity in computational social choice, we refer to Betzler et al. [5] and Bredereck et al. [6].

## 3 Central Problem

We consider the problem of *combinatorial voter control* in four variants. The variants differ in whether they are *constructive* or *destructive*, meaning that the goal is to make one selected candidate win or lose the election. This goal can be achieved by either *adding* voters to or *deleting* voters from the given election. Due to space constraints, we only provide the definition of constructive control. Destructive control is defined analogously.

**Combinatorial Constructive Control by Adding**
(resp. **Deleting**) **Voters** [**C-Cons-Add** (resp. **C-Cons-Del**)]

**Input:** An election $E = (C, V)$, a set $W$ of unregistered voters with $V \cap W = \emptyset$, a bundling function $\kappa\colon W \to 2^W$ (resp. $\kappa\colon V \to 2^V$), a preferred winner $p \in C$, and an integer $k \in \mathbb{N}$.

**Quest.:** Is there a size-at-most $k$ subset $W' \subseteq W$ (resp. $V' \subseteq V$) of voters such that $p$ wins the election $(C, V \cup \kappa(W'))$ (resp. $(C, V \setminus \kappa(V'))$)?

It is straight-forward to see that all four problem variants are contained in NP since we can check in polynomial time whether a given subset $W'$ (or $V'$) is a desired solution of size at most $k$.

Throughout this work, when speaking of the *"adding"* or *"deleting"* variants, we mean those variants in which voters are added or, respectively, deleted. In similar fashion, we speak of the *constructive* and *destructive* (abbr. by "Cons" and by "Des", respectively) problem variants. Further, we refer to the set $W'$ of voters as the solution for the "adding" variants (the set $V'$ of voters for the "deleting" variants, respectively) and denote $k$ as the solution size.

**Our Contributions.** We study both the classical and the parameterized complexity of the four voter control variants. We are particularly interested in the real-world setting where the given election has a small number of candidates and where only a few voters are associated to a voter. On the one hand, we were able to confirm the conjecture given by Bulteau et al. [8] and Chen [9] that when parameterized by the solution size, C-Cons-Del, C-Des-Add, and C-Des-Del are all intractable even for just two candidates or for bundle sizes of

**Table 1.** Computational complexity of the four combinatorial voter control variants with the Plurality rule. The parameters are "the solution size $k$", "the number $m$ of candidates" and "the maximum bundle size $b$". We refer to $|I|$ as the instance size. The rows distinguish between different maximum bundle sizes $b$ and the number $m$ of candidates. All parameterized intractability results are for the parameter "solution size $k$". ILP-FPT means FPT based on a formulation as an integer linear program and the result is for the parameter "number $m$ of candidates".

| | C-Cons-Add | C-Cons-Del | C-Des-Add | C-Des-Del | References |
|---|---|---|---|---|---|
| $\kappa$ symmetric | | | | | |
| $b = 2$ | $O(\|I\|)$ | P | $O(m \cdot \|I\|)$ | $O(m \cdot \|I\|)$ | Obs 2, Thm 3 Thm 5 |
| $b = 3$ | | | | | |
| $m = 2$ | $O(\|I\|^5)$ | $O(\|I\|^5)$ | $O(\|I\|^5)$ | $O(\|I\|^5)$ | Thm 2, Cor 1+2 |
| $m$ unbounded | NP-h | NP-h | $O(m \cdot \|I\|^5)$ | $O(m \cdot \|I\|^5)$ | Obs 1, Prop 2, Cor 2 |
| $b$ unbounded | | | | | |
| $m = 2$ | W[2]-h | W[2]-h | W[2]-h | W[2]-h | [8], Thm 1 |
| $m$ unbounded and $\kappa$ disjoint | W[1]-h | W[2]-h | $O(m \cdot \|I\|)$ | $O(m \cdot \|I\|)$ | Thm 4+5 |
| $\kappa$ anonymous | ILP-FPT | ILP-FPT | ILP-FPT | ILP-FPT | Thm 1 |
| $\kappa$ arbitrary | | | | | |
| $b = 3$, $m = 2$ | W[1]-h | W[1]-h | W[1]-h | W[1]-h | Thm 1 |

at most three, and that when parameterized by the number of candidates, they are fixed-parameter tractable for anonymous bundling functions. On the other hand, we identify interesting special cases where the four problems differ in their computational complexity. We conclude that in general, destructive control tends to be easier than constructive control: For symmetric bundles with at most three voters, C-Cons-Add is known to be NP-hard, while both destructive problem variants are polynomial-time solvable. For disjoint bundles, constructive control is parameterized intractable (for the parameter "solution size $k$"), while destructive control is polynomial-time solvable. Unlike for C-Cons-Del, a polynomial-time approximation algorithm for C-Cons-Add does not exist, unless P = NP. Our results are gathered in Table 1.

The following theorem summarizes the conjecture given by Bulteau et al. [8] and Chen [9]. The corresponding proof can be found in our technical report [21].

**Theorem 1.** *All four combinatorial voter control problem variants are*

*(i)* W[2]*-hard with respect to the solution size $k$, even for only two candidates and for symmetric bundling functions $\kappa$*

*(ii)* W[1]*-hard with respect to the solution size $k$, even for only two candidates and for bundle sizes of at most three.*

*(iii) fixed-parameter tractable with respect to the number $m$ of candidates if the bundling function $\kappa$ is anonymous.*

**Relations between the four problem variants.** We provide some reductions between the problem variants. They are used in several sections of this paper. The key idea for the reduction from destructive control to constructive control is to guess the candidate that will defeat the distinguished candidate and ask whether one can make this candidate win the election. The key idea for the reduction from the "deleting" to the "adding" problem variants is to build the "complement" of the registered voter set.

**Proposition 1.** *For each $X \in \{$Add, Del$\}$,* C-Des-$X$ *with $m$ candidates is Turing reducible to* C-Cons-$X$ *with two candidates. For each $Y \in \{$Cons, Des$\}$,* C-$Y$-Del *with two candidates is many-one reducible to* C-$Y$-Add *with two candidates. All these reductions preserve the property of symmetry of the bundling functions.*

## 4 Controlling Voters with Symmetric and Small Bundles

In this section, we study combinatorial voter control when the voter bundles are symmetric and small. This could be the case when a voter's bundle models his close friends (including himself), close relatives, or office mates. Typically, this kind of relations is symmetric, and the number of friends, relatives, or office mates is small. We show that for symmetric bundles and for bundles size at most three, both destructive problem variants become polynomial-time solvable, while both constructive variants remain NP-hard. However, if there are only two candidates, then we can use dynamic programming to also solve the constructive control variants in polynomial time. If we restrict the bundle size to be at most two, then all four problem variants can be solved in polynomial time via simple greedy algorithms.

As already observed in Section 2, we only need to consider the undirected version of the bundling graph for symmetric bundles. Moreover, if the bundle size is at most two, then the resulting bundling graph consists of only cycles and trees. However, Bulteau et al. [8] already observed that C-Cons-Add is NP-hard even if the resulting bundling graph solely consists of cycles, and Chen [9] observed that C-Cons-Add remains NP-hard even if the resulting bundling graph consists of only directed trees of depth at most three.

**Observation 1.** C-Cons-Add *is* NP*-hard even for symmetric bundling functions with maximum bundle size $b = 3$.*

It turns out that the reduction used by Bulteau et al. [8] to show the adding voters case (Observation 1) can be adapted to show NP-hardness for the deleting voters case.

**Proposition 2.** C-Cons-Del *is* NP*-hard even for symmetric bundling functions with maximum bundle size $b = 3$.*

If, in addition to the bundles being symmetric and of size at most three, we have only two candidates, then we can solve C-Cons-Add in polynomial time.

First of all, due to these constraints, we can assume that the bundling graph $G_\kappa$ is undirected and consists of only cycles and paths. Then, it is easy to verify that we can consider each cycle and each path separately. Finally, we devise a dynamic program for the case when the bundling graph is a path or a cycle, maximizing the score difference between our preferred candidate $p$ and the other candidate. The crucial idea behind the dynamic program is that the bundles of a minimum-size solution induce a subgraph where each connected component is small.

**Lemma 1.** *Let $(E = (C, V), W, \kappa, p, k)$ be a* C-Cons-Add *instance such that $C = \{p, g\}$, and $\kappa$ is symmetric with $G_\kappa$ being a path. Then, finding a size-at-most-$k$ subset $W' \subseteq W$ of voters such that the score difference between $p$ and $g$ in $\kappa(W')$ is maximum can be solved in $O(|W|^5)$ time, where $|W|$ is the size of the unregistered voter set $W$.*

*Proof.* Since $G_\kappa$ is a path, each bundle has at most three voters. We denote the path in $G_\kappa$ by $(w_1, w_2, \ldots, w_{|W|})$ and introduce some definitions for this proof. The set $W(s,t) \coloneqq \{w_i \in W \mid s \leq i \leq t\}$ contains all voters on a sequence from $w_s$ to $w_t$. For every subset $W' \subseteq W$ we define $\mathsf{gap}(W') \coloneqq s_p(\kappa(W')) - s_g(\kappa(W'))$ as the score difference between $p$ and $g$. One can observe that if $W'$ is a solution for $(E = (C, V), W, \kappa, p, k)$ then $\mathsf{gap}(W') \geq s_g(V) - s_p(V)$; note that we only have two candidates. An *$(s,t)$-proper-subset* $W'$ is a subset of $W(s,t)$ such that $\kappa(W') \subseteq W(s,t)$. A *maximum $(s,t)$-proper-subset* $W'$ additionally requires that each $(s,t)$-proper-subset $W'' \subseteq W$ with $|W''| = |W'|$ has $\mathsf{gap}(W'') \leq \mathsf{gap}(W')$.

We provide a dynamic program in which a table entry $T[r,s,t]$ contains a maximum $(s,t)$-proper-subset $W'$ of size $r$. We first initialize the table entries for the case where $t - s + 1 \leq 9$ and $r \leq 9$ in linear time.

For $t - s + 1 > 9$, we compute the table entry $T[r,s,t]$ by considering every possible partition of $W(s,t)$ into two disjoint parts.

$$T[r,s,t] \coloneqq T[r-i, s, s+j] \cup T[i, s+j+1, t],$$
$$\text{where } i,j = \operatorname*{arg\,max}_{\substack{0 \leq i \leq r \\ 0 \leq j \leq t-s-2}} \mathsf{gap}(T[r-i,s,s+j]) + \mathsf{gap}(T[i,s+j+1,t]).$$

Note that a maximum $(1, |W|)$-proper-subset $W'$ of size $r-1$ *could* have a higher $\mathsf{gap}$ than a maximum $(1, |W|)$-proper-subset $W''$ of size $r$.

To show the correctness of our program, we define the maximization and minimization function on a set of voters $W'$, which return the largest and smallest index of all voters on the path induced by $W'$, respectively:

$$\max(W') \coloneqq \operatorname*{arg\,max}_{i \in |W'|} \{w_i \in (W')\} \text{ and } \min(W') \coloneqq \operatorname*{arg\,min}_{i \in |W'|} \{w_i \in (W')\}.$$

First, we use the following claim to show that each maximum $(s,t)$-proper-subset $W'$ can be partitioned into two $(s,t)$-proper-subsets $W_1, W_2$ such that the two sets $\kappa(W_1)$ and $\kappa(W_2)$ are disjoint. The correctness proof of the following claim can be found in our technical report [21].

**Claim 1.** *Let $W'$ be a maximum $(s,t)$-proper-subset such that $(\max\kappa(W') - \min\kappa(W') + 1) > 9$. Then, there is a $j$ with $s < j < t$ such that there is an $(s,j)$-proper-subset $W_1$ and a $(j+1,t)$-proper-subset $W_2$ with $|W_1| + |W_2| \leq |W'|$ and $\kappa(W_1 \cup W_2) = \kappa(W')$.*

Now, we show that the two subsets $W_1$ and $W_2$ from Claim 1 are indeed *optimal*: There is a $j$ such that $W_1$ is a maximum $(s,j)$-proper-subset and $W_2$ is a maximum $(j+1,t)$-proper-subset.

Assume towards a contradiction that $W_2$ is a $(j+1,t)$-proper-subset but not a maximum $(j+1,t)$-proper-subset. Therefore, there exists a maximum $(j+1,t)$-proper-subset $W_2'$ where $|W_2| = |W_2'|$. This implies that $\mathsf{gap}(W_1 \cup W_2') > \mathsf{gap}(W_1 \cup W_2)$. This is a contradiction to $W' = W_1 \cup W_2$ being a maximum $(s,t)$-proper-subset. The case in which $W_1$ is not a maximum $(s,j)$-proper-subset is analogous.

Altogether, we have shown that we can compute $T[k,s,t]$ in constant time if $t - s + 1 \leq 9$, and that otherwise there exist an $i$ and a $j$ such that $T[k,s,t] = T[k-i,s,t-j] \cup T[i,t-j+1,t]$. The dynamic program considers all possible $i$ and $j$. The table entry $T[i,1,|W|]$ contains a subset $W' \subseteq W$ of size $i$ with maximum $\mathsf{gap}$ such that $\kappa(W') \subseteq W(1,|W|)$, which is identical to $\kappa(W') \subseteq W$.

This completes the correctness proof of our dynamic program. The table has $O(k \cdot |W|^2)$ entries. To compute one entry the dynamic program accesses $O(k \cdot |W|)$ other table entries. Note that the value $\mathsf{gap}(T[i,s,t])$ can be computed and stored after the entry $T[i,s,t]$ is computed. This takes at most $O(|W|)$ steps. Thus, the dynamic program runs in $O(|W|^5)$ time. □

The dynamic program can be adapted to solve the same problem on cycles:

**Lemma 2.** *Let $(E = (C,V), W, \kappa, p, k)$ be a* C-Cons-Add *instance such that $C = \{p,g\}$, and $\kappa$ is symmetric with $G_\kappa$ being a cycle. Then, finding a size-at-most-$k$ subset $W' \subseteq W$ of voters such that the score difference between $p$ and $g$ in $\kappa(W')$ is maximum can be solved in $O(|W|^5)$ time, where $|W|$ is the size of the unregistered voter set $W$.*

Altogether, we obtain the following.

**Theorem 2.** C-Cons-Add *with a symmetric bundling function, maximum bundle size of three, and for two candidates can be solved in $O(|W|^5)$ time, where $|W|$ is the size of the unregistered voter set.*

*Proof.* Let $(E = (C,V), W, \kappa, p, k)$ be a C-Cons-Add instance, where the maximum bundle size $b$ is three, $\kappa$ is symmetric, and $C = \{p,g\}$. This means that all connected components $C_1, \ldots, C_\ell$ of $G_\kappa$ are path or cycles. Furthermore, all bundles only contain voters from one connected component. We define a dynamic program in which each table entry $A[i,s,t]$ contains a solution $W' \subseteq W$ of size $i$, where $\kappa(W') \subseteq V(C_s) \cup \cdots \cup V(C_t)$ and $1 \leq s \leq t \leq \ell$:

(i) If $s = t = j$, then $A[i,s,t] = T[i,1,|V(C_j)|]$, where $T$ is the dynamic program of $C_j$, depending on whether $C_j$ is a path or cycle.

(ii) Otherwise, we build the table as follows:

$$A[d,s,t] = A[d-i,s,s+j] \cup A[i,s+j+1,t], \text{ where}$$
$$i,j = \underset{\substack{0\le i\le d\\ 1\le j\le t-s-1}}{\arg\max}\ \mathsf{gap}(A[d-i,s,s+j]) + \mathsf{gap}(A[i,s+j+1,t]).$$

Each of the table entries $A[i,j,j]$ can be computed in $O(i^2 \cdot |V(C_j)|^3)$ time (see Lemmas 1 and 2) and each of the table entries $A[i,s,t]$ for $s < t$ can be computed in $O(k \cdot \ell)$ time. Since we have $k \cdot \ell^2$ entries, the total running time is

$$\textstyle\sum_{i=1}^{\ell} O(k^2 \cdot |V(C_j)|^3) = O(k^2) \sum_{i=1}^{\ell} O(|V(C_i)|) = O(k^2 \cdot |W|^3).$$ □

From the polynomial-time solvability result of Theorem 2 and by Proposition 1, we obtain the following:

**Corollary 1.** C-Cons-Del *with a symmetric bundling function, maximum bundle size of three, and two candidates can be solved in $O(|V|^5)$ time, where $|V|$ denotes the number of the voters.*

**Corollary 2.** C-Des-Add *and* C-Des-Del *with a symmetric bundling function and maximum bundle size three can be solved in time $O(m \cdot |W|^5)$ and $O(m \cdot |V|^5)$, respectively, where $m$ is the number of candidates, and $|W|$ and $|V|$ are the numbers of unregistered and registered voters, respectively.*

## 5 Controlling Voters with Disjoint Bundles

We have seen in Section 4 that the interaction between the bundles influences the computational complexity of our combinatorial voter control problems. For instance, adding a voter $v$ to the election may lead to adding another voter $v'$ with $v \in \kappa(v)$. This is crucial for the reductions used to prove Theorem 1 and Observation 1. Thus, it would be interesting to know whether the problem becomes tractable if it is not necessary to add two bundles that share some voter(s). More specifically, we are interested in the case where the bundles are disjoint, meaning that we do not need to consider every single voter, but only the bundles as a whole, as it does not matter which voters of a bundle we select.

First, we consider disjoint bundles of size at most two. This is the case for voters who have a partner. If a voter is convinced to participate in or leaves the election, then the partner is convinced to do the same. Note that this is equivalent to having symmetric bundles of size at most two. Bulteau et al. [8, Theorem 6] constructed a linear-time algorithm for C-Cons-Add if the maximum bundle size is two and $\kappa$ is a full-$d$ bundling function (which implies symmetry). We can verify that their algorithm actually works for disjoint bundles of size at most two. Thus, we obtain the following.

**Observation 2.** C-Cons-Add *with a symmetric bundling function and with bundles of size at most two can be solved in $O(|I|)$ time, where $|I|$ is the input size.*

If we want to delete instead of add voter bundles, the problem reduces to finding a special variant of the $f$-Factor problem, which is a generalization of the well-known matching problem and can still be solved in polynomial time [1, 2].

**Theorem 3.** C-Cons-Del *with a symmetric bundling function and with bundles of size at most two can be solved in polynomial time.*

If we drop the restriction on the bundle sizes but still require the bundles to be disjoint, then C-Cons-Add and C-Cons-Del become parameterized intractable with respect to the solution size.

**Theorem 4.** *Parameterized by the solution size $k$,* C-Cons-Add *and* C-Cons-Del *are* W[1]*-hard and* W[2]*-hard respectively, even for disjoint bundles.*

*Proof (with only the construction for the* W[1]*-hardness proof of* C-Cons-Add*).* We provide a parameterized reduction from the W[1]-complete problem Independent Set (parameterized by the "solution size") which, given an undirected graph $G = (V(G), E(G))$ and a natural number $h \in \mathbb{N}$, asks whether $G$ admits a size-$h$ *independent set* $U \subseteq V(G)$, that is, all vertices in $U$ are pairwise non-adjacent. Let $(G, h)$ be an Independent Set instance with $E(G) = \{e_1, \ldots, e_{m-1}\}$ and $V(G) = \{u_1, \ldots, u_n\}$. Without loss of generality, we assume that $G$ is connected and $h \geq 3$. We construct an election $E = (C, V)$ with candidate set $C \coloneqq \{p\} \cup \{g_j \mid e_j \in E(G)\}$. For each edge $e_j \in E$, we construct $h - 1$ registered voters that all have $g_j$ as their favorite candidate. In total, $V$ consists of $(h-1) \cdot (m-1)$ voters.

The unregistered voter set $W$ is constructed as follows: For each vertex $u_i \in V(G)$, add a $p$-voter $p_i$, and for each edge $e_j$ incident with $u_i$, add a $g_j$-voter $a_j^{(i)}$. The voters constructed for each vertex $u_i$ are bundled by the bundling function $\kappa$. More formally, for each $u_i \in V(G)$ and each $e_j \in E(G)$ with $u_i \in e_j$, it holds that

$$\kappa(p_i) = \kappa(a_j^{(i)}) \coloneqq \{p_i\} \cup \{a_{j'}^{(i)} \mid u_i \in e_{j'} \text{ for some } e_{j'} \in E(G)\}.$$

To finalize the construction, we set $k \coloneqq h$. The construction is both a polynomial-time and a parameterized reduction, and all bundles are disjoint. To show the correctness, we note that $p$ can only win if only if her score can be increased to at least $h$ without giving any other candidate more than one more point. The solution corresponds to exactly to a subset of $h$ vertices that are pairwise non-adjacent. The detailed correctness proof and the remaining proof for the W[2]-hardness result can be found in the our technical report [21]. □

For destructive control, it is sufficient to guess a potential defeater $d$ out of $m-1$ possible candidates that will have a higher score than $p$ in the final election and use a greedy strategy similar to the one used for Observation 2 to obtain the following result.

**Theorem 5.** C-Des-Add *and* C-Des-Del *with a symmetric bundling function and disjoint bundles can be solved in $O(m \cdot |I|)$ time, where $|I|$ is the input size and $m$ the number of candidates.*

## 6 Controlling Voters with Unlimited Budget

To analyze election control, it is interesting to know whether a solution exist at all, without bounding its size. Indeed, Bartholdi III et al. [3] already considered the case of unlimited solution size for the constructive candidate control problem. They showed that the problem is already NP-hard, even if the solution size is not bounded. (The non-combinatorial destructive control by adding unlimited amount of candidates is shown to be also NP-hard by Hemaspaandra et al. [19].) In contrast, the non-combinatorial voter control variants are linear-time solvable via simple greedy algorithms [3]. This leads to the question whether the combinatorial structure increases the complexity. To this end, we relax the four problem variants so that the solution can be of arbitrary size and call these problems C-Cons-Add-Unlim, C-Des-Add-Unlim, C-Cons-Del-Unlim and C-Des-Del-Unlim.

First of all, we observe that C-Cons-Del-Unlim becomes trivial if no unique winner is required.

**Lemma 3.** *Every* C-Cons-Del-Unlim *instance is a yes-instance.*

If we consider a voting rule $\mathcal{R}$ that only returns unique winners, then C-Des-Del-Unlim also becomes tractable since we only need to delete all voters.

For the constructive adding voters case, we obtain NP-hardness. The idea for the reduction derives from the W[1]-hardness proof of C-Cons-Add shown by Bulteau et al. [8].

**Lemma 4.** C-Cons-Add-Unlim *is* NP*-hard.*

Lemma 4 immediately implies the following inapproximability result for the optimization variant of C-Cons-Add (denoted as Min-C-Cons-Add), aiming at minimizing the solution size.

**Theorem 6.** *There is no polynomial-time approximation algorithm for* Min-C-Cons-Add*, unless* P = NP*.*

## 7 Conclusion

We extend the study of combinatorial voter control problems introduced by Bulteau et al. [8] and obtain that the destructive control variants tend to be computationally easier than their constructive cousins.

Our research leads to several open questions and further research opportunities. First, we have shown hardness results for the adding candidate case: if the bundling function consists of disjoint cliques, then parameterized by the solution size, C-Cons-Add is W[1]-hard and C-Des-Add is W[2]-hard. If one could also determine the complexity upper bound, that is, under the given restrictions, if C-Cons-Add would be contained in W[1], then this would yield another difference in complexity between the destructive and the constructive variants.

This also leads to the question whether the problem variants in their general setting are not only W[2]-hard, but W[2]-complete.

Second, we have only shown that Min-C-Cons-Add is inapproximable and Min-C-Des-Del is trivially polynomial-time solvable. For the other two problem variants, we do not know whether they can be approximated efficiently or not.

Another open question is whether there are FPT-results for any natural combined parameters. As a starting point, we conjecture that all problem variants can be formulated as a monadic second-order logic formula with length of at most $f(k, b, m)$ (where $k$ is the solution size, $b$ is the maximum bundle size, $m$ is the number of candidates, and $f$ is a computable function). Courcelle and Engelfriet [11] showed that every graph problem expressible as a monadic second-order logic formula $\rho$ can be solved in $g(|\rho|, \omega) \cdot |I|$ time, where $\omega$ is the treewidth of the input graph and $|I|$ is the input size. Our conjecture would provide us with a fixed-parameter tractability result with respect to the solution size, the maximum bundle size, the number of candidates, and the treewidth of our bundling graph $G_\kappa$.

We have studied the Plurality rule exclusively. Thus it is still open which of our results also hold for other voting rules, especially for the Condorcet rule. Since with two candidates, the Condorcet rule is equivalent to the strict majority rule, we can easily adapt some of our results to work for the Condorcet rule as well. Other results (i.e., the Turing reductions) cannot be easily adapted to work for the Condorcet rule.

## References


1. R. P. Anstee. An algorithmic proof of Tutte's $f$-factor theorem. *Journal of Algorithms*, 6(1):112–131, 1985.
2. R. P. Anstee. Minimum vertex weighted deficiency of $(g, f)$-factors: A greedy algorithm. *Discrete Applied Mathematics*, 44(1–3):247–260, 1993.
3. 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.
4. N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52): 43–53, 2009.
5. N. Betzler, R. Bredereck, J. Chen, and R. Niedermeier. Studies in computational aspects of voting. In *The Multivariate Algorithmic Revolution and Beyond*, pages 318–363. Springer, 2012.
6. R. Bredereck, J. Chen, P. Faliszewski, J. Guo, R. Niedermeier, and G. J. Woeginger. Parameterized algorithmics for computational social choice: Nine research challenges. *Tsinghua Science and Technology*, 19(4):358–373, 2014.
7. R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Large-scale election campaigns: Combinatorial shift bribery. *Journal of Artificial Intelligence Research*, 55:603–652, 2016.
8. L. Bulteau, J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Combinatorial voter control in elections. *Theoretical Computer Science*, 589:99–120, 2015.
9. J. Chen. *Exploiting Structure in Computationally Hard Voting Problems*. PhD thesis, Technische Universität Berlin, 2016.

10. J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Elections with few voters: Candidate control can be easy. In *AAAI '15*, pages 2045–2051, 2015.
11. B. Courcelle and J. Engelfriet. *Graph structure and monadic second-order logic: A language-theoretic approach*, volume 138. Cambridge University Press, 2012.
12. R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.
13. 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.
14. P. Faliszewski and J. Rothe. Control and bribery in voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7. Cambridge University Press, 2016.
15. P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.
16. P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.
17. P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Weighted electoral control. *Journal of Artificial Intelligence Research*, 52:507–542, 2015.
18. J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.
19. E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5):255–285, 2007.
20. L. A. Hemaspaandra, R. Lavaee, and C. Menton. Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. *Annals of Mathematics and Artificial Intelligence*, 77(3-4):191–223, 2016.
21. L. Kellerhals, V. Korenwein, P. Zschoche, R. Bredereck, and J. Chen. On the computational complexity of variants of combinatorial voter control in elections. Technical Report arXiv:1701.05108 [cs.MA], arXiv.org, Jan. 2017.
22. H. Liu and D. Zhu. Parameterized complexity of control problems in Maximin election. *Information Processing Letters*, 110(10):383–388, 2010.
23. 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.
24. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.
25. A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *IJCAI '07*, pages 1476–1481, 2007.
26. J. Rothe and L. Schend. Challenges to complexity shields that are supposed to protect elections against manipulation and control: A survey. *Annals of Mathematics and Artificial Intelligence*, 68(1–3):161–193, 2013.