

Complexity of Shift Bribery in Committee Elections

Robert Brederick¹, Piotr Faliszewski², Rolf Niedermeier¹, and Nimrod Talmon¹

¹Institut für Softwaretechnik und Theoretische Informatik, TU Berlin, Berlin, Germany
 {robert.bredereck, rolf.niedermeier}@tu-berlin.de, nimrodtalmon77@gmail.com

²AGH 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 rules, 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. 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.

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 (Meir et al. 2008; Aziz et al. 2015b)).

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 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 (2011), Skowron et al. (2015), and Elkind et al. (2014) 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. (2014)). In this case, we may prefer other rules, such as the Chamberlin–Courant family of rules (Chamberlin and Courant 1983), 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. (2008) and Lu and Boutilier (2011) 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 (2011) and Skowron et al. (2015)). Following Caragiannis et al. (2014) and Elkind et al. (2014), 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. (2009; 2010), and since then a number of other researchers studied both SHIFT BRIBERY (e.g. Schlotter et al. (2011) and Bredereck et al. (2014; 2015b)), and related campaign management problems (e.g. Dorn and Schlotter (2012), Baumeister et al. (2012), and Faliszewski et al. (2014)). Naturally, the problem also resembles other bribery problems, such as the original bribery problem of Faliszewski et al. (2009) or those studied by Mattei et al. (2012) and Mattei, Goldsmith, and Klapper (2012). We point the reader to the overview of Faliszewski and Rothe (2015) 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 and for the Borda rule, for which it is NP-hard, there is a good polynomial-time approximation algorithm (Elkind and Faliszewski 2010) and exact FPT algorithms (Bredereck et al. 2014). In the multiwinner setting the situation is quite different. The main findings of our research are as follows (see also Table 1):

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 (Monroe 1995), or other Approval-based rules (see, e.g., the works of Brams and Kilgour (2014) and Aziz et al. (2015a; 2015b)), in order to compare our results to those for the single-winner setting, while keeping the considered set of rules small. Due to space constraints, most proofs are deferred to a full version of the paper.

Preliminaries

Elections and Voting Rules. For each integer n , we set $[n] := \{1, \dots, n\}$. An election $E = (C, V)$ consists of a set of candidates $C = \{c_1, \dots, c_m\}$ and a collection of voters $V = (v_1, \dots, v_n)$. 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 $v: c_1 \succ c_2 \succ c_3$ we mean that v likes c_1 best, then c_2 , and then c_3 . We write $\text{pos}_v(c)$ to denote the position of candidate c in voter v 's preference order (e.g., in the preceding example we

would have $\text{pos}_v(c_1) = 1$). 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 $\beta_v(c) = m - \text{pos}_v(c)$. 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*. 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 (Procaccia, Rosenschein, and Zohar 2008; Lu and Boutilier 2011), but is in FPT when parameterized by the number of voters or candidates (Betzler, Slinko, and Uhlmann 2013).

Greedy-Borda-CC is a $(1 - \frac{1}{e})$ -approximation algorithm for the Borda-CC rule, due to Lu and Boutilier (2011). (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 -

¹There 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.

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

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 (2015) 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 time $f(\rho) \cdot |I|^{O(1)}$, 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, $\text{FPT} \subseteq \text{W}[1] \subseteq \text{W}[2] \subseteq \dots \subseteq \text{XP}$. It is widely believed that if a problem is hard for one of the $\text{W}[\cdot]$ 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.

Definition 1. An instance of MULTICOLORED INDEPENDENT SET consists of a graph where each vertex has one of h colors. We ask if there are h vertices of pairwise-distinct colors such that no two of them are connected by an edge.

Parameterized by the solution size h , MULTICOLORED INDEPENDENT SET is $\text{W}[1]$ -complete. 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. 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. (2015), Downey and Fellows (2013), Flum and Grohe (2006), and Niedermeier (2006).

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 $\pi_v: [m] \rightarrow \mathbb{N}$, 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). Brederick et al. (2014) 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

³Our 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.

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, \dots, s_n) of natural numbers, that for each voter specifies by how many positions to shift p . If $\vec{s} = (s_1, \dots, s_n)$ is a shift action, then we write $\text{shift}(E, \vec{s})$ to denote the election obtained from E by shifting p an appropriate number of positions forward in each vote. If $\Pi = (\pi_1, \dots, \pi_n)$ 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.

Definition 2. 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 $\Pi = (\pi_1, \dots, \pi_n)$ of price functions for the voters, and an integer B , the budget. We ask if there is a shift action $\vec{s} = (s_1, \dots, s_n)$ such that: (a) $\Pi(\vec{s}) \leq B$ and (b) 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 Brederick et al. (2014), we consider the most natural parameterizations by the number n of the voters, by the number m of the candidates, and by the minimum number s of unit shifts in a successful shift action.

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. In this section, when we say that a given rule has a winner determination procedure with a given complexity, we mean a procedure with that complexity that tests whether a given candidate is in some winning committee. All polynomial-time, FPT, and XP winner determination procedures for the rules we study in this paper can be modified to answer such queries.

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 (Lenstra 1983). We believe that, using the MILP technique of Brederick et al. (2015a), it is also possible to generalize this result to all-or-nothing price functions.

Theorem 1. For the parameterization 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.

(Theorem 1 does not mention SNTV and Bloc since, as we will see in the next section, for them the problem is in P.)

Due to their round-based nature, this result is somewhat intricate for the case of Greedy-Approval-CC, PTAS-CC, and Greedy-Borda-CC. For these rules, we first guess how the rounds proceed and only then use Lenstra's algorithm.

voting rule \mathcal{R}		\mathcal{R} -WINNER DE- TERMINATION	\mathcal{R} -SHIFT BRIBERY		
			#candidates (m)	#voters (n)	#shifts (s)
single winner	t -Approval	P^\star	P^∇		
	Borda		FPT $^\diamond$	FPT(0/1-pr.), FPT-AS $^\diamond$, and W[1]-h (Thm. 4)	FPT $^\diamond$
multi winner	SNTV	P^\star	P (Thm. 3)		
	Bloc		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)
	k -Borda			FPT (Prop. 2)	Para-NP-h $^\spadesuit$
	Borda-CC	NP-h $^\spadesuit$, FPT(n) $^\heartsuit$, and FPT(m) $^\heartsuit$			
	Approval-CC			P^\star	W[2]-h (Thm. 7)
	Greedy-Approval-CC				
	PTAS-CC				
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 the voting rules listed in the leftmost column which span the height of the cell. All the 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 (cf. Theorem 2). Note that all variants which are W[·]-hard are also in XP. Results marked by ∇ follow from the work of Elkind et al. (2009), by $^\diamond$ follow from the work of Bredecek et al. (2014), by $^\spadesuit$ follow from the works of Procaccia et al. (2008) and Lu and Boutilier (2011), by $^\heartsuit$ follow from the work of Betzler et al. (2013), and by * are folk results.

Second, 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. (2014), 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 winner determination procedure. 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 $(1 + \varepsilon)\text{OPT}(I)$.*

Proof. Bredecek et al. (2014) 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 $\vec{s} = (s_1, \dots, s_n)$ tries a shift action $\vec{s}' = (s'_1, \dots, s'_n)$ 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. \square

Among our rules, only Greedy-Borda-CC, Greedy-Approval-CC, and PTAS-CC are not candidate-monotone (see the work of Elkind et al. (2014) 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 winner determination procedure. 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, it suffices to try all subsets of voters: for each subset, for each vote from it, shift p to the top. If this makes p a winner without exceeding the budget, then accept. \square

A similar approach works for those of our rules that are based on approval scores, even for arbitrary price functions: with approval scores, for each voter we either shift our candidate exactly to the first approved position or we do not shift him or her at all. Thus, 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 algorithm, 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. For every multiwinner rule with an XP winner determination procedure, there is an XP algorithm for SHIFT BRIBERY and arbitrary price functions.*

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

SNTV, Bloc, and k -Borda

We now move on to results specific to 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 Brederick et al. (2014).

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).*

The situation for k -Borda is different. SHIFT BRIBERY is NP-hard for Borda due to Elkind et al. (2009) 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. 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 $V^{(i)} = \{v_1^{(i)}, \dots, v_q^{(i)}\}$ 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 the 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.

First, for each color $i \in [h]$, we introduce four voters: voters x_i and x'_i with the following preference orders:

$$\begin{aligned} x_i: S(v_1^{(i)}) \succ S(v_2^{(i)}) \succ \dots \succ S(v_q^{(i)}) \succ p \succ R(i), \\ x'_i: \overleftarrow{S(v_q^{(i)})} \succ \overleftarrow{S(v_{q-1}^{(i)})} \succ \dots \succ \overleftarrow{S(v_1^{(i)})} \succ p \succ R(i), \end{aligned}$$

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.

Second, we create a voter z with the preference order

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

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 .

First, 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 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 $(\bigcup_{t \in [q]} E(v_t^{(i)})) \setminus E(v_{s_i}^{(i)})$.

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).

Second, 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 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. \square

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.

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

Chamberlin–Courant and Its Variants

We now move on to the CC rules and their approximate variants. For the parameterizations by the number of candidates, Theorem 1 gives FPT results. For the parameterization by the number of voters, by Proposition 2 we have FPT results for (Greedy)-Approval-CC and PTAS-CC. We inherit $W[1]$ -hardness for (Greedy)-Borda-CC from Theorem 4.

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 $W[1] = \text{FPT}$, 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 (even for unit price functions).*

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 their 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 for Greedy-Borda-CC, Greedy-Approval-CC, and PTAS-CC (even in case of unit prices). Further, unless $W[2] = \text{FPT}$, in these cases the problem is not α -approximable for any constant α .*

Conclusions

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).

Tractability results with respect to the number of candidates. 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

⁴The literature speaks of hardness of computing the score of a winning committee, but one can show that deciding whether a given candidate is in some winning committee is NP-hard as well.

unique and, hence, our results already apply. For the CC rules, we can build upon the maximum matching algorithm of Betzler, Slinko, and Uhlmann (2013) (trying matchings where p is already matched to one part of the voters, and other ones, where p is not matched at all).

Tractability results with respect to the number of voters or the 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 algorithm of Betzler, Slinko, and Uhlmann (2013) that partitions the voters into groups of voters with the same representative and check whether p is part of all cheapest matchings of representatives to candidates.

Areas of future research include studying bribery problems for multiwinner settings with partial preference orders and studying multiwinner rules based on the Condorcet criterion.

Acknowledgments

The authors were supported in part by the DFG project PAWS (NI 369/10), the NCN project DEC-2012/06/M/ST1/00358. Nimrod Talmon was supported by the DFG Research Training Group MDS (GRK 1408). Piotr Faliszewski's visit to TU Berlin was supported by the COST action IC1205.

References

- Aziz, H.; Brill, M.; Conitzer, V.; Elkind, E.; Freeman, R.; and Walsh, T. 2015a. Justified representation in approval-based committee voting. In *Proc. of AAAI '15*, 784–790.
- Aziz, H.; Gaspers, S.; Gudmundsson, J.; Mackenzie, S.; Mattei, N.; and Walsh, T. 2015b. Computational aspects of multi-winner approval voting. In *Proc. of AAMAS '15*, 107–115.
- Baumeister, D.; Faliszewski, P.; Lang, J.; and Rothe, J. 2012. Campaigns for lazy voters: Truncated ballots. In *Proc. of AAMAS '12*, 577–584. IFAAMAS.
- Betzler, N.; Slinko, A.; and Uhlmann, J. 2013. On the computation of fully proportional representation. *Journal of Artificial Intelligence Research* 47:475–519.
- Brams, S., and Kilgour, M. 2014. Satisfaction approval voting. In Fara, R.; Leech, D.; and Salles, M., eds., *Voting Power and Procedures: Essays in Honour of Dan Felsenthal and Moshé Machover*. Springer. 323–346.
- Bredereck, R.; Chen, J.; Faliszewski, P.; Nichterlein, A.; and Niedermeier, R. 2014. Prices matter for the parameterized complexity of shift bribery. In *Proc. of AAAI '14*, 1398–1404.
- Bredereck, R.; Faliszewski, P.; Niedermeier, R.; Skowron, P.; and Talmon, N. 2015a. Elections with few candidates: Prices, weights, and covering problems. In *Proc. of ADT '15*, 414–431.
- Bredereck, R.; Faliszewski, P.; Niedermeier, R.; and Talmon, N. 2015b. Large-scale election campaigns: Combinatorial shift bribery. In *Proc. of AAMAS '15*, 67–75.
- Caragiannis, I.; Kaklamanis, C.; Karanikolas, N.; and Procaccia, A. 2014. Socially desirable approximations for Dodgson's voting rule. *ACM Transactions on Algorithms* 10(2):Article No. 6.
- Chamberlin, B., and Courant, P. 1983. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review* 77(3):718–733.
- Cygan, M.; Fomin, F. V.; Kowalik, L.; Lokshantov, D.; Marx, D.; Pilipczuk, M.; Pilipczuk, M.; and Saurabh, S. 2015. *Parameterized Algorithms*. Springer.
- Dorn, B., and Schlotter, I. 2012. Multivariate complexity analysis of swap bribery. *Algorithmica* 64(1):126–151.
- Downey, R. G., and Fellows, M. R. 2013. *Fundamentals of Parameterized Complexity*. Springer.
- Elkind, E., and Faliszewski, P. 2010. Approximation algorithms for campaign management. In *Proc. of WINE '10*, volume 6484 of *LNCS*, 473–482. Springer.
- Elkind, E.; Faliszewski, P.; Skowron, P.; and Slinko, A. 2014. Properties of multiwinner voting rules. In *Proc. of AAMAS '14*, 53–60. IFAAMAS.
- Elkind, E.; Faliszewski, P.; and Slinko, A. 2009. Swap bribery. In *Proc. of SAGT '09*, volume 5814 of *LNCS*, 299–310. Springer.
- Faliszewski, P., and Rothe, J. 2015. Control and bribery in voting. In Brandt, F.; Conitzer, V.; Endriss, U.; Lang, J.; and Procaccia, A. D., eds., *Handbook of Computational Social Choice*. Cambridge University Press. chapter 7.
- Faliszewski, P.; Reisch, Y.; Rothe, J.; and Schend, L. 2014. Complexity of manipulation, bribery, and campaign management in Bucklin and fallback voting. In *Proc. of AAMAS '14*, 1357–1358. IFAAMAS.
- Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. A. 2009. How hard is bribery in elections? *Journal of Artificial Intelligence Research* 35:485–532.
- Flum, J., and Grohe, M. 2006. *Parameterized Complexity Theory*. Springer.
- Lenstra, H. W. 1983. Integer programming with a fixed number of variables. *Mathematics of Operations Research* 8(4):538–548.
- Lu, T., and Boutilier, C. 2011. Budgeted social choice: From consensus to personalized decision making. In *Proc. of IJCAI '11*, 280–286. AAAI Press.
- Mattei, N.; Pini, M. S.; Rossi, F.; and Venable, K. B. 2012. Bribery in voting over combinatorial domains is easy. In *Proc. of ISAAC '12*.
- Mattei, N.; Goldsmith, J.; and Klapper, A. 2012. On the complexity of bribery and manipulation in tournaments with uncertain information. In *Proc. of FLAIRS '12*, 549–554. AAAI Press.
- Meir, R.; Procaccia, A.; Rosenschein, J.; and Zohar, A. 2008. The complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research* 33:149–178.
- Monroe, B. 1995. Fully proportional representation. *American Political Science Review* 89(4):925–940.
- Niedermeier, R. 2006. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press.
- Procaccia, A.; Rosenschein, J.; and Zohar, A. 2008. On the complexity of achieving proportional representation. *Social Choice and Welfare* 30(3):353–362.
- Schlotter, I.; Faliszewski, P.; and Elkind, E. 2011. Campaign management under approval-driven voting rules. In *Proc. of AAAI '11*, 726–731. AAAI Press.
- Skowron, P.; Faliszewski, P.; and Lang, J. 2015. Finding a collective set of items: From proportional multirepresentation to group recommendation. In *Proc. of AAAI '15*, 2131–2137. AAAI Press.
- Skowron, P.; Faliszewski, P.; and Slinko, A. 2015. Achieving fully proportional representation: Approximability results. *Artificial Intelligence* 222:67–103.