Complexity of Shift Bribery in Committee Elections[*]

Robert Bredereck[1], Piotr Faliszewski[2], Rolf Niedermeier[1],
Nimrod Talmon[1][†]

[1]TU Berlin, Berlin, Germany
{robert.bredereck, rolf.niedermeier}@tu-berlin.de, nimrodtalmon77@gmail.com
[2]AGH University of Science and Technology, Krakow, Poland
faliszew@agh.edu.pl

Abstract

We study the (parameterized) complexity of SHIFT BRIBERY for multiwinner voting rules. We focus on the SNTV, Bloc, k-Borda, and Chamberlin-Courant, as well as on approximate variants of the Chamberlin-Courant rule, since the original rule is NP-hard to compute. We show that SHIFT BRIBERY tends to be significantly harder in the multiwinner setting than in the single-winner one by showing settings where SHIFT BRIBERY is easy in the single-winner cases, but is hard (and hard to approximate) in the multiwinner ones. Moreover, we show that the non-monotonicity of those rules which are based on approximation algorithms for the Chamberlin-Courant rule sometimes affects the complexity of SHIFT BRIBERY.

1   Introduction

We study the complexity of campaign management—modeled as the SHIFT BRIBERY problem—for the case of multiwinner elections. In the SHIFT BRIBERY problem we want to ensure that our candidate is in a winning committee by convincing some of the voters—at a given price—to rank him or her more favorably. In particular, this models campaigns based on direct meetings with voters, in which the campaigner presents positive features of the candidate he or she works for. While the complexity of campaign management is relatively well-studied for single-winner elections, it has not been studied for the multiwinner setting yet (there are, however, studies of manipulation and control for multiwinner elections [2, 29]).

Based on the preferences of the voters, the goal of a multiwinner election is to pick a committee of k candidates. These k candidates might, for example, form the country's next parliament, be a

---

[*]An extended abstract of this paper appears in the Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 16).

[†]Currently at Weizmann Institute of Science.

group of people short-listed for a job opening, or be a set of items a company offers to its customers (see the papers of Lu and Boutilier [25], Skowron et al. [34], and Elkind et al. [17] for a varied description of applications of multiwinner voting). Since the election results can affect the voters and the candidates quite significantly, we expect that they will run campaigns to achieve the most desirable results: a person running for parliament would want to promote her or his political platform; a job candidate would want to convince the HR department of her or his qualities.

We study the standard, ordinal model of voting, where each voter ranks the candidates from the one he or she likes best to the one he or she likes least. We focus on rules that are based either on the Borda scores of the candidates or on their t-Approval scores. Briefly put, if we have m candidates, then a voter gives Borda score $m - 1$ to his or her most preferred candidate, score $m - 2$ to the next one, and so on; a voter gives t-Approval score 1 to each of his or her top-t candidates and score 0 to the other ones.

The most basic multiwinner rules simply pick k candidates with the highest scores (for example, SNTV uses 1-Approval scores, Bloc uses k-Approval scores, and k-Borda uses Borda scores). While such rules may be good for short-listing tasks, they do not seem to perform well for cases where the committee needs to be varied (or represent the voters proportionally; see the work of Elkind et al. [17]). In this case, we may prefer other rules, such as the Chamberlin-Courant family of rules [11], which try to ensure that every voter is represented well by some member of the committee (see the Preliminaries section for an exact definition).

Unfortunately, while the winners of SNTV, Bloc, and k-Borda rules are polynomial-time computable, this is not the case for the Chamberlin-Courant rules (Procaccia et al. [32] and Lu and Boutilier [25] show NP-hardness). We deal with this problem in two ways. First, there are FPT algorithms for computing Chamberlin-Courant winners (for example, for the case of few voters). Second, there are good approximation algorithms (due to Lu and Boutilier [25] and Skowron et al. [35]). Following Caragiannis et al. [10] and Elkind et al. [17], we consider these approximation algorithms as voting rules in their own right (societies may use them in place of the original, hard-to-compute ones).

The idea of the SHIFT BRIBERY problem is as follows. We are given an election and a preferred candidate p, and we want to ensure that p is a winner (in our case, is a member of a winning committee) by shifting him or her forward in some of the votes, at an appropriate cost, without exceeding a given budget. The costs of shifting p correspond to investing resources into convincing the voters that our candidate is of high quality. For example, if a company is choosing which of its products to continue selling, the manager responsible for a given product may wish to prepare a demonstration for the company's higher management. Similarly, a person running for parliament would invest money into meetings with the voters, appropriate leaflets, and so on. Thus, we view SHIFT BRIBERY as a model of (a type of) campaign management.

SHIFT BRIBERY was introduced by Elkind et al. [15, 16], and since then a number of other researchers studied both SHIFT BRIBERY (e.g. Schlotter et al. [33] and Bredereck et al. [7, 9]), and related campaign management problems (e.g. Dorn and Schlotter [13], Baumeister et al. [3], and Faliszewski et al. [20]). Naturally, the problem also resembles other bribery problems, such as the original bribery problem of Faliszewski et al. [19] or those studied by Mattei et al. [28] and Mattei,

Goldsmith, and Klapper [27]. We point the reader to the overview of Faliszewski and Rothe [18] for more details and references.

For single-winner elections, SHIFT BRIBERY is a relatively easy problem. Specifically, it is polynomial-time solvable for the t-Approval rules. For the Borda rule, for which it is NP-hard, there are a good polynomial-time approximation algorithm [15] and exact FPT algorithms [7]. In the multiwinner setting the situation is quite different. The main findings of our research are as follows (see also Table 1 in Section 3):

1. The computational complexity of SHIFT BRIBERY for multiwinner rules strongly depends on the setting. In general, for the cases of few candidates we find FPT algorithms while for the cases where the preferred candidate is shifted by few positions only we find hardness results (even though these cases are often easy in the single-winner setting).

2. The computational complexity for the case of few voters most strongly depends on the underlying scoring rule. Generally, for the rules based on t-Approval scores the complexity of SHIFT BRIBERY tends to be lower than for analogous rules based on Borda scores.

We did not study such multiwinner rules as the STV rule, the Monroe rule [30], or other Approval-based rules (see, e.g., the works of Brams and Kilgour [5] and Aziz et al. [1, 2]), in order to compare our results to those for the single-winner setting, while keeping the considered set of rules small.

2 Preliminaries

Elections and Voting Rules. For each integer n, we set $[n] := \{1, \ldots, n\}$. An election $E = (C, V)$ consists of a set of candidates $C = \{c_1, \ldots, c_m\}$ and a collection of voters $V = (v_1, \ldots, v_n)$. Each voter v is associated with a preference order, i.e., with a ranking of the candidates in decreasing order of appreciation by the voter. For example, if $C = \{c_1, c_2, c_3\}$, then by writing $v\colon c_1 \succ c_2 \succ c_3$ we mean that v likes c_1 best, then c_2, and then c_3. We write $\mathrm{pos}_v(c)$ to denote the position of candidate c in voter v's preference order (e.g., in the preceding example we would have $\mathrm{pos}_v(c_1) = 1$). When we write a subset $A \subseteq C$ of candidates in a description of a preference order, we mean listing all members of A in some fixed, easily computable order. If we put $\overleftarrow{A}$ in a preference order, then we mean listing members of A in the reverse of this fixed order.

Let $E = (C, V)$ be an election with m candidates and n voters. The Borda score of candidate c in the vote of v, $v \in V$, is $\beta_v(c) = m - \mathrm{pos}_v(c)$. The Borda score of c in the election E is $\beta_E(c) = \sum_{v \in V} \beta_v(c)$. The single-winner Borda rule elects the candidate with the highest Borda score (if there are several such candidates, they tie as winners). For each $t \in [m]$, we define the t-Approval score as follows: for a candidate c and voter v, $\alpha_v^t(c) = 1$ if v ranks c among the top t positions and otherwise it is 0; we set $\alpha_E^t(c) = \sum_{v \in V} \alpha_v^t(c)$. We define the single-winner t-Approval rule analogously to the Borda rule.

A multiwinner voting rule $\mathcal{R}$ is a function that, given an election $E = (C, V)$ and an integer $k \in [|C|]$, outputs a set $\mathcal{R}(E, k)$ of k-element subsets of C. Each size-k subset of C is called a *committee* and each member of $\mathcal{R}(E, k)$ is called a *winning committee*.

The most natural task that arises when considering (multiwinner) voting rules is the task of deciding whether a given candidate is among the winners (resp. is part of some winning committee). We will refer to this task as the WINNER DETERMINATION problem. Sometimes, winner determination procedures considered in the literature consider slightly different goals (e.g. computing the score of a winning committee). However, all polynomial-time, FPT, and XP winner determination procedures for the rules we study in this paper can be modified to solve WINNER DETERMINATION.

We consider the following rules (below, $E = (C, V)$ is an election and k is the committee size):

SNTV, *Bloc*, and *k-Borda* compute the score of each candidate and output the committee of k candidates with the highest scores (or all such committees, if there are several). SNTV and Bloc use, respectively, 1-Approval and k-Approval scores, while k-Borda uses Borda scores. For these rules winners can be computed in polynomial time.[1]

Under the *Chamberlin-Courant rules* (the CC rules), for a committee S, a candidate $c \in S$ is a representative of those voters that rank c highest among the members of S. The score of a committee is the sum of the scores that the voters give to their representatives (highest-scoring committees win); *Borda-CC* uses Borda scores, *t-Approval-CC* uses t-Approval scores. WINNER DETERMINATION for CC rules is NP-hard [25, 32], but is in FPT when parameterized by the number of voters or candidates [4].

Greedy-Borda-CC is a $(1 - \frac{1}{e})$-approximation algorithm for the Borda-CC rule, due to Lu and Boutilier [25]. (The approximation is in the sense that the score of the committee output by the algorithm is at least a $1 - \frac{1}{e}$ fraction of the score of the winning committee under Borda-CC.) The algorithm starts with an empty set W and executes k iterations, in each one adding to W the candidate c that maximizes the Borda-CC score of $(W \cup \{c\})$.[2] For example, it always picks a Borda winner in the first iteration. Greedy-Borda-CC always outputs a unique winning committee.

Greedy-Approval-CC works in the same way, but uses t-Approval scores instead of Borda scores. It is a $(1 - \frac{1}{e})$-approximation algorithm for t-Approval-CC. We refer to t-Approval-Greedy-CC for $t = \lceil \frac{m \cdot w(k)}{k} \rceil$ (where w is Lambert's W function; $w(k)$ is $O(\log k)$) as *PTAS-CC*; it is the main part of Skowron et al.'s [35] polynomial-time approximation scheme for Borda-CC.

Parameterized Complexity.   In a parameterized problem, we declare some part of the input as the *parameter* (e.g., the number of voters). A parameterized problem is fixed-parameter tractable (is in FPT) if there is an algorithm that solves it in $f(\rho) \cdot |I|^{O(1)}$ time, where $|I|$ is the size of a given instance encoding, ρ is the value of the parameter, and f is some computable function. There is a hierarchy of classes of hard parameterized problems, FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP. It is widely believed that if a problem is hard for one of the W[·] classes, then it is not in FPT. The notions of hardness and completeness for parameterized classes are defined through parameterized reductions. For this paper, it suffices to use standard polynomial-time many-one reductions that guarantee that the value of the parameter in the problem we reduce to exclusively depends on the value of the parameter of the problem we reduce from. The following problems will be used in our reductions.

---

[1] There may be exponentially many winning committees, but it is easy to compute their score and to check for a subset of candidates if it can be extended to a winning committee.

[2] If there is a tie between several candidates, then we assume that the algorithm breaks it according to a prespecified order.

Definition 1. *An instance of* CLIQUE *consists of a graph $G = (V(G), E(G))$ and an integer h. We ask whether there is a set of h vertices such that there is an edge between each two vertices from the set.*

Definition 2. *An instance of* MULTICOLORED INDEPENDENT SET *consists of a graph $G = (V(G), E(G))$, where each vertex has one of h colors. We ask whether there are h vertices of pairwise-distinct colors such that no two of them are connected by an edge.*

Definition 3. *An instance of* SET COVER *consists of a set U of elements, a family $\mathcal{S}$ of subsets of U, and and integer h. We ask whether there is a subset of h sets from $\mathcal{S}$ whose union is U.*

CLIQUE and MULTICOLORED INDEPENDENT SET are both $W[1]$-complete while SET COVER is $W[2]$-complete (in all cases, the solutions size h is the parameter).

If a parameterized problem can be solved in polynomial time under the assumption that the parameter is constant, then we say that it is in XP. Recall that membership in FPT additionally requires that the degree of the polynomial is a constant independent from the parameter. If a problem is NP-hard even for some constant value of the parameter, then we say that it is para-NP-hard.

For details on parameterized complexity, we point to the books of Cygan et al. [12], Downey and Fellows [14], Flum and Grohe [21], and Niedermeier [31].

3 Shift Bribery

Let $\mathcal{R}$ be a multiwinner rule. In the $\mathcal{R}$-SHIFT BRIBERY problem we are given an election $E = (C, V)$ with m candidates and n voters, a preferred candidate p, a committee size k, voter price functions (see below), and an integer B, the budget. The goal is to ensure that p belongs to at least one winning committee (according to the rule $\mathcal{R}$),[3] and to achieve this goal we are allowed to shift p forward in the preference orders of the voters. However, each voter v has a price function $\pi_v \colon [m] \to \mathbb{N}$, and if we shift p by i positions forward in the vote of v, then we have to pay $\pi_v(i)$. We assume that the price functions are nondecreasing (i.e., it cannot cost less to shift our candidate farther than to shift her or him nearer) and that the cost of not shifting p is zero (i.e., $\pi_v(0) = 0$ for each v). Bredereck et al. [7] have considered several different families of price functions. In this paper we focus on two of them: unit price functions, where for each voter v it holds that $\pi_v(i) = i$, and all-or-nothing price functions, where for each voter v it holds that $\pi_v(i) = q_v$ for each $i > 0$ (where q_v is some voter-dependent value) and $\pi_v(0) = 0$.

A shift action is a vector $(s_1, \ldots, s_n)$ of natural numbers, that for each voter specifies by how many positions to shift p. If $\vec{s} = (s_1, \ldots, s_n)$ is a shift action, then we write $\mathrm{shift}(E, \vec{s})$ to denote the election obtained from E by shifting p an appropriate number of positions forward in each vote. If $\Pi = (\pi_1, \ldots, \pi_n)$ are the price functions of the n voters, then we write $\Pi(\vec{s}) = \sum_{i=1}^{n} \pi_i(s_i)$ to denote the total cost of applying $\vec{s}$. For a shift action $\vec{s}$, we define $\#\vec{s} = \sum_{i=1}^{n} s_i$ and we call it the number of unit shifts in $\vec{s}$.

Formally, we define $\mathcal{R}$-SHIFT BRIBERY as follows.

---

[3] Our approach is a natural extension of the non-unique winner model from the world of single-winner rules. Naturally, one might alternatively require that p is a member of all winning committees or put an even more demanding goal that would involve other candidates. We refer to a brief discussion in the Conclusion section.

	voting rule $\mathcal{R}$	$\mathcal{R}$-WINNER DETERMINATION	$\mathcal{R}$-SHIFT BRIBERY		
			#candidates (m)	#voters (n)	#shifts (s)
single winner	t-Approval	P★	P$^\triangledown$		
	Borda		FPT$^\diamond$	FPT(0/1-pr.), FPT-AS$^\diamond$, and W[1]-h (Thm. 4)	FPT$^\diamond$
multi winner	SNTV	P★	P (Thm. 3)		
	Bloc				
	k-Borda	NP-h♠, FPT(n)$^\heartsuit$, and FPT(m)$^\heartsuit$	FPT (Thm. 1)	FPT(0/1-pr.) (Prop. 1), FPT-AS (Thm. 2), and W[1]-h (Cor. 1+Cor. 2)	W[1]-h (Thm. 5)
	Borda-CC				Para-NP-h♠
	Approval-CC			FPT (Prop. 2)	
	Greedy-Approval-CC	P★			
	PTAS-CC				W[2]-h (Thm. 7)
	Greedy-Borda-CC			W[1]-h (Cor. 2)	

Table 1: Overview of our complexity results for the SHIFT BRIBERY problem (for reference, we also mention the complexity of the WINNER DETERMINATION problem). The results in each cell apply to all voting rules listed in the leftmost column which span the height of the cell. All results are for the case of unit price functions, with the exceptions of those marked as FPT(0/1-pr.), which are for all-or-nothing price functions (many other results extend to other price functions, but we do not list them here). FPT-AS stands for FPT approximation scheme (see Theorem 2). Note that all variants which are W[·]-hard are also in XP. Results marked by $\triangledown$ follow from the work of Elkind et al. [16], by $\diamond$ follow from the work of Bredereck et al. [7], by ♠ follow from the works of Procaccia et al. [32] and Lu and Boutilier [25], by $\heartsuit$ follow from the work of Betzler et al. [4], and by ★ are folk results.

Definition 4. *Let $\mathcal{R}$ be a multiwinner voting rule. An instance I of $\mathcal{R}$-SHIFT BRIBERY consists of an election $E = (C, V)$, a preferred candidate $p \in C$, a committee size k, a collection $\Pi = (\pi_1, \ldots, \pi_n)$ of price functions for the voters, and an integer B, the budget. We ask whether there is a shift action $\vec{s} = (s_1, \ldots, s_n)$ such that:*

1. *$\Pi(\vec{s}) \leq B$, and*

2. *there is a committee $W \in \mathcal{R}(\text{shift}(E, \vec{s}), k)$ such that $p \in W$.*

We refer to such a shift action as a successful shift action*; we write $\text{OPT}(I)$ to denote the cost of the least expensive successful shift action.*

Following Bredereck et al. [7], we consider the most natural parameterizations by the number n of voters, by the number m of candidates, and by the minimum number s of unit shifts in a successful shift action.

Now, we formally defined all central concepts and problems studied in this work, using this, Table 1 summarizes our and some previous results. The reminder of this paper is structured as follows. In Section 4, we present findings applying to the multiwinner context as a whole. In Section 5,

we present specific results for the voting rules SNTV, Bloc, and k-Borda. In Section 6, we present our results for Chamberlin-Courant rules and their approximate variants. We conclude with a final discussion and an outlook in Section 7.

4 General Results

We start our discussion by providing several results that either apply to whole classes of multiwinner rules (including many of those that we focus on) or that are proven using general, easily adaptable techniques. These results form a baseline for our research regarding specific rules.

First, we note that for each of the rules that we study, SHIFT BRIBERY with unit price functions is in FPT when parameterized by the number of candidates. This result follows by applying the standard technique of modeling the problem through an integer linear program and invoking Lenstra's theorem [24]. We believe that, using the MILP technique of Bredereck et al. [8], it is also possible to generalize this result to all-or-nothing price functions.

Note that the following theorem does not mention SNTV and Bloc since, as we will see in the next section, for them the problem is even in P.

Theorem 1. *Parameterized by the number of candidates,* SHIFT BRIBERY *with unit prices is in* FPT *for k-Borda, Approval-CC, Borda-CC, Greedy-Approval-CC, PTAS-CC, and Greedy-Borda-CC.*

In order to prove Theorem 1, we introduce an algorithmic scheme similar to that of Dorn and Schlotter [13] for single-winner SWAP BRIBERY. We will make use of the fact that integer linear programs (ILPs) can be solved in FPT time with respect to the number of (integer) variables (following a famous result by Lenstra [24] which was later improved by Kannan [23] and by Fredman and Tarjan [22]). We first introduce the algorithmic scheme and the basic ILP formulation. Then, we show how to extend the ILP such that the algorithmic scheme works for k-Borda (by proving Lemma 1), for Approval-CC and Borda-CC (by proving Lemma 2), and for Greedy-Approval-CC, PTAS-CC, and Greedy-Borda-CC (by proving Lemma 3).

The idea of the algorithmic scheme is to guess the members of the winning committee $W \subseteq C$, $|W| = k$, $p \in W$, and to verify the guess by an ILP. More precisely, we try all possible winning committees in the outer loop of our algorithm and call the corresponding ILP for each of the (less than 2^m) potential winning committees that contain p. For the round-based rules (Greedy-Approval-CC, PTAS-CC, and Greedy-Borda-CC) we furthermore guess a function $w : [k] \to W$ mapping each "position" in the committee to a specific candidate from W. This allows to specify when each member joined the committee according to the round-based rules and can be realized with an additional factor of $k! \leq m!$ to the running time. For the ease of presentation let W^j denote the set containing the first j members according to the function w, that is, $W^j = \{w(j') \mid 1 \leq j' \leq j\}$.

Observe that there are $m!$ different preference orders, and, by ordering them arbitrarily, we can consider the ith preference order (for $i \in [m!]$).

For each $i \in [m!]$ and $j \in [m!]$ we create an integer variable $S_{i,j}$ which represents the number of voters which vote as the ith preference order in the original election and vote as the jth preference

order in the bribed election. Based on these variables we add the following constraints for each $i \in [m!]$, ensuring that each original vote is turned into exactly one bribed vote:

$$\sum_{j \in [m!]} S_{i,j} = \#(i),$$

where $\#(i)$ denotes the number of voters which vote as the ith preference order in the original election. Then, we add the following constraint, ensuring that the cost of our bribery action does not exceed the budget:

$$\sum_{i \in [m!], j \in [m!]} S_{i,j} \cdot \text{cost}(i, j) \leq B,$$

where $\text{cost}(i, j)$ is the budget needed to transform the ith preference order to the jth preference order (and, for formal correctness, equals $B + 1$ if it is not possible at all by shifting only p).

For each $i \in [m!]$ we create an integer variable N_i which represents the number of voters which vote as the ith preference order in the bribed election. Based on the $S_{i,j}$ variables, we make sure that the N_i variables are correct, by adding, for each $i \in [m!]$, the following constraint:

$$N_i = \sum_{j \in [m!]} S_{j,i}.$$

This describes the basic ILP which will be extended in the proofs of the following lemmas.

Lemma 1. *Parameterized by the number m of candidates, k-Borda* SHIFT BRIBERY *is in FPT.*

Proof. To make p a member of the winning committee for k-Borda we have to ensure that only the other members of the winner committee may have a larger Borda score than p. Hence, for each $c \notin W$, we add the following constraint to the basic ILP, ensuring that, in the bribed election, p has at least as much Borda score (based on the N_i variables) as all candidates that are not in the winning committee:

$$\sum_{i \in [m!]} N_i \cdot \beta_i(p) \geq \sum_{i \in [m!]} N_i \cdot \beta_i(c),$$

where $\beta_i(c)$ is the Borda score of candidate c in the ith preference order.

This finishes the description of the extended ILP. $\qquad\square$

Lemma 2. *Parameterized by the number m of candidates, both Approval-CC* SHIFT BRIBERY *and Borda-CC* SHIFT BRIBERY *are in* FPT.

Proof. To make p a member of the winning committee W for Approval-CC (respectively, Borda-CC) we have to ensure that no other committee has a larger Approval score (respectively, Borda score) than our guessed committee W. Hence, for each other committee W', we add the following constraint to the basic ILP, ensuring that, in the bribed election, the score of W (based on the N_i variables) is at least as high as the score of W':

$$\sum_{i \in [m!]} \phi(i, W) \cdot N_i \geq \sum_{i \in [m!]} \phi(i, W') \cdot N_i,$$

8

where $\phi(i, X)$ is the score given by a voter voting as the ith preference order to the committee X. Note that this can be computed in polynomial-time by selecting the right representative and taking the underlying scoring rule, that is, either Approval or Borda, into account.

This finishes the description of the extended ILP. □

Lemma 3. *Parameterized by the number m of candidates,* SHIFT BRIBERY *is in* FPT *for Greedy-Approval-CC, PTAS-CC, and Greedy-Borda-CC.*

Proof. Since PTAS-CC is a special case of Greedy-Approval-CC it suffices to describe the extension of the ILP for Greedy-Approval-CC and Greedy-Borda-CC.

To make p a member of the winning committee W for Greedy-Approval-CC (respectively, Greedy-Borda-CC) we have to ensure that the candidate $w(j)$ (which joined to the committee in the jth round) maximizes the Approval score (respectively, Borda score) among all possible extensions. Hence, for each round j and each $c \in C \setminus W^j$ we add the following constraint to the basic ILP, ensuring that, in the bribed election, the score of W^j is at least as large as the score of $W^{j-1} \cup \{c\}$:

$$\sum_{i \in [m!]} \phi(i, W^j) \cdot N_i \geq \sum_{i \in [m!]} \phi(i, W^{j-1} \cup \{c\}) \cdot N_i,$$

where $\phi(i, X)$ is the score given by a voter voting as the ith preference order to the committee X.

This finishes the description of the extended ILP. □

As second general result, we note that for the parameterization by the number of voters we can provide a strong, general FPT approximation scheme for candidate-monotone rules. *Candidate monotonicity*, a notion introduced by Elkind et al. [17], requires that if a member of a winning committee is shifted forward in some vote, then this candidate still belongs to some (possibly different) winning committee.

Theorem 2. *Consider parameterization by the number of voters. Let $\mathcal{R}$ be a candidate-monotone multiwinner rule with an* FPT *algorithm for* WINNER DETERMINATION. *Then, for every positive constant number ε there is an* FPT *algorithm that, given an instance I of $\mathcal{R}$-*SHIFT BRIBERY *(for arbitrary price functions), outputs a successful shift action $\vec{s}$ with cost at most $(1 + \varepsilon)\mathrm{OPT}(I)$.*

Proof. Bredereck et al. [7] show an FPT algorithm (parameterized by the number of voters) that, given an instance I of SHIFT BRIBERY and a positive value ε, for each possible shift action $\vec{s} = (s_1, \ldots, s_n)$ tries a shift action $\vec{s}' = (s'_1, \ldots, s'_n)$ such that for each $i \in [n]$ we have $s'_i \geq s_i$, and the cost of $\vec{s}'$ is at most $(1 + \varepsilon)$ greater than that of $\vec{s}$. This algorithm also works for multiwinner rules. □

Among the rules considered in this work, only Greedy-Borda-CC, Greedy-Approval-CC, and PTAS-CC are not candidate-monotone (see the work of Elkind et al. [17] for the argument regarding Greedy-Borda-CC). Thus, the above result applies to all the remaining rules.

For the case of all-or-nothing prices, we can strengthen the above result to an exact FPT algorithm.

Proposition 1. *Consider parameterization by the number of voters. Let $\mathcal{R}$ be a candidate-monotone multiwinner rule with an* FPT *algorithm for* WINNER DETERMINATION. *Then, there is an* FPT *algorithm for* $\mathcal{R}$-SHIFT BRIBERY *with all-or-nothing price functions.*

Proof. Since $\mathcal{R}$ is candidate-monotone and we have all-or-nothing prices, for every vote where we shift the candidate p forward, we can shift p to the top. In effect, it suffices to try all subsets of voters: For each subset check whether shifting p forward in each vote from the subset ensures the victory of p without exceeding the budget. □

Using a very similar approach, we can solve SHIFT BRIBERY for those of our rules which are based on approval scores, even for arbitrary price functions (even the round-based ones). The trick is that, with approval scores, for each voter we either shift our candidate right to the first approved position or we do not shift him or her at all. Thus, again, trying all subsets of voters suffices.

Proposition 2. *There is an* FPT *algorithm for* SHIFT BRIBERY *under Approval-CC, Greedy-Approval-CC, and PTAS-CC, for the parameterization by the number of voters and for arbitrary price functions.*

Finally, using smart brute-force, we provide XP algorithms for SHIFT BRIBERY parameterized either by the number of voters or the number of unit shifts (for rules that can be efficiently computed in the given setting).

Proposition 3. *Consider parameterization by the number of voters. Then, for every multiwinner rule with an* XP *algorithm for* WINNER DETERMINATION*, there is an* XP *algorithm for* SHIFT BRIBERY *and arbitrary price functions.*

Proof. For each voter, we guess the amount which the preferred candidate is shifted by. Since the maximum amount is m, and we have n voters, we have $O(m^n)$ possibilities to check. For each possibility we check if the preferred candidate is a member of a winning committee in XP time. □

Proposition 4. *Consider parameterization by the number of unit shifts. Then, for every multiwinner rule with a polynomial-time algorithm for* WINNER DETERMINATION*, there is an* XP *algorithm for* SHIFT BRIBERY *and arbitrary price functions.*

Proof. The idea of the proof is similar to that behind Proposition 3. Let s be the number of unit shifts that we can perform and let n be the number of voters. We can view a solution as a vector of length at most s, where an entry in the ith position specifies the number of voters in whose preference order we perform the ith unit shifts. We try all $O(n^s)$ such vectors and for each we test if the shift action it defines is within budget and ensures that the preferred candidate is in the winning committee. □

5   SNTV, Bloc, and k-Borda

We now move on to results specific to the voting rules SNTV, Bloc, and k-Borda. These rules pick k candidates with the highest 1-Approval, k-Approval, and Borda scores, respectively, and, so, one

might suspect that the efficient algorithms for corresponding single-winner rules would translate to the multiwinner setting. While this is the case for SNTV and Bloc, for k-Borda the situation is more intricate. As a side effect of our research, we resolve the parameterized complexity of Borda-SHIFT BRIBERY, left open by Bredereck et al. [7].

We first show that SHIFT BRIBERY is polynomial-time solvable for SNTV and Bloc. We use the same algorithm for both SNTV and Bloc. Briefly put, the idea is to guess the final score of the preferred candidate and to compute the set of candidates that have higher scores. Then, it is easy to compute the cheapest way to ensure that all but $k - 1$ of them, where k is the committee size, have smaller score than the guessed score of p, while ensuring that p indeed obtains this guessed score.

Theorem 3. *SNTV-SHIFT BRIBERY and Bloc-SHIFT BRIBERY are both in* P *(for arbitrary price functions).*

Proof. We use the same algorithm for both SNTV and Bloc. Consider an input instance I with an election $E = (C, V)$, where p is the preferred candidate, and where the committee size is k. Our algorithm proceeds as follows.

As first step, we guess the final score that p would have after a successful bribery, denoted by $\mathrm{endscore}(p)$. Since there are only polynomially many possibilities, we can simply branch into all possible values of $\mathrm{endscore}(p)$ to realize the first step. Then, we consider the set $C' \subseteq C$ of those candidates whose score is greater than $\mathrm{endscore}(p)$. It is clear that to ensure that p is in some winning committee, we need to decrease the score of all but $k-1$ candidates from C'. If C' contains at most $k - 1$ candidates, we do not need to decrease the scores of any candidates.

To this end, we sort the candidates in C' by the cost of decreasing their score (by appropriate shifts of p) to be equal to $\mathrm{endscore}(p)$, and pick all of the candidates in C', besides the $k - 1$ most expensive ones. Since for each bribed vote one can decrease the score of exactly one candidate, this defines a shift action. If this shift action does not guarantee that p has score $\mathrm{endscore}(p)$, then we complement it by shifting p to the first approved position in sufficiently many cheapest votes, to ensure that p has score $\mathrm{endscore}(p)$.

If the thus computed shift action is within budget, we accept. Otherwise, we try another guess of $\mathrm{endscore}(p)$. If we try all possibilities without accepting, then we reject. □

The situation for k-Borda is different. SHIFT BRIBERY is NP-hard for Borda due to Elkind et al. [16], so the same holds for k-Borda. We show that Borda-SHIFT BRIBERY is W[1]-hard for parameterization by the number of voters, resolving a previously open case [7]. This result immediately implies the same hardness for all our Borda-based rules.

Theorem 4. *Parameterized by the number of voters, Borda* SHIFT BRIBERY *is* W[1]*-hard (even for unit price functions).*

Proof. We give a parameterized reduction from the MULTICOLORED INDEPENDENT SET problem. Let (G, h) be our input instance. Without loss of generality, we assume that the number of vertices of each color is the same and that there are no edges between vertices of the same color. We write $V(G)$ to denote the set of G's vertices, and $E(G)$ to denote the set of G's edges. Further, for every color $i \in [h]$, we write $V^{(i)} = \{v_1^{(i)}, \ldots, v_q^{(i)}\}$ to denote the set of vertices of color i. For each

vertex v, we write $E(v)$ to denote the set of edges incident to v. For each vertex v, we write $\delta(v)$ to denote its degree, i.e., $\delta(v) = |E(v)|$ and we let $\Delta = \max_{u \in V(G)} \delta(u)$ be the highest degree of a vertex G.

We form an instance of Borda-SHIFT-BRIBERY as follows. We let the candidate set be

$$C = \{p\} \cup V(G) \cup E(G) \cup F(G) \cup D' \cup D'',$$

where $F(G)$, D', and D'' are sets of special dummy candidates. For each vertex v, we let $F(v)$ be a set of $\Delta - \delta(v)$ dummy candidates, and we let $F(G) = \bigcup_{v \in V(G)} F(v)$ and $F(V, -i) = \bigcup_{v \in V(i'), i' \neq i} F(v)$. We will specify D' and D'' later. For each vertex v, we define the partial preference order $S(v)$ to be $v \succ E(v) \succ F(v)$. For each color i, we define $R(i)$ to be a partial preference order that ranks first all members of D', then all vertex candidates of colors other than i, then all edge candidates corresponding to edges that are not incident to a vertex of color i, then all dummy vertices from $F(V, -i)$, and finally all candidates from D''.

We use unit price functions and we set the budget to be $B = h(q + (q-1)\Delta)$. We set D' and D'' to consist of $2B$ dummy candidates each.

We create the following voters:

1. For each color $i \in [h]$, we introduce four voters: voters x_i and x_i' with the following preference orders:

$$x_i \colon S(v_1^{(i)}) \succ S(v_2^{(i)}) \succ \cdots \succ S(v_q^{(i)}) \succ p \succ R(i),$$
$$x_i' \colon \overleftarrow{S(v_q^{(i)})} \succ \overleftarrow{S(v_{q-1}^{(i)})} \succ \cdots \succ \overleftarrow{S(v_1^{(i)})} \succ p \succ R(i),$$

and voters y_i and y_i' whose preference orders are reverses of those of x_i and x_i', respectively, except that candidates from D'' are ranked last in their votes as well.

2. We create a voter z with the preference order

$$z \colon F(G) \succ V(G) \succ E(G) \succ D' \succ p \succ D'',$$

and a voter z' with the preference order that is obtained from that of z by first reversing it, and then shifting each member of $V(G) \cup E(G)$ by one position forward, and shifting p by B positions back.

Let L be the score of p prior to executing any shift actions. The scores of the candidates in our election are as follows: each candidate in $V(G) \cup E(G)$ has score $L + B + 1$, and each candidate in $F(G) \cup D' \cup D''$ has score at most $L + B$.

We show that it is possible to ensure the victory of p in our election by a bribery of cost at most B if and only if there is a multicolored independent set for G of size h.

For the "if" case, we show that if G has a multicolored independent set, then there is a successful shift action of cost B in our election. Let us fix a multicolored independent set for G and, for each color $i \in [h]$, let $v_{s_i}^{(i)}$ be the vertex of color i from this set. For each pair of voters x_i, x_i', we shift p so that in x_i he or she ends up right in front of $v_{s_i+1}^{(i)}$ (or p does not move if $s_i = q$), and in x_i' he

12

or she ends up right in front of $v_{s_i}^{(i)}$. This way, p passes every vertex candidate from $V^{(i)}$ and every edge candidate from $\left(\bigcup_{t \in [q]} E(v_t^{(i)}) \right) \setminus E(v_{s_i}^{(i)})$. This shift action costs B/h for every pair of voters x_i, x_i', so, in total, costs exactly B. Further, clearly, it ensures that p passes every vertex candidate so each of them has score $L + B$. Finally, since we chose vertices from an independent set, every edge candidate also has score at most $L + B$: If p does not pass some edge e between vertices of colors i and j for a pair of voters x_i, x_i', then p certainly passes e in the pair of votes x_j, x_j' because $v_{s_i}^i$ and $v_{s_j}^j$ are not adjacent.

For the "only if" case, we show that if there is a successful shift action for our instance, then there is a multicolored independent set for G. We note that a shift action of cost B gives p score $L + B$. Thus, for the shift action to be successful, it has to cause all candidates in $V(G) \cup E(G)$ to lose a point. We claim that a successful shift bribery has to use exactly $B/h = (q + (q-1)\Delta)$ unit shifts for every pair of voters x_i, x_i'. Why is this so? Let us fix some color $i \in [h]$. Every successful shift action has to decrease the score of every vertex candidate and x_i, x_i' are the only votes where p can pass the vertex candidates from $V^{(i)}$ without exceeding the budget. If we spend less than B/h units of budget on x_i, x_i', then there will be some vertex candidates corresponding to a vertex from $V^{(i)}$ that p did not pass (and, in effect, which does not lose a point), and so p will not be a winner. Thus, we know that a successful shift action spends B/h units of budget on every pair of voters x_i, x_i'. Further, we can assume that for each color i there is a vertex $v_{s_i}^{(i)} \in V^{(i)}$ such that in x_i candidate p is shifted to be right in front of $v_{s_i+1}^{(i)}$ and in x_i' candidate p is shifted to be right in front of $v_{s_i}^{(i)}$. We call such a vertex $v_{s_i}^{(i)}$ *selected*. If for a given pair of voters x_i, x_i' neither of the vertices from $V^{(i)}$ was selected, then there would be some vertex candidate in $V^{(i)}$ that p does not pass. If for some pair of voters x_i, x_i' vertex $v_{s_i}^{(i)}$ is selected, then in this pair of votes p does not pass the edge candidates from $E(v_{s_i}^{(i)})$. However, this means that in a successful shift action the selected vertices form an independent set of G. If two vertices $v_{s_i}^{(i)}$ and $v_{s_j}^{(j)}$ were selected, $i \neq j$, and if there were an edge e connecting them, then p would not pass the candidate e in either of the pairs of votes x_i, x_i' or x_j, x_j'. Since these are the only votes where p can pass e without exceeding the budget, in this case e would have $L + B + 1$ points, p would have $L + B$ points and would lose. □

In effect, we have the following corollary (we discuss other Borda-based rules later).

Corollary 1. *Parameterized by the number of voters, k-Borda-*SHIFT BRIBERY *is* W[1]-*hard.*

Corollary 1 shows that the FPT approximation scheme from Theorem 2 can presumably not be replaced by an FPT algorithm. By Proposition 1, we also know that k-Borda-SHIFT BRIBERY is in FPT for all-or-nothing prices and the parameterization by the number of voters.

The next result is, perhaps, even more surprising than Theorem 4: It turns out that k-Borda-SHIFT BRIBERY is W[1]-hard also for the parameterization by the number of unit shifts, whereas Borda-SHIFT BRIBERY is in FPT. To this end, we describe a parameterized reduction from CLIQUE.

Theorem 5. *Parameterized by the number s of unit shifts, k-Borda* SHIFT BRIBERY *is* W[1]-*hard.*

Proof. We provide a parameterized reduction from the W[1]-complete CLIQUE problem in which we are given a graph G with $V(G) = \{v_1, \ldots, v_n\}$ and $E(G) = \{e_1, \ldots, e_m\}$ and we ask whether there is a set of h pairwise adjacent vertices in G.

Given an instance for the CLIQUE problem, create an instance for k-Borda SHIFT BRIBERY as follows. Set the budget $B := \binom{h}{2} \cdot (2 + h^3)$, use unit price functions, and set the size of the committee $k := n - h + 1$. The candidate set is $C = \{p\} \cup V(G) \cup D(G) \cup F$, where the sets $D(G)$ and F are defined as follows. Let H be a set of B dummy candidates and for each edge e from the graph let $D(e)$ be a set of h^3 dummy candidates. Set $D(G) := \left(\bigcup_{e \in E(G)} D(e) \right) \cup H$. Define F to contain $B + (h - 1)$ dummy candidates.

We form the set of voters as follows:

1. For each edge $e = \{u, v\}$ from G we introduce voter x_e with preference order: $u \succ v \succ D(e) \succ p \succ D(G) \setminus D(e) \succ V(G) \setminus \{u, v\} \succ F$, and voter y_e whose preference order is the reverse of that of x_e with candidates from F shifted to the bottom positions.

2. We introduce two voters, z and z', where z has preference order $V(G) \succ F \succ p \succ D(G)$ and z' has preference order $F \succ p \succ \overleftarrow{V(G)} \succ D(G)$.

All vertex candidates have the same score in this election, and we denote it by L. Candidate p has score $L - (h - 1) - B$, and all remaining candidates have score lower than L (note that we can assume that G has more than $\binom{h}{2}$ edges as otherwise it certainly does not contain a size-h clique). Intuitively, shifting p to the top positions in votes x_e corresponding to a size-h clique is the only way to ensure p's victory

It remains to show the correctness of the construction. More precisely, we show that G contains a clique of size h if and only if there is a successful shift action for our instance of k-Borda-SHIFT BRIBERY.

For the "only if" case, assume that there is a clique if size h in G. Then, a successful bribery can shift p to the front of all x_e voters corresponding to the edges inside this clique. This gives p additional B points and causes each vertex from the clique to lose $h - 1$ points. In effect, there are $n - h$ vertex candidates with score higher than that of p and h vertex candidates with the same score as p. Since all other candidates already had lower scores, p belongs to at least one winning committee.

For the "if" case, note that p can join some winning committee only if at least h vertex candidates lose $h - 1$ points each. Without exceeding the budget, p can pass vertex candidates only in x_e votes. Through simple arithmetic, we see that within a given budget we can shift p to pass some vertex candidates in at most $\binom{h}{2}$ of these votes and, so, in each of them we can shift p to the top position. That is, a successful shift action passes vertices corresponding to $\binom{h}{2}$ edges. This can lead to h candidates losing at least $h - 1$ points each (or, in fact, exactly $h - 1$ points each) only if these edges form a size-h clique. $\qed$

6   Chamberlin-Courant and Its Variants

We now move on to the Chamberlin-Courant (CC) rules and their approximate variants. These rules try to find a committee such that every voter is represented well by some member of the committee. Recall that WINNER DETERMINATION for Borda-CC and Approval-CC is NP-hard but can be

solved efficiently for the approximate variants. To some extend, this difference in the computational complexity is also reflected by our finding for SHIFT BRIBERY.

Note that many results for the CC-based rules (see also Table 1) follow from our results from previous sections. For the parameterizations by the number of candidates, Theorem 1 gives FPT results for all CC-based rules. For the parameterization by the number of voters, by Proposition 2 we have FPT results for Approval-CC, Greedy-Approval-CC, and PTAS-CC. We inherit W[1]-hardness for Borda-CC and Greedy-Borda-CC from Theorem 4, since both rules coincide with the single-winner Borda rule in case of committee size $k = 1$.

Corollary 2. SHIFT BRIBERY *parameterized by the number of voters is* W[1]*-hard for Borda-CC and for Greedy-Borda-CC even for unit price functions.*

By Theorem 2, we have that there is an FPT approximation scheme for Borda-CC. However, since Theorem 2 strongly relies on candidate monotonicity of the rule, it does not apply to Greedy-Borda-CC. Indeed, we believe that there is no constant-factor FPT approximation algorithm for Greedy-Borda-CC-SHIFT BRIBERY (parameterized by the number of voters). So far we could prove this only for the case of weighted elections, i.e., for the case where each voter v has an integer weight w_v and counts as w_v separate voters for computing the result of the election (but not for the computation of the parameter). On the one hand, one could say that using weighted votes goes against the spirit of parameterization by the number of voters and, to some extent, we agree. On the other hand, however, all our FPT results for parameterization by the number of voters (including the FPT approximation scheme) do hold for the weighted case. By a parameterized reduction from the MULTICOLORED CLIQUE problem, we obtain the following.

Theorem 6. *Unless* W[1] = FPT, *Greedy-Borda-CC-*SHIFT BRIBERY *with weighted votes is not α-approximable for any constant α, even in* FPT *time with respect to the number of voters and even for unit price functions.*

Proof. We first prove W[1]-hardness of the problem and then argue that this proof implies the claimed inapproximability result.

We give a reduction from the MULTICOLORED CLIQUE problem for the case of regular graphs which is W[1]-complete for the parameter solution size h (e.g. [26, Lemma 3.2]). To this end, let $G = (V(G), E(G))$ be our input graph and let h be the size of the desired clique (and the number of vertex colors). We use the following notation. For each color $i \in [h]$, we let $V^{(i)} = \{v_1^{(i)}, \ldots, v_n^{(i)}\}$ be the set of vertices from G with color i. For each vertex $v \in V(G)$, we write $E(v)$ to denote the set of edges incident to v. Since G is regular, we let d be the common degree of all the vertices (i.e., for each vertex v, $|E(v)| = d$). For each pair of distinct colors $i, j \in [h]$, $i < j$, we write $E(i, j)$ to denote the set of edges between vertices of color i and vertices of color j.

We make the following observation regarding Greedy-Borda-CC. In each iteration it picks a candidate with the highest score, where this score is computed as follows: Let W be the set of candidates already selected by Greedy-Borda-CC at this point. Consider candidate c and voter v, and let d be the candidate from W that v ranks highest. Voter v gives $\max(0, \text{pos}_v(c) - \text{pos}_v(d))$ points to c (i.e., the number of points by which adding c to W would increase the score of v's

representative). The score of a candidate in a given iteration is the sum of the scores it receives from all the voters.

We form an instance of Greedy-Borda-CC-SHIFT BRIBERY as follows.

The candidates. We let the candidate set be $C = \{b, p, p'\} \cup V(G) \cup E(G) \cup D$, where p is the preferred candidate, p' is p's direct competitor in the sense that either p or p' will be the committee, b is the "bar" candidate (see explanation below), and D is a set of dummy candidates. Throughout the construction we will introduce many dummy candidates and we do not give them special names; at the end of the construction it will be clear that we add only polynomially many of them. We will ensure that b, the bar candidate, is always chosen first into the committee, so—in essence—the scores of all other candidates can be computed relative to b. So when we describe a preference order, we list only top parts of the voters' preference orders, until candidate b. Candidate p is ranked last in every vote in which we do not explicitly require otherwise.

We also use the following notation in the descriptions of the preference orders. For a number L, by writing $[L]$ in a preference order we mean introducing L new dummy candidates that are put in the following positions in this preference order, but that in every other preference order are ranked below b (and, thus, after b is selected receive no points from these voters).

The voters. We introduce the following voters, where N, T_v, T_e, and T_p are four large numbers such that N is much bigger than T_v, T_v is much bigger than T_e, and T_e is much bigger than T_p; we will provide their exact values later. Each voter has weight one unless specified otherwise.

1. For each color $i \in [h]$, we introduce two *vertex-score* voters with the following preference orders:

$$V^{(i)} \succ [N \cdot (T_v - i)] \succ b,$$
$$\overleftarrow{V^{(i)}} \succ [N \cdot (T_v - i)] \succ b,$$

and two *vertex-selection voters* with the following preference orders:

$$V^{(i)} \succ p \succ b,$$
$$\overleftarrow{V^{(i)}} \succ p \succ b.$$

2. For each pair of distinct colors $i, j \in [h]$, $i < j$, we introduce two *edge-score voters* with the following preference orders:

$$E(i, j) \succ [N \cdot (T_e - (i \cdot h + j))] \succ b,$$
$$\overleftarrow{E(i, j)} \succ [N \cdot (T_e - (i \cdot h + j))] \succ b,$$

and two *edge-selection voters* with the following preference orders:

$$E(i, j) \succ p \succ b,$$
$$\overleftarrow{E(i, j)} \succ p \succ b.$$

Each of the edge-selection voters has weight $\omega = 4\binom{h}{2}n(d+1)$ (and these are the only voters with non-unit weights).

16

3. For each color $i \in [h]$ we introduce two *verification voters* with the following preference orders:

$$p \succ v_1^{(i)} \succ E(v_1^{(i)}) \succ \cdots \succ v_n^{(i)} \succ E(v_n^{(i)}) \succ b,$$
$$p \succ v_n^{(i)} \succ \overleftarrow{E(v_n^{(i)})} \succ \cdots \succ v_1^{(i)} \succ \overleftarrow{E(v_1^{(i)})} \succ b.$$

4. We introduce the following two voters, the p/p'-*score voters*, with the following preference orders:

$$p' \succ [N \cdot T_p] \succ b,$$
$$p \succ [N \cdot T_p + h(n+1)(d+1)] \succ b.$$

5. Let H be the total weight of voters introduced so far (clearly, H is polynomially upper-bounded in the input size of the MULTICOLORED CLIQUE instance (G, h)). We introduce $H + 1$ pairs of voters with preference orders $b \succ C \setminus \{b\}$ and $b \succ \overleftarrow{C \setminus \{b\}}$. We refer to these voters as the *bar-score voters*.

We assume that the internal tie-breaking prefers p to p'—we could modify the construction slightly if it were the other way round.

Committee size and budget. We set the committee size to be $k = 1 + h + \binom{h}{2} + 1$. We use unit prices for the voters and we set the budget $B = |V| - h + |E| - \binom{h}{2}$.

We claim that for an appropriate choice of N, T_v, T_e, and T_p it is possible to ensure that p is in a winning committee if and only if there is multicolored size-h clique for G. We now argue why this is the case.

The idea. The general idea is to show that every shift action (even the zero-vector, that means not bribing the voters) of costs at most B leads to a committee that contains

1. the bar vertex b,

2. for each color i one candidate corresponding to a vertex of color i,

3. for each color pair $\{i, j\}, i \neq j$ one candidate corresponding to an edge incident to a vertex of color i and to a vertex of color j

4. candidate p if the selected vertices and edges encode a multicolored clique; otherwise the committee contains p'.

Furthermore, any such combination of vertices and edges can be selected within the given budget, that is, there is a successful shift action if a multicolored clique of size h exists.

Correctness. Observe that due to the bar-score voters, irrespective how we shift p within the budget, Greedy-Borda-CC will first choose b. Thus, from this point on, we compute the score of all candidates relative to b (and, in later rounds, the other selected members of the committee, but there is a limited number of such interactions).

We now describe the next $h + \binom{h}{2} + 1$ rounds, first describing the situation as if p were not shifted, and then indicate how it could be changed with appropriate shifts.

After the first iteration, when b is selected, for each color $i \in [h]$, every vertex in $V^{(i)}$ has score:

$$\underbrace{(2N \cdot (T_v - i) + (n+1))}_{\text{vertex score voters}} + \underbrace{(2n+3)}_{\text{vertex-selection voters}} + \underbrace{((n+1)(d+1))}_{\text{verification voters}}.$$

The points in the first bracket come from the vertex-score voters, in the second bracket from the vertex-selection voters, and in the last bracket from the verification voters. Further, since T_v is much larger than T_e and T_p, every non-vertex candidate has significantly lower score.

Thus, in the next h rounds, for each color $i \in [h]$, Greedy-Borda-CC adds into the committee one vertex candidate of color i. Note that as soon as it picks some vertex candidate of color i, the score of all the other vertex candidates of this color immediately drops by at least $2N \cdot (T_v - i)$ and, so, their score is much too low to be selected.

By shifting candidate p in the vertex-selection votes, for each color $i \in [h]$ and each vertex in $V^{(i)}$ it is possible to ensure that exactly this vertex is selected (it suffices to ensure that every other vertex candidate of this color loses one point due to p passing him or her). The costs of such shifts are at most $|V| - h$ in total.

In other words, we can assume that after these h iterations Greedy-Borda-CC picks one vertex candidate of each color, and that by shift action of cost at most $|V| - h$ it is possible to choose precisely which ones.

In the next $\binom{h}{2}$ iterations, Greedy-Borda-CC picks one edge candidate for each pair of colors. Not counting the verification voters, for each pair of colors $i, j \in [h]$, $i < j$, every edge candidate connecting vertices of colors i and j has score:

$$\underbrace{(N(T_e - (i \cdot h + j)) + |E(i,j)| + 1)}_{\text{edge-score voters}} + \underbrace{(\omega(|E(i,j)| + 2))}_{\text{edge-selection voters}},$$

where the points from the first bracket come from the edge-score voters and the points in the second bracket come from the edge-selection voters. Further, every such candidate receives less than $\frac{\omega}{2}$ points from the verification voters.

Since T_e is much larger than T_p, and since by shifting p forward in the votes of edge-selection voters it is possible to remove ω points from the scores of all but one edge candidate in each $E(i,j)$. Moreover, it is possible to precisely select for each $E(i,j)$ which of its members is added to the committee with a shift action of total cost $|E| - \binom{h}{2}$. Analogously to the case of vertices, note that whenever some candidate from $E(i,j)$ is selected, the other ones lose so many points that they have no chance of being selected in any of the following iterations.

In the final iteration, the algorithm either selects p' or p. Candidate p' has score $N \cdot T_p$, whereas the score of p depends on the vertex and edge candidates that were so far introduced into the committee. If we disregarded all committee members selected after b, p would have score:

$$N \cdot T_p + h(n+1)(d+1).$$

For each color $i \in [h]$, however, p loses $(n+1)(d+1)$ points from the verification voters. This is true since some candidate from $V^{(i)}$ is in the committee, we compute p's score relative to this

vertex candidate and not relative to b. If these were the only points that p loses due to the committee members already selected, then—by tie-breaking— p would win against p'. However, if for some pair of colors $i, j \in [h]$, $i < j$, the committee contains some edge e that connects vertices that are not both in the committee, then p loses at least one more point from the verification voters (either for color i or for color j or for both) because at least one of these verification voters ranks e ahead of all the vertex candidates from the committee. Then p' is selected. This means that p ends up in the committee if and only if due to an appropriate shift action we select vertices and edges corresponding to a multicolored clique. This proves the correctness of the reduction for an appropriate choice of N, T_v, T_e, and T_p, which is discussed next.

The values of N, T_v, T_e, and T_p. While one could pick tight precise values, for the correctness of the proof it suffices to take, say, $T_p = (\binom{h}{2} \cdot |V| \cdot |E|)^3$, $T_e = T_p^3$, $T_v = T_e^3$, and $N = T_v^3$.

Finally, we finally discuss the inapproximability result that is implied by our reduction.

Inapproximability. Observe that, in fact, the above proof gives our inapproximability result. The reason is that for a given constant factor α, we could increase N by the same factor and it would be impossible for p to pass the bar candidate in any of the votes, even if we were to spend α times the necessary budget. In effect, for p to succeed we would still have to find a multicolored clique. $\qquad\square$

For the parameterization by the number of unit shift actions, both Borda-CC and Approval-CC are para-NP-hard due to the hardness of WINNER DETERMINATION.[4] For Greedy-Approval-CC, PTAS-CC, and Greedy-Borda-CC we obtain W[2]-hardness results and inapproximability results.

Theorem 7. *Parameterized by the total number s of unit shifts, SHIFT BRIBERY is W[2]-hard even in case of unit prices for Greedy-Borda-CC, Greedy-Approval-CC, and PTAS-CC. Further, unless W[2] = FPT, in these cases the problem is not α-approximable for any constant α.*

Proof. First, we show the result for Greedy-Approval-CC for t-Approval satisfaction function with $t \geq 3$ (which includes PTAS-CC). Second, we show how the proof ideas can be adapted to obtain the same result for Greedy-Borda-CC.

Greedy-Approcal-CC. We reduce from the SET COVER problem which is W[2]-hard parameterized by the set cover size h. Given an instance $(\mathcal{S}, U, h)$ of SET COVER with $\mathcal{S} = (S_1, \ldots, S_s)$ denoting the given sets over the universe $U = \{u_1, \ldots, u_r\}$, we construct a Greedy-Approval-CC SHIFT BRIBERY instance as follows.

Important candidates. Our election will consist of $2|U| + 2|\mathcal{S}| + 2$ *important candidates*: For each element $u \in U$ we create two *element candidates* $c^-(u)$ and $c^+(u)$. Analogously, we create two *set candidates* $c^-(S)$ and $c^+(S)$ for each set $S \in \mathcal{S}$. Furthermore, we create the preferred candidate p and a candidate p'.

Dummy candidate. For each voter (to be specified later), we introduce up to $(t - 1)$ further dummy candidates. The dummy candidates will not have any chance to be part of the committee,

---

because each of them will only be approved once and there are enough important candidates that have at least two approvals no matter how one bribes the election. However, the dummy candidates intuitively allow some voter to approve any number t', $1 \leq t' \leq t$, of important candidates by approving the desired set of important candidates and some further dummy candidates which are exclusively approved by this voter.

We have basically everything needed to explain the rough idea of the construction.

The idea. In the unbribed election the candidates $c^-(u)$ and $c^-(S)$ for each $u \in U$ and $S \in \mathcal{S}$ together with the candidate p' are elected as committee. The decisive direct effect of successfully bribing the voters will be to decrease the score of at most h candidates $c^-(S)$. Doing this will replace the corresponding $c^-(S)$ with $c^+(S)$ in the committee. Furthermore, each $c^-(u)$ with $u \in S$ for some replaced candidate $c(S)$ will then be replaced by $c^+(u)$. Finally our preferred candidate p will replace candidate p' in the committee if and only if $c^-(u)$ is replaced by the corresponding $c^+(u)$ for each $u \in U$.

Committee size and budget. As already indicated in the description of the idea, we set the budget equal to the size h of the set cover and the committee size to $|\mathcal{S}| + |U| + 1$.

Next, we discuss the voters of the election that allow us to implement the above idea.

Specifying the voters. Observe that for Greedy-Approval-CC SHIFT BRIBERY, specifying the set of approved candidates, the last-ranked approved candidate, that is, the candidate at position t, and the price for shifting the preferred candidate p to position t completely describes the influence of a voter to the unbribed and bribed election: Shifting p to any position greater than t will not change the score of any candidate in any step of Greedy-Approval-CC. Shifting p to some position $\leq t$ will increase the score of p by one and decrease the score of the last-ranked approved candidate by one—independently how far beyond position t we shift p. Hence, for each voter we simply say which important candidates this voter approves, which candidate among the approved candidates is ranked last, and the price of moving p the the first approved position.

The voters. The set of voters contains $|\mathcal{S}|$ many *$\mathcal{S}$-voters*, $|\mathcal{S}| \cdot |U|$ many *$\mathcal{S}$-U-voters*, and $|U|$ many *U-voters*:

- For each set $S \in \mathcal{S}$ there is one S-voter that approves $c^-(S)$ (and some dummy candidates) such that it costs one to disapprove $c^-(S)$ but approve p instead.

- For each set $S \in \mathcal{S}$ and element $u \in U$ there is one S-u-voter that approves $c^-(u)$ only if $u \in S$ and that approves approves $c^+(S)$ (and some dummy candidates) in any case.

- For each element $u \in U$ there is one u-voter that approves p' and $c^+(u)$.

There are further auxiliary voters that allow us to appropriately set the number of approvals for each candidate:

- For each $S_j \in \mathcal{S}$ there are $|\mathcal{S}|^5 \cdot |U|^5 - j$ voters that approve $c^-(S)$ and $c^+(S)$ (and some dummy candidates) and $|U| - 1$ voters that only approve $c^-(S)$ (and some dummy candidates).

- For each $u_i \in U$ there are $|\mathcal{S}|^4 \cdot |U|^4 - i$ voters that approve $c^-(u)$ and $c^+(u)$ (and some dummy candidates) and $|\{S \in \mathcal{S} \mid u_i \in S\}| - 1$ voters that only approve $c^+(u)$ (and some dummy candidates).

- There are $|\mathcal{S}|^2 \cdot |U|^2$ voters that approve p and p' (and some dummy candidates) and $h - 1$ voters that only approve p' (and some dummy candidates).

Except for the $\mathcal{S}$-voters, it is too expensive to bribe some voter with the effect that p is approved instead of some other candidate. This can be easily implemented by putting p more than budget many positions behind the last approved candidate.

This construction can clearly be computed in polynomial time and our parameter, the number of unit shifts, which is upper bounded by the budget, is identical to the set cover size h. Before we prove the correctness of the reduction, let us briefly discuss (properties of) the unbribed election.

Scores, ties, and the unbribed election. First, consider the scores of the candidates in the very first round of the voting rule which are as follows.

- Both, candidate $c^-(S_j)$ and candidate $c^+(S_j)$ have $|\mathcal{S}|^5 \cdot |U|^5 - j + |U|$ approvals.

- Both candidate $c^-(u_i)$ and candidate $c^+(u_i)$ have $|\mathcal{S}|^4 \cdot |U|^4 - i + |\{S \in \mathcal{S} \mid u_i \in S\}|$ approvals.

- Candidate p' has $|\mathcal{S}|^2 \cdot |U|^2 + |U| + h - 1$ approvals.

- Candidate p has $|\mathcal{S}|^2 \cdot |U|^2$ approvals.

We assume that candidate $c^-(\cdot)$ is always preferred to candidate $c^+(\cdot)$ and candidate p' to candidate p by the tie-breaking of Greedy-Approval-CC.[5]

It is easy to verify that in the unbribed election the candidates will join the committee in the following order: $c^-(S_1), c^-(S_2), \ldots, c^-(S_s), c^-(u_1), c^-(u_2), \ldots, c^-(u_r)$ and finally p'. To see this, observe that each pair of candidates $c^-(\cdot)$ and $c^+(\cdot)$ is approved by almost the same set of candidates. As soon as one of $c^-(\cdot)$ and $c^+(\cdot)$ joins the committee, the other loses nearly all approvals and has no chance to join into the committee.

The possible impact of shift actions. The only shift actions with any effect that can be performed within the given budget $B = h$ are to shift p forward in the preference lists of up to h set voters. Let $c^-(S_{j_1}), \ldots, c^-(S_{j_h})$ be the candidates that were originally approved by the bribed voters instead of approving p. We call $\mathcal{S}^* := \{S_{j_1}, S_{j_2}, \ldots, S_{j_h}\}$ the *selected sets*. Applying the corresponding shift actions will decrease the score of each candidate $c^-(S_{j_\ell}), 1 \leq \ell \leq h$, by one and increase the score of p by h. It is easy to verify that, in effect, for each $1 \leq \ell \leq h$, Greedy-Approval-CC will select $c^+(S_{j_\ell})$ instead of $c^-(S_{j_\ell})$ to join the committee. Now, observe that for each $u \in S_{j_\ell}$, there is one voter that approves $c^+(S_{j_\ell})$ and $c^-(u)$. This means that the score of each $c^-(u)$ for $u \in \bigcup_{S \in \mathcal{S}^*} S$ is decreased by at least one after the first $|\mathcal{S}|$ candidates joined the committee. Hence, if $c^+(S_{j_\ell})$ joins the committee instead of $c^+(S_{j_\ell})$, then also $c^+(u)$ joins instead of $c^-(u)$. Finally, observe that, after $s + r$ candidates joined the committee, the score of p' is decreased by the number of candidates $c^+(u)$ that joined the committee instead of $c^-(u)$.

---

[5]The reduction can be adapted to work for any tie-breaking.

Correctness. We show that there is a subset of h sets from $\mathcal{S}$ whose union is U if and only if there is a successful set of shift actions of cost h.

For the "only if" case, assume that there is a set $\mathcal{S}' \subseteq \mathcal{S}$ of h sets whose union is U. Then, bribing the S-voter for each $S \in \mathcal{S}'$ to approve p instead of $c^-(S)$ costs h and successfully makes p a winner: From the above discussion about the impact of shift actions, we can immediately infer that the score of each candidate $c^-(u), u \in U$, is decreased by one and, hence, the score of p' is decreased by $|U|$. Furthermore, the score of p was increased by h. Thus, p has score $|\mathcal{S}|^2 \cdot |U|^2 + h$ whereas p' has score $|\mathcal{S}|^2 \cdot |U|^2 + h - 1$. This means that p joins the committee in the last round of Greedy-Approval-CC.

For the "if" case, assume that there is a set of shift actions with costs h that makes p join the committee. Since p can gain at most h points, p' has to lose at least $|U|$ points. However, the only (important) candidates that are approved together with p' by some voters are the element candidates $c^+(u), u \in U$. To decrease the score of p' by $|U|$, all these candidates $c^+(u), u \in U$, must join the committee instead of $c^-(u), u \in U$. From the above discussion about the impact of shift actions, we can infer that the union of the selected set is U.

Inapproximability. By a slight adaption of the above construction we can conclude (fixed-parameter) inapproximability: First, ensure that even within a budget of $\alpha \cdot B$, one can only afford to bribe the set voters. This can be reached by putting $(\alpha + 1) \cdot B$ dummy candidates between p and the first approved candidate for all but the set voters. Second, introduce another pair of important candidates d and d' and let the set voters additionally approve d. Introduce $|\mathcal{S}|^3 \cdot |U|^3$ voters that approve d and d' (and some dummy candidates) and $|S| - h$ voters that only approve d' (and some dummy candidates). Introduce $|\mathcal{S}| \cdot |U|$ voters that approve p' and d (and some dummy candidates) and further $|\mathcal{S}| \cdot |U|$ voters that only approve d' (and some dummy candidates). Increase the committee size by one.

The first $|\mathcal{S}| + |U|$ rounds of the Greedy-Approval-CC procedure clearly work analogously to the original construction. As long as less than h set voters are bribed, candidate d will join the committee in round $|\mathcal{S}| + |U| + 1$. (We assume that Greedy-Approval-CC prefers d to d' when there is a tie.) Then, candidate d' loses almost all points and has no chance to join the committee and candidate p' loses all additional approvals (introduced by the extension of the construction). That is, the last round works analogously to the original construction. However, if one bribes more than h set voters, then candidate d' will join the committee in round $|\mathcal{S}| + |U| + 1$, p' keeps the additionally introduced approvals, and p has no chance to join the committee in the last round.

It follows that, even with a budget of $\alpha \cdot B$, one can only make p become member of a winning committee if one selects a subset of at most h sets from $\mathcal{S}$ whose union is U.

Greedy-Borda-CC. We give a reduction from the SET COVER problem. The basic idea of the construction is very similar to that in the proof for Greedy-Approval-CC. However, to implement this idea, we also use some concepts from the proof of Theorem 6. To this and, we use the same notational conventions as in the proof of Theorem 6, and we use the bar candidate in the same way.

Given an instance $(\mathcal{S}, U, h)$ of SET COVER with $\mathcal{S} = (S_1, \ldots, S_s)$ denoting the given sets over the universe $U = \{u_1, \ldots, u_r\}$, we construct a Greedy-Borda-CC SHIFT BRIBERY instance as follows.

We form the following set of candidates:

1. We introduce the preferred candidate p, his or her opponent p', and the bar candidate.

2. For each set $S_i \in \mathcal{S}$, we introduce two candidates $c^-(S_i)$ and $c^+(S_i)$.

3. For each element $u_j \in U$, we introduce candidates $c^-(u_j)$ and $c^+(u_j)$.

4. We introduce sufficiently many dummy candidates.

Let N, T_s, T_u, and T_p be some sufficiently large numbers such that N is much larger than T_s, T_s is much larger than T_u, and T_u is much larger than T_p (we will specify their values later). We introduce the following voters:

1. For each set $S_i \in \mathcal{S}$, we introduce two *set-score voters* with preference orders

$$c^-(S_i) \succ c^+(S_i) \succ [N \cdot (T_s - i)] \succ b,$$
$$c^+(S_i) \succ c^-(S_i) \succ [N \cdot (T_s - i)] \succ b.$$

Further, for each set we introduce two *set-selection voters* with preference orders

$$c^-(S_i) \succ p \succ b,$$
$$c^+(S_i) \succ [1] \succ b.$$

2. For each element $u_j \in U$, we introduce two *element-score voters* with preference orders:

$$c^-(u_j) \succ c^+(u_j) \succ [N \cdot (T_u - j)] \succ b,$$
$$c^+(u_j) \succ c^-(u_j) \succ [N \cdot (T_u - j)] \succ b.$$

3. For each $u_j \in U$, we introduce a *verification voter* $c^+(u_j) \succ p' \succ b$.

4. For each element $u_j \in U$, and each set $S_i \in \mathcal{S}$ such that $u_j \in S_i$, we introduce a *covering voter* with preference order:
$$c^+(S_i) \succ c^-(u_j) \succ b.$$

   Further, for each candidate $c \in U \cup \{c^-(S_1), c^+(S_1), \ldots, c^-(S_s), c^+(S_s)\}$, we introduce exactly so many *filler voters* with preference orders of the form $c \succ b$, so that, relative to b, all these candidates receive the same score from the verification, covering, and filler voters (taken together).

5. We introduce two *p/p'-score voters* with preference orders $p' \succ [N \cdot (T_p) + h] \succ b$ and $p \succ [N \cdot (T_p)] \succ b$.

6. Let H be the number of voters introduced so far (clearly, H is polynomially upper-bounded in the size of the input instance). We introduce $H + 1$ pairs of voters with preference orders $b \succ C \setminus \{b\}$ and $b \succ \overleftarrow{C \setminus \{b\}}$. We refer to these voters as the *bar-score voters*.

We set the committee size to be $1 + s + r + 1$, and we set the budget $B = h$. We use unit price functions. The internal tie-breaking is such that p precedes p', for each $S_i \in \mathcal{S}$, $c^-(S_i)$ precedes $c^+(S_i)$, and for each $u_j \in U$, $c^-(u_j)$ precedes $c^+(u_j)$.

The correctness proof works analogous to that for Greedy-Approval-CC. To see this, let us now analyze how Greedy-Borda-CC proceeds on the just-constructed election. As in the proof of Theorem 6, it is clear that in the first iteration it picks b. Due to the values of N and T_s, in the next s iterations, for each $S_i \in \mathcal{S}$, Greedy-Borda-CC either adds $c^-(S_i)$ to the committee or it adds $c^+(S_i)$ to the committee. With a shift action of cost h—by shifting p forward in the votes of the set-selection voters—we can select which h of the $c^+(S_i)$ candidates are introduced into the committee (indeed, we need to introduce h for them to increase p's score—in the final iteration—by h).

In the next r iterations, for each j Greedy-Borda-CC picks either $c^-(u_j)$ or $c^+(u_j)$. It is easy to verify that it picks exactly those $c^+(u_j)$ candidates for which in the preceding iterations it has picked at least one candidate $c^+(S_i)$ such that $u_j \in S_i$.

In the final iteration, Greedy-Borda-CC either picks p or p'. It picks the former one exactly if it managed to pick h candidates from $\mathcal{S}' := \{c^+(S_{j_1}), \ldots, c^+(S_{j_h})\}$ and all candidates $c^+(u_j)$ (since then p gains additional h points and p' loses all points from the verification voters; p wins due to tie-breaking). This happens if and only if we applied a shift action that ensured selection of those h of the $c^+(S_i)$ candidates that correspond to a set cover, that is, $\bigcup_{S \in \mathcal{S}'} S = U$.

To complete the proof for the Greedy-Borda-CC case, we need to pick the values of N, T_s, T_u, and T_p. It is easy to see that the values $T_p = (r \cdot s \cdot h)^3$, $T_u = T_p^3$, $T_s = T_u^3$, and $N = T_s^3$ suffice.

This proves W[2]-hardness of SHIFT-BRIBERY for Greedy-Borda-CC. To see the inapproximability result, one can use an extension to the construction that works analogously to the extension in the proof for Greedy-Approval-CC. $\qquad\square$

7 Conclusion

We studied the complexity of SHIFT BRIBERY for two families of multiwinner rules: SNTV, Bloc, and k-Borda, which pick k best candidates according to appropriate single-winner scoring rules, and the Chamberlin-Courant family of rules and their approximate variants, which focus on providing good representatives. While we have shown low complexity for SNTV and Bloc (just like for the single-winner rules on which they are based), we have shown that SHIFT BRIBERY is significantly harder to solve for k-Borda than for its single-winner variant, Borda. The situation is even more dramatic for the Chamberlin-Courant family of rules, where in addition to W[1]- and W[2]-hardness results, we also obtain inapproximability results.

We focused on the case where we want to ensure a candidate's membership in *some* winning committee; it would also be natural to require membership in *all* winning committees. In fact, all our results hold in this model as well. Below we briefly explain why this is so for the tractability results (for the intractability ones, it requires minor tweaks).

For the tractability results with respect to the *number of candidates* this can be seen as follows.

- For SNTV, Bloc, and k-Borda, we can ensure in our ILP formulations that the score of p is strictly greater than the score of the candidates which are not part of the committee.

- For the round-based rules, the committee is always unique and, hence, our results already apply.

- For the CC rules, we can build upon the maximum matching algorithm of Betzler et al. [4] (trying matchings where p is already matched to one part of the voters, and other ones, where p is not matched at all).

For the tractability results with respect to the *number of voters* or *number of shifts*, our algorithms basically try all bribed elections where p is in at least one winning committee (except for the FPT-AS, where we overshoot; due to monotonicity, this does not hurt). Then, for each bribed election we can adopt the WINNER DETERMINATION algorithm of Betzler et al. [4, Proposition 1] that partitions the voters into groups of voters with the same representative and checks whether p is part of all cheapest matchings of representatives to candidates (basically checking all possible partitions).

Putting an even more demanding bribery goal of involving more than one candidate to become part of the winning committee(s) is left to future studies. Areas of future research also include studying bribery problems for multiwinner settings with partial preference orders and studying multiwinner rules based on the Condorcet criterion. Furthermore, our fixed-parameter algorithms with respect to the parameter number of candidates rely on integer linear programming formulations. It seems challenging to replace these algorithms by direct combinatorial algorithms that give us a better understanding of the problems and potentially better running times. This refers to a general challenge in context of parameterized algorithms for Computation Social Choice problem [6, Key question 1].

8 Acknowledgments

References

[1] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*, pages 784–790. AAAI Press, 2015.

[2] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15)*, pages 107–115. ACM, 2015.

[3] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '12)*, pages 577–584. IFAAMAS, 2012.

[4] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research*, 47:475–519, 2013.

[5] S. Brams and M. Kilgour. Satsifaction approval voting. In R. Fara, D. Leech, and M. Salles, editors, *Voting Power and Procedures: Essays in Honour of Dan Felsenthal and Moshé Machover*, pages 323–346. Springer, 2014.

[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, 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. Long version available as arXiv:1502.01253 [cs.MA].

[8] R. Bredereck, P. Faliszewski, R. Niedermeier, P. Skowron, and N. Talmon. Elections with few candidates: Prices, weights, and covering problems. In *Proceedings of the 4th International Conference on Algorithmic Decision Theory (ADT '15)*, volume 9346 of *LNCS*, pages 414–431, 2015.

[9] 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)*, pages 67–75. ACM, 2015.

[10] I. Caragiannis, C. Kaklamanis, N. Karanikolas, and A. Procaccia. Socially desirable approximations for Dodgson's voting rule. *ACM Transactions on Algorithms*, 10(2):Article No. 6, 2014.

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

[12] M. Cygan, F. V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh. *Parameterized Algorithms*. Springer, 2015.

[13] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.

[14] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013.

[15] E. Elkind and P. Faliszewski. Approximation algorithms for campaign management. In *Proceedings of the 6th International Workshop On Internet And Network Economics (WINE '10)*, volume 6484 of *LNCS*, pages 473–482. Springer, 2010.

[16] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT '09)*, volume 5814 of *LNCS*, pages 299–310. Springer, 2009.

[17] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '14)*, pages 53–60. IFAAMAS, May 2014.

[18] 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, 2015.

[19] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[20] P. Faliszewski, Y. Reisch, J. Rothe, and L. Schend. Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '14)*, pages 1357–1358. IFAAMAS, 2014.

[21] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, 2006.

[22] M. L. Fredman and R. E. Tarjan. Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34(3):596–615, 1987. doi: 10.1145/28869.28874.

[23] R. Kannan. Minkowski's convex body theorem and integer programming. *Mathematics of Operations Research*, 12(3):415–440, 1987.

[24] H. W. Lenstra. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[25] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, pages 280–286. AAAI Press, 2011.

[26] L. Mathieson and S. Szeider. Editing graphs to satisfy degree constraints: A parameterized approach. *Journal of Computer and System Sciences*, 78(1):179–191, 2012.

[27] N. Mattei, J. Goldsmith, and A. Klapper. On the complexity of bribery and manipulation in tournaments with uncertain information. In *Proceedings of the 25th International Florida Artificial Intelligence Research Society Conference (FLAIRS '12)*, pages 549–554. AAAI Press, 2012.

[28] N. Mattei, M. Pini, F. Rossi, and K. Venable. Bribery in voting over combinatorial domains is easy. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '12)*, pages 1407–1408. IFAAMAS, 2012.

[29] 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.

[30] B. Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

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

[32] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[33] I. Schlotter, P. Faliszewski, and E. Elkind. Campaign management under approval-driven voting rules. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI '11)*, pages 726–731. AAAI Press, 2011.

[34] P. Skowron, P. Faliszewski, and J. Lang. Finding a collective set of items: From proportional multi-representation to group recommendation. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI '15)*, pages 2131–2137. AAAI Press, 2015.

[35] P. Skowron, P. Faliszewski, and A. Slinko. Achieving fully proportional representation: Approximability results. *Artificial Intelligence*, 222:67–103, 2015.