Large-Scale Election Campaigns: Combinatorial Shift Bribery

Robert Bredereck*
TU Berlin, Berlin, Germany
robert.bredereck@tu-berlin.de

Piotr Faliszewski†
AGH University,
Krakow, Poland
faliszew@agh.edu.pl

Rolf Niedermeier
TU Berlin, Berlin, Germany
rolf.niedermeier@tu-berlin.de

Nimrod Talmon‡
TU Berlin, Berlin, Germany
nimrodtalmon77@gmail.com

ABSTRACT

We study the complexity of a combinatorial variant of the Shift Bribery problem in elections. In the standard Shift Bribery problem, we are given an election where each voter has a preference order over the candidate set and where an outside agent, the briber, can pay each voter to rank the briber's favorite candidate a given number of positions higher. The goal is to ensure the victory of the briber's preferred candidate. The combinatorial variant of the problem, introduced in this paper, models settings where it is possible to affect the position of the preferred candidate in multiple votes, either positively or negatively, with a single bribery action. This variant of the problem is particularly interesting in the context of large-scale campaign management problems (which, from the technical side, are modeled as bribery problems). We show that, in general, the combinatorial variant of the problem is highly intractable (NP-hard, hard in the parameterized sense, and hard to approximate), but we provide some (approximation) algorithms for natural restricted cases.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent systems*

Keywords

campaign management, preference aggregation, voting, computational complexity, parameterized complexity, fixed-parameter tractability, approximation algorithms

*Supported by the DFG, project PAWS (NI 369/10).

†Supported by the DFG, project PAWS (NI 369/10) and by AGH University grant 11.11.230.124 (statutory research).

‡Supported by the DFG, RTG MDS (GRK 1408).

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015), Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.*

1. INTRODUCTION

We study the computational complexity of election campaign management, for the case where campaign actions (such as airing a TV advertisement, launching a web-based campaign, or organizing meetings with voters) may have large-scale effects which affect multiple voters. Further, we are interested in settings where these actions can have both positive effects (for example, some voters may choose to rank the promoted candidate higher because they find arguments presented in a given advertisement appealing) as well as negative ones (for example, because some other voters find the advertisement to be too aggressive). Thus, in our settings, the two major issues faced by a campaign manager are (a) choosing actions that positively affect as many voters as possible, and (b) balancing the negative effects of campaigning actions (for example, by concentrating these negative effects on voters who disregard the promoted candidate anyway).

Our research falls within the field of computational social choice, a subarea of multiagent systems. We use the standard election model, where we are given a set C of candidates and a collection V of voters, each represented by his or her preference order (that is, a ranking of the candidates from the most preferred one to the least desired one). That is, we assume that we know the preferences of all the voters. While having such perfect knowledge is impossible in practice, this assumption is a convenient simplification modeling the information we have from preelection polls.

We consider two voting rules, the Plurality rule (where we pick the candidate who is ranked first by most voters) and the Borda rule (where each candidate c gets from each voter v as many points as there are candidates that v prefers c to, and we pick the candidate with the most points). We chose these rules because the Plurality rule is the most widespread rule in practice and because the Borda rule is very well-studied in the context of campaign management.

Within computational social choice, the term *campaign management* (introduced by Elkind et al. [11, 12]) is an alternative name for the bribery family of problems (introduced by Faliszewski et al. [13]) that focuses on modeling actions available during election campaigns: as a result of money spent by a campaign manager, some of the voters change their votes. In this paper we focus on campaign management as modeled by the Shift Bribery problem [5, 11, 12]. In Shift Bribery we have a candidate p whom we want to win, for each voter v we have a price $\pi_v(i)$ for which this voter is willing to shift p forward by i positions in his or her preference order[1], and we ask for the lowest cost of ensuring that p is a winner (see "Related Work" below for references to other campaign management problems).

Shift Bribery has one major drawback as a model for campaign management: it is incapable of modeling large-scale effects of campaign actions. In particular, if one puts out a TV spot promoting a given candidate, then some voters will react positively and rank the candidate higher, some will be oblivious to it, and some will react negatively, by ranking the candidate lower. Shift Bribery cannot model such correlated effects. In this paper we introduce and study the Combinatorial Shift Bribery problem, allowing campaign actions to have effects, positive or negative, on whole groups of voters.

We are interested in seeing how such a more realistic model of campaign management affects the complexity of the problem. Indeed, Shift Bribery is, computationally, a very well-behaved problem. For example, for the Plurality rule it is solvable in polynomial time and for the Borda rule it is NP-complete [12], but there is a polynomial-time 2-approximation algorithm [11, 12] and there are fixed-parameter tractable (FPT) algorithms, either exact or capable of finding solutions arbitrarily close to the optimal ones [5]. We ask to what extent do we retain these good computational properties when we allow large-scale effects. The results are surprising both positively and negatively:

1. Combinatorial Shift Bribery becomes both NP-complete and W[1]-hard even for the Plurality rule, even for very restrictive choice of parameters, even if the correlated effects of particular campaign actions are limited to at most two voters. These hardness results also imply that good, general approximation algorithms do not exist, and are particularly strong when we allow negative effects of some campaign actions.

2. In spite of the above, it is still possible to derive relatively good (approximation) algorithms, both for the Plurality rule and for the Borda rule, provided that we restrict the effects of the campaign actions to be only positive and to either only involve few voters each, or to only involve groups of consecutive voters.

Due to space constraints several proof details are deferred to a full version of the paper.

Related Work. Our work builds on top of two main research ideas: first, on studying campaign management/bribery problems, and second, on studying combinatorial variants of election problems.

The study of the computational complexity of bribery in elections was initiated by Faliszewski et al. [13], and continued by a number of researchers [14, 18, 21, 22]. Elkind et al. [11, 12] realized that the formalism of election bribery problems is useful from the point of view of planning election campaigns. In particular, they defined the Swap Bribery problem and its restricted variant, Shift Bribery. In the former it is possible, at a given price, to swap any two adjacent candidates in a given vote. In the latter, we are only allowed to shift the preferred candidate forward. Various problems, modeling different flavors of campaign management, have been studied, including, for example, the possibility to alter the number of approved/ranked candidates [2, 15, 26]. Different (positive) applications of bribery problems have been revealed. For example, in the Margin of Victory view of the problem, the goal of the briber is to prevent some candidate from winning; if it is possible to do so at low cost, then this suggests that the election could have been tampered with [6, 20, 27].

The most related works are those of Elkind et al. [11, 12], Bredereck et al. [5], and Dorn and Schlotter [9]. The former ones study Shift Bribery, which we generalize (with the work of Bredereck et al. [5] focusing on parameterized complexity), whereas the work of Dorn and Schlotter [9] pioneers the use of parameterized complexity analysis for bribery problems.

Our work is largely inspired by the work of Chen et al. [7], who introduced and studied a combinatorial variant of voter control (control is a very well-studied topic in computational social choice, initiated by Bartholdi et al. [1]). We stress that by combinatorial election problems, in this case, we do not mean the large line of work on modeling combinatorial candidate spaces (see, for example, the papers [3, 8, 22]), but use the same jargon from the mentioned literature with the term "Combinatorial" refering to the "combinations" of voters affected by each bribery action.

2. PRELIMINARIES

Elections. An election $E = (C, V)$ consists of a set $C = \{c_1, \dots, c_m\}$ of candidates and a collection $V = (v_1, \dots, v_n)$ of voters. Each voter is represented through his or her preference order, that is, a linear ranking of the candidates from the most preferred one to the least preferred one. For example, if $C = \{c_1, c_2, c_3\}$, then voter v_1 may have preference order $v_1 \colon c_1 \succ c_2 \succ c_3$ to indicate that he or she likes c_1 best, then c_2, and then c_3. We assume that an arbitrary (but fixed) canonical order over the set of candidates exists (for example, one could order the candidates lexicographically, by their names). If A is some subset of candidates, then writing $\overrightarrow{A}$ within a preference order means listing the candidates from A in this canonical order.

Voting Rules. A voting rule $\mathcal{R}$ is a function that, given an election $E = (C, V)$, outputs a set $\mathcal{R}(E) \subseteq C$ of (tied) election winners. We consider two election rules, the Plurality rule and the Borda rule. Each of these rules assigns points to candidates and outputs those with the highest score. Under the Plurality rule, each candidate receives one point for each voter ranking him or her first. Under the Borda rule, each candidate receives i points for each voter that preferring this candidate to exactly i other ones.

We use the nonunique-winner model. That is, all the candidates output by a given voting rule are viewed as equally successful winners (in practice, of course, one has to use some sort of a tie-breaking rule to resolve the situation, but disregarding ties simplifies the analysis; however, an interested reader should consult papers on the effects of tie-breaking on the complexity of election problems [24, 25]).

[1]Of course, this price does not necessarily reflect a direct money transfer to the voter, but rather the cost of convincing the voter to change his or her mind.

Parameterized Complexity. We assume that the familiarity with standard notions regarding algorithms and complexity theory, but briefly review notions regarding parameterized complexity theory [10, 16, 23].

In parameterized complexity theory, we measure the complexity of a given problem with respect to both the input size and a particular *parameter* of the problem. Typical parameters for election problems include the number of candidates, the number of voters, and the solution size (for example, the number of campaign actions one can perform). We say that a parameterized problem is *fixed-parameter tractable* (is in FPT) if there is an algorithm that, given an input instance I with parameter k, solves the problem in time $g(k)|I|^{O(1)}$, where g is some computable function and $|I|$ is the length of the encoding of I. There is also a hierarchy of hardness classes for parameterized problems, of which the two most important levels are formed by the classes W[1] and W[2]. The most convenient way of defining these classes is through an appropriate reduction notion and their complete problems. Specifically, we say that a parameterized problem A reduces to a parameterized problem B if there are two computable functions, g and g', with the following properties: given an instance I of A with parameter k, $g(I)$ outputs in FPT time an instance I' of B with parameter $k' \leq g'(k)$, such that I is a "yes"-instance of A if and only if I' is a "yes"-instance of B. In other words, g is a standard many-one reduction from A to B, which is allowed to run in FPT time, but such that the parameter parameter of the output instance must be upper-bounded by a function of the input instance's parameter.

The class W[1] is defined as the class of problems that parameterically reduce to the CLIQUE problem, and W[2] as the class of problems that parameterically reduce to the SET COVER problem, where both problems are parameterized by the solution size.

Definition 1. In the CLIQUE problem, we are given an undirected graph $G = (V(G), E(G))$ and an integer h. We ask for the existence of a set H of h vertices such that there is an edge between each pair of vertices from H.

Definition 2. In the SET COVER problem, we are given a universe set X, a family $\mathcal{S}$ of subsets of X, and an integer h. We ask for the existence of at most h sets from $\mathcal{S}$ whose union gives X.

A parameterized problem is contained in the class XP if there is an algorithm that, given an instance I for it with parameter k, solves it in time $|I|^{g(k)}$, where g is some computable function. It holds that $\mathsf{FPT} \subseteq \mathsf{W[1]} \subseteq \mathsf{W[2]} \subseteq \mathsf{XP}$.

3. THE PROBLEM

In this section, we define our COMBINATORIAL SHIFT BRIBERY problem (CSB) in its full generality and describe why and how we simplify it for the remainder of our study.

The Definition. Let $\mathcal{R}$ be some voting rule. The definition of $\mathcal{R}$-CSB is somewhat involved therefore we first define some necessary components. We are given an election $E = (C, V)$ and a *preferred candidate* p. The goal is to ensure that p is an $\mathcal{R}$-winner of the election. To this end, we have a number of possible actions to choose from.

Let $m := |C|$ be the number of candidates in E and let $n := |V|$ be the number of voters. A *shift action* f is an n-dimensional vector of (possibly negative) integers, $f = (f^{(1)}, \dots, f^{(n)})$. In $\mathcal{R}$-CSB we are given a family $F = (f_1, \dots, f_\zeta)$ of shift actions. Each particular shift action models a possible campaigning action, such as airing a TV spot or organizing a meeting with the voters. The components of a given shift action measure the effect of the corresponding campaigning action on the particular voters. Further, each shift action f_j ($1 \leq j \leq \zeta$) comes with a nonnegative integer cost $w(f_j)$ for applying this shift action. For a given subset $F' \subseteq F$ of available shift actions, we define the effect of F' on voter v_i ($1 \leq i \leq n$) as $\mathcal{E}^{(i)}(F') = \sum_{f_j \in F'} f_j^{(i)}$.

Each voter v_i ($1 \leq i \leq n$) has his or her individual *threshold function* $\pi_i : \mathbb{Z} \to \mathbb{Z}$ describing how shift actions affect this voter. We require that $\pi_i(0) = 0$ and that π_i is nondecreasing. Let F' be a collection of shift actions. After applying the shift actions from F', each voter v_i ($1 \leq i \leq n$) shifts the preferred candidate p by $t > 0$ positions forward exactly if (a) $\mathcal{E}^{(i)}(F') > 0$, and (b) $\pi_i(t) \leq \mathcal{E}^{(i)}(F') < \pi_i(t+1)$. The shift is by $t > 0$ positions back if (a) $\mathcal{E}^{(i)}(F') < 0$, and (b) $\pi_i(-t) \geq \mathcal{E}^{(i)}(F') > \pi_i(-t-1)$.

Finally, we are given a nonnegative integer B, the *budget*. We ask for the existence of a collection $F' \subseteq F$ of available shift actions with total cost $\sum_{f_j \in F'} w(f_j)$ at most B and such that after applying them, p is an $\mathcal{R}$-winner of the given election. If this is the case, then we say that F' is *successful*.

Example 1. Consider an election with $C = \{a, b, c, p\}$, where p is the preferred candidate, and with three voters, $v_1 : c \succ b \succ p \succ a$, $v_2 : b \succ a \succ c \succ p$, and $v_3 : p \succ a \succ b \succ c$. We are using the Borda rule. Candidates a, b, c, and p have, respectively, 4, 6, 4, and 4 points. There are three available shift actions: $f_1 = (2, 4, 0)$, $f_2 = (6, 0, -3)$, and $f_3 = (0, 2, 0)$. The threshold functions are such that: (1) $\pi_1(-1) = -4$, $\pi_1(0) = 0$, $\pi_1(1) = 6$, $\pi_1(2) = 100$, (2) $\pi_2(0) = 0$, $\pi_2(1) = 2$, $\pi_2(2) = \pi_2(3) = 100$, and (3) $\pi_3(-3) = \pi_3(-2) - 100$, $\pi_3(-1) = -3$, $\pi_3(0) = 0$. Each shift action has the same cost, $w(f_1) = w(f_2) = w(f_3) = 1$.

It is easy to see that applying any single shift action does not ensure p's victory. However, applying shift actions $F' = \{f_2, f_3\}$ results in p being a winner. The total effect of these two shift actions is $(6, 2, -3)$. According to the threshold functions, this means that p is shifted forward by one position in v_1 and v_2, and is shifted backwards by one position in v_3. After these shifts, the modified election has the following votes: $v_1' : c \succ p \succ b \succ a$, $v_2' : b \succ a \succ p \succ c$, and $v_3' : a \succ p \succ b \succ c$. Therefore, candidate c has 3 points, while all other candidates have 5 points each; a, b, and p are tied winners. Therefore, F' is successful.

Formally, given a voting rule $\mathcal{R}$, we define the $\mathcal{R}$-CSB problem as follows:

> $\mathcal{R}$-COMBINATORIAL SHIFT BRIBERY ($\mathcal{R}$-CSB)
> **Input:** An election $E = (C, V)$, where $C = \{c_1, \dots, c_m\}$ is the set of candidates and $V = (v_1, \dots, v_n)$ is the collection of voters, a set $F = \{f_1, \dots, f_\zeta\}$ of shift actions with costs $w(f_1), \dots, w(f_\zeta)$, threshold functions $\pi_1, \dots, \pi_n$, and a nonnegative integral budget B. One of the candidates is designated as the preferred candidate p.
> **Question:** Is there a subset $F' \subseteq F$ of shift actions with total cost at most B such that, after

we apply the shift actions from F', candidate p is an $\mathcal{R}$-winner of the resulting election?

While this definition is quite complicated, it captures some important features of campaigning. For example, the use of threshold functions allows us to model voters who are unwilling to change the position of the preferred candidate beyond a certain range, irrespective of the strength of the campaign. The fact that different shift actions have different costs models the fact that particular actions (for example, airing TV spots or organizing meetings) may come at different costs.

Relation to Standard Shift Bribery. It is now necessary to comment on the relation between our COMBINATORIAL SHIFT BRIBERY problem and its noncombinatorial variant, SHIFT BRIBERY [11, 12]. If we restrict our shift actions such that each shift action has a positive entry for exactly one voter, then—in effect—we obtain SHIFT BRIBERY for the case of *convex price functions* [5]. This is a very general variant of the SHIFT BRIBERY problem (but nonetheless not the most general one) for which, for example, all the known NP-hardness results hold. We decided not to complicate our definition further, to obtain a full generalization. As we will see below, doing so would obfuscate the problem without visible gain.

Hardness Result. The problem as defined above is so general that it allows for the following, sweeping, hardness result.[2]

THEOREM 1. *Both Plurality-*CSB *and Borda-*CSB *are (weakly)* NP*-hard even for five voters and two candidates.*

PROOF (CONSTRUCTION ONLY). We reduce from the following (weakly NP-hard) variant of the SUBSET SUM problem. We are given a set $A := \{a_1, \ldots, a_n\}$ of integers and we ask for the existence of a nonempty set A', $A' \subseteq A$, such that $\sum_{a_i \in A'} a_i = 0$.

Note that for elections with two candidates Plurality and Borda coincide. Given an instance $A = \{a_1, \ldots, a_n\}$ of our variant of SUBSET SUM, we construct an instance of Plurality-CSB with two candidates (indeed, since Borda and Plurality coincide for elections with two candidates, our hardness result transfers to Borda-CSB).

We take two candidates d and p, two voters, v_1 and v_2, preferring p, and three voters, v_3, v_4, and v_5, preferring d. For each element $a_i \in A$ we create one shift action f_i with effect a_i on v_1, effect $-a_i$ on v_2, effect 1 on v_3, and no effect on the other two voters. Each shift action has the same unit cost. The voter threshold functions are as follows. Candidate p is shifted to the last position for v_1 and v_2 if the effect on these voters is negative (that is, $\pi_1(-1) = \pi_2(-1) = -1$). Candidate p is shifted to the top position for the third voter if the effect is positive (that is, $\pi_3(1) = 1$). We set the cost of each shift action to be one and we set our budget to be n. $\square$

Effectively, Theorem 1 shows that studying large-scale effects of campaign actions through the full-fledged $\mathcal{R}$-CSB problem leads to a hopelessly intractable problem as we obtain hardness even for elections with both a fixed number of candidates and a fixed number of voters.

[2]Note, however, that we prove weak NP-hardness. That is, our result may not hold if we assume that all occurring numbers are encoded in unary. On the contrary, all other hardness proofs in this paper give strong hardness results and are independent of such number encoding issues.

Restriction of CSB. Throughout the rest of the paper, we assume the individual threshold functions to be identity functions (that is, for each voter i and each integer t, it holds that $\pi_i(t) = t$) and we assume each shift action to have the same unit cost. Note that the instances built in the proof of Theorem 1 satisfy these restrictions and, so, considering harder instances is beyond point. Notably, however we will consider restricted types of shift actions as well (see Section 4).

Our restrictions on the costs and on the threshold functions require discussion. First, identity threshold functions mean that we model societies that are prone to propaganda. Second, assuming that every shift action has the same unit cost models settings where the costs of particular campaign actions are similar enough that small differences between them are irrelevant; the actual number of actions we choose to perform is a sufficiently good approximation of the real cost. This is true, for example, for the case of organizing meetings with voters, which often have comparable prices. It is also likely to be the case when shift actions model actions such as airing TV spots: each spot has a similar cost to produce/broadcast. The greatest disadvantage of assuming unit costs is that we no longer can model mixed campaigns that use actions of several different types (meetings with voters, TV spots, web campaigns, etc.).

Considered Parameters. We consider the influence of the following parameters on the computational complexity of our CSB problems: (1) the number n of voters, (2) the number m of candidates, (3) the budget B, (4) the maximum effect Γ of a single shift action (that is, $\Gamma := \max_{f \in F} \max_{1 \le i \le n} f^{(i)}$ where $f^{(i)}$ denotes the ith position of the vector f), and (5) the maximum number Λ of voters affected by a single shift action (that is, $\Lambda := \max_{f \in F} |\{i : f^{(i)} \neq 0\}|$).

4. RESULTS

It turns out that even with the above restrictions in place, COMBINATORIAL SHIFT BRIBERY is computationally hard in many settings. Thus, what we present here is our quest for understanding the border between tractability and intractability of CSB. To this end, we employ the following techniques and ideas: (1) we seek both regular complexity results (NP-hardness results) and parameterized complexity results (FPT algorithms, W[1] and W[2]-hardness results, XP algorithms) (2) we consider structural restrictions on the sets of shift actions available, and (3) we seek approximation algorithms and inapproximability results.

We consider the following families of shift actions:

Unrestricted Shift Actions. Here we put no restrictions on the allowed shift actions; this models the most general (and, naturally, the least tractable) setting.

Bounded-Effect Shift Actions. Here we consider a parameter Γ, and require that for each shift action $f = (f^{(1)}, \ldots, f^{(n)})$, it holds that for each j ($1 \le j \le n$), we have $|f^{(j)}| \le \Gamma$. This is still a very general setting, where we assume that each campaigning action has only a limited impact on each voter.

Unit-Effect Shift Actions. This is a class of bounded-effect shift actions for $\Gamma = 1$. For each given voter, applying a given shift action can either leave the preferred candidate p unaffected or it can shift p one position up or down.

Interval Shift Actions. This a subclass of unit-effect shift actions that never affect voters negatively, and where for each shift action there is an interval of voters that are affected positively (the interval is with respect to the order of the voters in the input collection V). This class of shift actions models campaigns associated with a time window where certain voters can be reached or campaigns that are local to given neighborhoods[3] (for example, that include putting up multiple posters, organizing meetings, etc.).

Unit-Effect on Two Voters Shift Actions. This is a subclass of unit-effect shift actions that affect two voters at most. We focus on shift actions that affect both voters positively, denoted as $(+1,+1)$-shift actions, and that affect one voter positively and one voter negatively, denoted as $(+1,-1)$-shift actions. The reason for studying these families is not because they model particularly natural types of election campaigns, but rather to establish the limits of tractability for our problem. For example, we consider $(+1,-1)$-shift actions to understand how intractable are shift actions that have negative effects; $(+1,-1)$-shift actions are the simplest shift actions of this type that may be useful in the campaign (one would never deliberately use $(-1,-1)$-shift actions).

Figure 1 presents graphically the difference between interval shift actions and unit-effect on two voters shift actions.

The remainder of this section has the following structure. First, in Section 4.1, we show relations of Plurality-CSB to the problem of combinatorial control by adding voters, establishing quite general hardness results (applied already to unit-effect shift actions). Then, in Section 4.2, we present a series of strong hardness results covering all our classes of shift actions for very restrictive sets of parameters (for example, many of our results apply already to the case of two candidates). Then, in Sections 4.3 and 4.4, we present some ways of dealing with our hardness results. Table 1 gives an overview of our main results.

4.1 Connection to Combinatorial Control

The study of combinatorial variants of problems modeling ways of affecting election results was initiated by Chen et al. [7], who considered combinatorial control by adding voters (COMB-CCAV) for the Plurality rule and the Condorcet rule. It turns out that for the Plurality rule we can reduce the problem of (COMBINATORIAL) CCAV to that of (COMBINATORIAL) SHIFT BRIBERY. For the non-combinatorial variants of these problems this does not give much since both are easily seen to be polynomial-time solvable. However, there are strong hardness results for Plurality-COMB-CCAV which we can transfer to the case of Plurality-CSB. Formally, Plurality-COMBINATORIAL-CCAV is defined as follows [7].

Definition 3. An instance of Plurality-COMB-CCAV consists of a set C of candidates with a preferred candidate $p \in C$, a collection V of registered voters, a collection W of unregistered voters, a bundling function $\kappa \colon W \to 2^W$ (for each $w \in W$, it holds that $w \in \kappa(w)$), and a nonnegative integer k. Each voter has a preference order over C. We ask for the existence of a collection $W' \subseteq W$ of at most k voters such that p is a winner of the modifed election $(C, V \cup \bigcup_{w' \in W'} \kappa(w'))$.

Intuitively, for each unregistered voter $w \in W$, we have his or her bundle, $\kappa(w)$ (given explicitly in the input), such that when we add w to the election (for example, by somehow convincing him or her to vote), all the voters in his or her bundle also join the election (for example, people choose to vote under an influence of a friend).

[3]In the neighborhood scenario, we take the simplified view that a society of the voters lives on a line. Of course, it would be more natural to take two-dimensional neighborhoods into account. We view this as an interesting direction for future research, but for the time being we consider as simple settings as possible. In the time window scenario, a natural ordering of the voters is the point of time when they cast their votes or can be affected by the campaign.

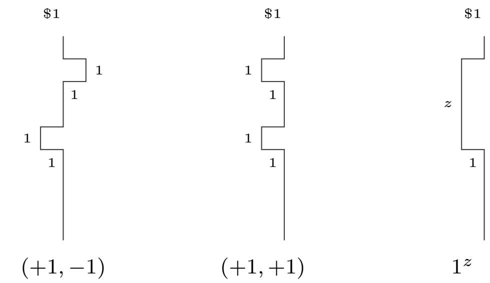

Figure 1: Restrictions on the shift actions. We visualize (from left to right) a shift action "$(+1,-1)$" with effect of $+1$ on one voter and effect of -1 on another voter, a shift action "$(+1,+1)$" with effect of $+1$ on two voters, and a shift action with effect of $+1$ on an interval of size z "1^z". The intended interpretation is that voters are listed vertically, from top to bottom.

THEOREM 2. *Plurality-*COMB-CCAV *is polynomial-time many-one reducible to Plurality-*CSB*. For an instance of Plurality-*COMB-CCAV *with m candidates, the reduction outputs an instance with $m+1$ candidates.*

The idea of the proof is as follows. We create an election with both the registered and unregistered voters present. For the unregistered voters that rank p first, we put a new candidate d on top, such that shifting p forward has the effect of "adding" this voter. For the unregistered voters that rank some candidate other than p first, we shift p to the top, such that shifting p back (with a negative effect of a shift action) has the effect of "adding" this voter (we also need some additional voters to maintain the same score differences as in the original control instance). Using this reduction and the results of Chen et al. [7], we obtain the following.

COROLLARY 3. *Plurality-*CSB *is $W[2]$-hard with respect to parameter B even if $m = 3$, it is $W[1]$-hard with respect to parameter B even for shift actions with unit effect on up to 6 voters, and it is* NP*-hard even for shift actions with unit effects on up to 4 voters.*

4.2 Hardness Results

The results from the previous section show that we are bound to hit hard instances for COMBINATORIAL SHIFT BRIBERY even in very restricted settings. Now, we explore how restrictive these hard settings are.

We start by considering unit-effect shift actions. If the allowed effects are positive only, then we obtain NP-

Table 1: Overview of our results. We show exact algorithms and approximation algorithms for Plurality-CSB and Borda-CSB, for different restrictions on the shift actions (see Figure 1). Note that all of the variants are XP for parameter B. Recall that n denotes the number of voters and m denotes the number of candidates.

shift actions	rule	exact complexity	approximability
regular SHIFT BRIBERY (convex prices)	Plurality	polynomial-time solvable ([12])	—
	Borda	NP-complete but in FPT for B [5, 12]	2-approximable in poly. time [11, 12] FPT-approximation scheme for n [5]
unit effect	Both	W[2]-h for B even if $m = 2$ (Thm. 4) XP for n (Prop. 10)	inapproximable even in FPT-time for B and $m = 2$ (Thm. 5)
$(+1, -1)$	Plurality	FPT for n (Thm. 11)	—
	Borda	W[1]-hard for n (Thm. 8)	—
	Both	NP-h even if $m = 2$ (Thm. 6) W[1]-h for B and m combined (Thm. 7)	inapproximable even if $m = 2$ (Thm. 6)
$(+1, +1)$	Plurality	FPT for n (Thm 11)	—
	Both	W[1]-h for B and m combined (Thm. 7)	2-approximable in poly. time (Thm. 12)
1^z-intervals	Plurality	FPT for n (Thm. 11)	z-approximable in poly. time (Thm. 12)
	Borda	—	$2z$-approximable in poly. time (Thm. 12)
	Both	W[1]-h for B (Thm. 9)	2-approximable in m^z time (Thm. 14)

hardness/W[2]-hardness for parameterization by the budget B. If we allow also negative unit-effects, we even go beyond any hope for an approximation algorithm, even if the approximation algorithm was allowed to run in FPT time for parameter B. Quite strikingly, these results hold even if we only have two candidates.

THEOREM 4. *Plurality*-CSB *and Borda*-CSB *are both* NP*-hard and* W[2]*-hard for the parameter B, even for two candidates and even if each shift action has effects of either* $+1$ *or* 0 *on each voter.*

THEOREM 5. *Unless* W[2] $=$ FPT*, both Plurality*-CSB *and Borda*-CSB *are inapproximable even in* FPT*-time for the parameter B, even for two candidates and even for unit-effect shift actions.*

PROOF. Let $(\mathcal{S}, X, h)$ be a SET COVER instance. We construct an instance of Plurality-CSB with two candidates. Since Borda and Plurality coincide for elections with two candidates, our hardness result transfers to Borda-CSB.

For each element $x_i \in X$, create $|\mathcal{S}|$ *element voters* $v_i^1, \ldots, v_i^{|\mathcal{S}|}$, each with preference order $d \succ p$, and for each set $S_j \in \mathcal{S}$ create a *set voter* v_j^0 with preference order $p \succ d$. Create $|\mathcal{S}| \cdot |X| + |\mathcal{S}| - 2h$ *dummy voters*, each with preference order $d \succ p$. The set F of shift actions contains, for each set S_j, a shift action f_j having an effect of 1 on each element voter corresponding to an element of the set and an effect of -1 on the set voter corresponding to the set. Finally, set $B := h$. This completes the construction.

Next, we show that there is a set cover of size h if and only if there is a successful set of shift actions of size h.

For the "only if" part, assume that there is a set cover $\mathcal{S}'$ of size at most h. Then, $F' = \{f_j \mid S_j \in \mathcal{S}'\}$ is a successful set of shift actions: since $\mathcal{S}'$ is a set cover, p will be the preferred candidate for all $|\mathcal{S}| \cdot |X|$ element voters and $|\mathcal{S}| - h$ set voters (corresponding to the sets not from the set cover) and d will be the preferred candidate for all $|\mathcal{S}| \cdot |X| + |\mathcal{S}| - 2h$ dummy voters and for h set voters (corresponding to the sets from the set cover). Hence, d and p tie as winners.

For the "if" part, assume that there is a successful set of shift actions $F' \subseteq F$ of size at most h. Then, p must be the preferred candidate for all element voters in the bribed election: if there is an element voter with $d \succ p$, then there are $|\mathcal{S}| - 1$ further element voters with $d \succ p$ (namely those being identical with respect to the effect of the shift actions). However, there will be in total at most $|\mathcal{S}|(|X| - 1)$ element voters and $|\mathcal{S}|$ set voters preferring p, but at least $|\mathcal{S}| \cdot |X| + |\mathcal{S}| - 2h$ dummy voters and $|\mathcal{S}|$ element voters preferring d. It follows that $\mathcal{S}' := \{S_j \mid f_j \in F'\}$ must be a set cover in order to make p win the election, and the size of the set S' is at most h due to the budget being h.

Finally, we show that Plurality-CSB is inapproximable even in FPT time when parameterized by the budget. Assume, towards a contradiction, that a set of shift actions $F' \subseteq F$ with $|F'| > B$ exists. Then, in the bribed election, at least $|\mathcal{S}| \cdot |X| + |\mathcal{S}| - 2h$ dummy voters and also $|F'| \geq h + 1$ set voters prefer d, but at most $|\mathcal{S}| \cdot |X|$ element voters and at most $|\mathcal{S}| - (h + 1)$ set voters prefer p. Thus, d is the unique winner. Hence, *any* successful bribery action must be optimal and any FPT-algorithm for Plurality-CSB (parameterized by the budget) would solve the W[2]-hard problem SET COVER (parameterized by the solution size) in FPT time; a contradiction to FPT $\neq$ W[2]. $\square$

In the above theorems we do not limit the number of voters affected by each shift action. Even if we consider only $(+1, -1)$-shift actions, then the problem remains NP-hard and hard to approximate even for two candidates.

THEOREM 6. *Unless* P $=$ NP*, both Plurality*-CSB *and Borda*-CSB *neither can be solved exactly nor can be approximated in polynomial time even for two candidates and even if we only have* $(+1, -1)$*-shift actions.*

Note that, compared to Theorem 5, Theorem 6 does not yield W[2]-hardness for the parameter budget B; our proof uses a reduction from Set Cover in which the value of the budget is the size of the universe set X. If we insist on parameterized hardness for unit effects on two voters, then we have to accept larger sets of candidates. However, this increase is not too large: we show W[1]-hardness of CSB jointly parameterized by the budget and the number of candidates.

Theorem 7. *Both Plurality-CSB and Borda-CSB are W[1]-hard for the combined parameter (m, B), even if we either only have $(+1,-1)$-shift actions or only have $(+1,+1)$-shift actions.*

Proof. We give a parameterized reduction from the W[1]-hard Clique problem, parameterized by the solution size, to Plurality-CSB, parameterized by (m, B), with $(+1,+1)$-shift actions (we omit the other cases due to space constraints).

Let (G, h) be an instance of Clique with $V(G) = \{u_1, \ldots, u_{n'}\}$ and $E(G) = \{e_1, \ldots, e_{m'}\}$. We create an instance of Plurality-CSB as follows. The set of candidates is $\{p\} \cup D$, where $D = \{d_1, \ldots, d_{h-1}\}$. For each vertex $u_i \in V(G)$, we create a *vertex voter* v_i with preference order $\overrightarrow{D} \succ p$. Moreover, we create h *dummy voters* with preference order $\overrightarrow{D} \succ p$ each, and $n' - h$ *dummy voter* with preference order $p \succ \overrightarrow{D}$ each. For each edge $\{u_i, u_j\} \in E(G)$, we create a shift action $f_{\{u_i,u_j\}}$ with effect 1 on the vertex voters v_i and v_j and effect 0 on all other voters. Finally, we set the budget to $B := \binom{h}{2}$. This completes the construction.

We assume that d_1 is ranked first in the order $\overrightarrow{D}$. Observe that we have n' vertex voters and h dummy voters that rank d_1 first. We also have $n' - h$ dummy voters that rank p first. Hence, to make p win the election, one needs h additional voters to rank p first (and, in effect, not rank d_1 first).

It remains to show that (G, h) contains a clique of size at most h if and only if our constructed instance contains a successful set of shift actions F' of size at most h.

For the "only if" part, let $H \subseteq V(G)$ be a set of h vertices forming a clique and $E' \subseteq E(G)$ be the set of edges between vertices from H. Then, observe that $F' = \{f_{\{u_i,u_j\}} \mid \{u_i, u_j\} \in E'\}$ is a successful set of shift actions: for each vertex voter v_i with $u_i \in H$, candidate p is shifted $h-1$ positions forward. This means that h vertex voters rank p first and p ties as a winner of the election.

For the "if" part, let F' be a successful set of shift actions. To make p a winner of the election, p must be shifted to the top position in at least h vertex voters (dummy voters cannot be affected). That is, in total p must be shifted $h \cdot (h-1)$ positions forward. Since F' is of size at most $B = \binom{h}{2} = h \cdot (h-1)/2$ and each shift action affects only two vertex voters, F' must be of size exactly $\binom{h}{2}$ affecting exactly h vertex voters. By construction, this implies that there are $\binom{h}{2}$ edges in G incident to exactly h different vertices which is only possible if these h vertices form a clique. $\square$

It is quite natural to consider CSB also from a different perspective. Instead of asking what happens for a small number of candidates, we might ask about the complexity of CSB for a small number of voters (see, for example, the work of Brandt et al. [4] for some motivation as to why looking at elections with few voters is interesting). In this case we obtain hardness for Borda only; we will show later that Plurality-CSB is in FPT for this parameter.

Theorem 8. *Borda-CSB is W[1]-hard with respect to the number n of voters, even for $(+1,-1)$-shift actions.*

Finally, we consider interval shift actions. In the above theorems we allowed shift actions to have non-zero effects on two voters each, but these two voters could have been chosen arbitrarily. Now we show a hardness result for the case where we can positively affect multiple voters, but these voters have to form a consecutive interval in the election.

Theorem 9. *Both Plurality-CSB and Borda-CSB are NP-hard even if each shift action has effect 1 on a consecutive interval interval of z voters and effect 0 on all other voters.*

A brief summary of the results from this section is that $\mathcal{R}$-CSB is highly intractable. Theorems 4, 5, and 6 show that it is computationally hard (in terms of NP-hardness, W[2]-hardness, and inapproximability even by FPT algorithms) both for Plurality and Borda even for various very restricted forms of unit-effect shift actions, even for two candidates. This means that, in essence, the problem is hard for all natural voting rules, because for two candidates all natural voting rules boil down to the Plurality rule.

Further, Theorems 7 and 9 show that our problems are W[1]-hard even if we take the number of candidates and the budget as a joint parameter, even for extremely restricted shift actions. The problem remains hard (for the case of Borda) when parameterized by the number of voters (Theorem 8). On the contrary, for the case of Plurality, for this parameter we will show tractability.

4.3 Exact Algorithms

In spite of the pessimism looming from the previous section, we now show two exact FPT and XP algorithms for $\mathcal{R}$-CSB; in Section 4.4 we present some efficient approximation algorithms.

We show an XP algorithm for the case of bounded-effect shift actions (which include all unit-effect shift actions) for our problem, when parameterized by the number of voters.

Proposition 10. *If the maximum effect is upper-bounded by a constant, then both Plurality-CSB and Borda-CSB are in XP for the parameter budget B.*

Proposition 10 holds even if each shift action comes at an individual cost and if each voter has an individual threshold function. By expressing our problem as an integer linear program (ILP) and by using a famous result of Lenstra [19], we can strengthen the XP-membership to FPT-membership, for the case of Plurality.

Theorem 11. *For bounded-effect shift actions, Plurality-CSB is in FPT for the parameter number n of voters.*

Proof. Given an instance I of Plurality-CSB with n voters, our algorithm proceeds as follows. First, we guess a subset of the voters for whom we will guarantee that p is ranked first (there are 2^n guesses to try). For each guessed set of voters, we test whether p would be a winner of the election if p was shifted to the top position by the guessed voters and was not ranked first by the remaining voters. For each set of guessed subset V' of voters for which this test is positive, we check whether it is possible to ensure (by applying shift actions whose cost does not exceed the budget) that the voters from V' rank p first. We do so as follows.

Let Γ be the value bounding, component-wise, the effect of each shift action. Observe that there are at most $(2\Gamma+1)^n$ types of different shift actions. For each shift action type z, we introduce a variable x_z denoting the number of times a shift action of type z is present in the solution. For each voter v_i, denote by $s_{v_i}(p)$ the position of p in the original preference order of v_i. For each voter $v_i \in V'$, we add the following constraint:

$$\sum_{\gamma\in[-\Gamma,\Gamma]} \left(\gamma \sum_{\{z: f_z \text{ has an effect of } \gamma \text{ on } v_i\}} x_z\right) \geq s_{v_i}(p).$$

This ensures that p is indeed shifted to the top position in v_i's preference list. We add another constraint: $\sum x_z \leq B$, ensuring that the solution respects the budget. Finally, for each shift action type z we add a constraint ensuring that we use at most as many shift actions of type z as there are available in the input. This finishes the description of the ILP. By a result of Lenstra [19], we can solve this ILP in FPT time, because we have at most $(2\Gamma + 1)^n$ integer variables.

Correctness follows from the fact that, for any "yes"-instance, there must be at least one good guess of V'; for this V', the ILP will compute a correct answer. $\square$

Theorem 11 is the reason why Theorem 8 applies to Borda only.

4.4 Approximation Algorithms

We now explore the possibility of finding approximate solutions for Combinatorial Shift Bribery. Theorems 5 and 6 show that we cannot hope for approximate results when shift actions have negative effects. Thus, in this section, we focus on unit-effect shift-actions with positive effects only. We can make good use of the approximation algorithms for the non-combinatorial variant of Shift Bribery.

Theorem 12. *If each shift action has effects of either* 0 *or* 1 *on each voter, then Plurality*-CSB *can be Λ-approximated in polynomial-time and Borda*-CSB *can be* 2Λ*-approximated in polynomial time, where Λ denotes the maximum number of voters affected by a shift action.*

Proof. We consider the case of Borda-CSB first. The idea is to translate the instance into a regular Borda-Shift Bribery instance with convex price functions and apply the known 2-approximation algorithm [11, 12] and "greedily reconstruct" the corresponding solution with combinatorial shift actions.

Let $\lambda(i)$ denote the number of shift actions affecting voter i. We copy the election from our Borda-CSB instance and set the price function for each voter i such that, for $j \leq \lambda(i)$, shifting p by j positions costs j and for $j > \lambda(i)$ shifting p by j positions costs B^j (note that the exponential function B^j ensures that the price functions are convex and that one cannot shift p by more than $\lambda(i)$ positions). Then, we apply the 2-approximation algorithm whose solution yields, for each voter i, the number $s(i)$ of positions p is shifted in the preference list of voter i. The cost of this solution is $\sum_{1\leq i\leq n} s(i)$. To obtain a solution F' for the Borda-CSB instance we greedily select shift actions until p was shifted at least $s(i)$ positions for each voter i. Observe that $|F'| \leq \sum_{1\leq i\leq n} s(i)$. Now, assume towards a contradiction that there is a successful set of shift actions F'' smaller than $|F'|/2\Lambda$. Then, in this solution p was shifted less than $|F'|/2$ positions in total, because each shift action affects at most Λ voters. However, then there would also be a solution for the regular Shift Bribery instance with costs smaller than $\sum_{1\leq i\leq n} s(i)/2$; a contradiction to the fact that we applied a 2-approximation.

The case of Plurality-CSB follows analogously, but we use an exact polynomial-time algorithm for Plurality-Shift Bribery instead of the 2-approximate one. $\square$

We can achieve better approximation guarantees for Borda, when further restricting the allowed shift actions. The next result follows by finding (in polynomial-time) a b-matching [17] in a naturally constructed multigraph, followed by using a general framework for 2-approximation algorithms for Shift Bribery [11].

Theorem 13. *Borda*-CSB *is* 2*-approximable in polynomial time for* $(+1, +1)$*-shift actions.*

The above result is mostly of theoretical value: it shows that indeed we can retain high-quality polynomial-time approximation algorithms when some combinatorial shift-action effects are allowed. However, from a practical point of view, it is more interesting to see what we can achieve for, say, interval shift actions. Using the ideas of Elkind and Faliszewski [11] and a natural generalization of the maximum interval cover algorithm of Gabow [17], we obtain a 2-approximation algorithm for CSB with interval shift actions. Unfortunately, the algorithm requires XP time, parameterized by the longest interval's length.

Theorem 14. *Both Plurality*-CSB *and Borda*-CSB *can be* 2*-approximated in* XP*-time for interval shift actions, provided that we take z, the upper bound on the number of voters affected by each shift action, as the parameter.*

We see that it is still possible to achieve some approximation algorithms for our problems. However, the settings in which our algorithms are efficient are quite restrictive. This means that in practice one should seek good heuristics and use our algorithms only to guide the initial search.

5. CONCLUSIONS

We have defined and studied the computational complexity of a combinatorial variant of the Shift Bribery problem [5, 11, 12]. Our research was motivated by the desire to understand the computational difficulty imposed by correlated, large-scale effects of campaign actions. In this respect, our work was motivated by the combinatorial study of election control [7]. We have found that our problem, Combinatorial Shift Bribery, is highly worst-case intractable in multiple ways. Nonetheless, we found some initial positive results. Our results suggest studying further restrictions of the problem: for example, parameterization by the number of available shift actions gives immediate FPT results, so maybe there are other natural parameterizations that lead to more positive results? It would be also interesting to consider domain restrictions regarding voters' preferences (single-crossing seems particularly natural in the context of interval shift actions).

REFERENCES

[1] J. J. Bartholdi, III, C. A. Tovey, and M. A. Trick. How hard is it to control an election. *Mathematical and Computer Modelling*, 16(8–9):27–40, 1992.

[2] 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.

[3] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional *ceteris paribus* preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[4] F. Brandt, P. Harrenstein, K. Kardel, and H. Seedig. It only takes a few: On the hardness of voting with a constant number of agents. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '13)*, pages 375–382. IFAAMAS, 2013.

[5] 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.

[6] D. Cary. Estimating the margin of victory for instant-runoff voting. Presented at the 2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, 2011.

[7] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. Combinatorial voter control in elections. In *Proceedings of the 39th International Symposium on Mathematical Foundations of Computer Science (MFCS '14)*, volume 8635 of *LNCS*, pages 153–164. Springer, 2014.

[8] V. Conitzer, J. Lang, and L. Xia. How hard is it to control sequential elections via the agenda? In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI '10)*, pages 103–108. AAAI Press, 2009.

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

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

[11] 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.

[12] 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.

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

[14] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland Voting Computationally Resist Bribery and Constructive Control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

[15] 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.

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

[17] H. N. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the 15th Annual ACM Symposium on Theory of Computing (STOC '83)*, pages 448–456. ACM, 1983.

[18] N. Hazon, R. Lin, and S. Kraus. How to change a group's collective decision? In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI '13)*, pages 198–205. AAAI Press, 2013.

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

[20] T. Magrino, R. Rivest, E. Shen, and D. Wagner. Computing the margin of victory in IRV elections. Presented at the 2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, 2011.

[21] 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.

[22] 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.

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

[24] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI '11)*, pages 319–324. AAAI Press, 2011.

[25] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '11)*, pages 71–78, 2011.

[26] 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.

[27] L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC '12)*, pages 982–999. ACM Press, 2012.