Multivariate Complexity Analysis of Swap Bribery

Britta Dorn and Ildikó Schlotter[1]

Abstract

We consider the computational complexity of a problem modeling *bribery* in the context of voting systems. In the scenario of SWAP BRIBERY, each voter assigns a certain price for swapping the positions of two consecutive candidates in his preference ranking. The question is whether it is possible, without exceeding a given budget, to bribe the voters in a way that the preferred candidate wins in the election. We initiate a parameterized and multivariate complexity analysis of SWAP BRIBERY, focusing on the case of k-approval. We investigate how different cost functions affect the computational complexity of the problem. We identify a special case of k-approval for which the problem can be solved in polynomial time, whereas we prove NP-hardness for a slightly more general scenario. We obtain fixed-parameter tractability as well as W[1]-hardness results for certain natural parameters.

1 Introduction

In the context of voting systems, the question of how to manipulate the votes in some way in order to make a preferred candidate win the election is a very interesting question. One possibility is *bribery*, which can be described as spending money on changing the voters' preferences over the candidates in such a way that a preferred candidate wins, while respecting a given budget. There are various situations that fit into this scenario: The act of remunerating the voters in order to make them change their preferences, or paying money in order to get into the position of being able to change the submitted votes, but also the setting of systematically spending money in an election campaign in order to convince the voters to change their opinion on the ranking of candidates.

The study of bribery in the context of voting systems was initiated by Faliszewski, Hemaspaandra, and Hemaspaandra in 2006 [12]. Since then, various models have been analyzed. In the original version, each voter may have a different but fixed price which is independent of the changes made to the bribed vote. The scenario of nonuniform bribery introduced by Faliszewski [11] and the case of microbribery studied by Faliszewski, Hemaspaandra, Hemaspaandra, and Rothe in [13] allow for prices that depend on the amount of change the voter is asked for by the briber.

In addition, the SWAP BRIBERY problem as introduced by Elkind, Faliszewski, and Slinko [10] takes into consideration the ranking aspect of the votes: In this model, each voter may assign different prices for swapping two consecutive candidates in his preference ordering. This approach is natural, since it captures the notion of small changes and comprises the preferences of the voters. Elkind et al. [10] prove complexity results for this problem for several election systems such as Borda, Copeland, Maximin, and approval voting. In particular, they provide a detailed case study for k-approval. In this voting system, every voter can specify a group of k preferred candidates which are assigned one point each, whereas the remaining candidates obtain no points. The candidates which obtain the highest sum of points over all votes are the winners of the election. Two prominent special cases of k-approval are plurality, (where $k = 1$, i.e., every voter can vote for exactly one candidate) and veto (where $k = m - 1$ for m candidates, i.e., every voter assigns one point

[1]Supported by the Hungarian National Research Fund (OTKA 67651).

	Result	Reference
$k = 1$ (plurality)	P	[10]
$k = m - 1$ (veto)	P	[10]
$1 \leq k \leq m$, m or n constant	P	[10]
$1 \leq k \leq m$, all costs $= 1$	**P**	Thm. 1
$k = 2$	NP-complete	[2]
$3 \leq k \leq m - 2$, costs in $\{0, 1, 2\}$	NP-complete	[10]
$2 \leq k \leq m - 2$, costs in $\{0, 1\}$ and $\beta = 0$	**NP-complete**	[2], Prop. 2
$2 \leq k \leq m - 2$ is part of the input, costs in $\{0, 1\}$ and $\beta = 0$, n constant	**NP-complete**	[3], Prop. 2
$2 \leq k \leq m - 2$, costs in $\{\delta_1, \delta_2\}$, $\delta_2 \geq 2\delta_1 > 0$	**NP-complete**, **W[1]-hard** (β)	Thm. 3
$1 \leq k \leq m$	**FPT** (m)	Thm. 4
$1 \leq k \leq m$ is part of the input	**FPT** (β, n) **by kernelization**	Thm. 5
$1 \leq k \leq m$	**FPT** (β, n, k) **by kernelization**	Thm. 5

Table 1: Overview of known and new results for Swap Bribery for k-approval. The results obtained in this paper are printed in bold. Here, m and n denote the number of candidates and votes, respectively, and β is the budget. For the parameterized complexity results, the parameters are indicated in brackets. If not stated otherwise, the value of k is fixed.

to all but one disliked candidate). Table 1 shows a summary of research considering Swap Bribery for k-approval, including both previously known and newly achieved results.

This paper contributes to the further investigation of the case study of k-approval that was initiated in [10], this time from a parameterized point of view. The main goal of this approach is to find *fixed-parameter algorithms* confining the combinatorial explosion which is inherent in NP-hard problems to certain problem-specific parameters, or to prove that their existence is implausible. This line of research has been pioneered by Downey and Fellows [9], see also [15, 21] for two more recent monographs, and naturally expands into the field of multivariate algorithmics, where the influence of "combined" parameters is studied, see the recent survey by Niedermeier [22]. These approaches seem to be appealing in the context of voting systems, where NP-hardness is a desired property for various problems, like Manipulation, Lobbying, Control, or, as in our case, Swap Bribery. However, NP-hardness does not necessarily constitute a guarantee against such dishonest behavior. As Conitzer et al. [8] pointed out for the Manipulation problem, an NP-hardness result in these settings would lose relevance if an efficient fixed-parameter algorithm with respect to an appropriate parameter was found. Parameterized complexity can hence provide a more robust notion of hardness. The investigation of problems from voting theory under this aspect has started, see for example [1, 3, 4, 7, 20].

We show NP-hardness as well as fixed-parameter intractability of Swap Bribery for certain very restricted cases of k-approval if the parameter is the budget, whereas we identify a natural special case of the problem which can be solved in polynomial time. By contrast, we obtain fixed-parameter tractability with respect to the parameter 'number of candidates' for k-approval and a large class of other voting systems, and a polynomial kernel for k-approval if we consider certain combined parameters.

The paper is organized as follows. After introducing notation in Section 2, we investigate the complexity of Swap Bribery depending on the cost function in Section 3, where we show the connection to the Possible Winner problem, identify a polynomial-time solvable case of k-approval and a hardness result. In Section 4, we consider the parameter 'number of candidates' and obtain an FPT result for Swap Bribery for a large class of voting systems. We also consider the combination of parameters 'number of votes' and 'size of the budget'. We conclude with a discussion of open problems and further directions that might be interesting for future investigations.

2 Preliminaries

Elections. An *election* is a triple $E = (V, C, \mathcal{E})$, where $V = \{v_1, \ldots, v_n\}$ denotes the set of *votes* or *voters*, $C = \{c_1, \ldots, c_m\}$ is a set of *candidates*, and $\mathcal{E}$ is the *election system* which is a function mapping (V, C) to a set $W \subseteq C$ called the *winners* of the election. We will express our results for the *winner case* where several winners are possible, but our results can be adapted to the *unique winner case* where W consists of a single candidate only.

In our context, each vote is a strict linear order over the set C, and we denote by $\mathrm{rank}(c, v)$ the position of candidate $c \in C$ in a vote $v \in V$.

For an overview of different election systems, we refer to [6]. We will mainly focus on election systems that are characterized by a given *scoring rule*, expressed as a vector $(s_1, s_2, \ldots, s_m)$ where $m = |C|$. Given such a scoring rule, the *score* of a candidate c in a vote v, denoted by $\mathrm{score}(c, v)$, is $s_{\mathrm{rank}(c,v)}$. The score of a candidate c in a set of votes V is $\mathrm{score}(c, V) = \sum_{v \in V} \mathrm{score}(c, v)$, and the winners of the election are the candidates that receive the highest score in the given votes.

The election system we are particularly interested in is k-approval, which is defined by the scoring vector $(1, \ldots, 1, 0, \ldots, 0)$, starting with k ones. In the case of $k = 1$, this is the *plurality* rule, whereas $(m-1)$-approval is also known as *veto*. Given a vote v, we will say that a candidate c with $1 \leq \mathrm{rank}(c, v) \leq k$ takes a *one-position* in v, whereas a candidate c' with $k+1 \leq \mathrm{rank}(c', v) \leq m$ takes a *zero-position* in v.

Swap Bribery, Possible Winner, Manipulation. Given V and C, a *swap* in some vote $v \in V$ is a triple (v, c_1, c_2) where $\{c_1, c_2\} \subseteq C, c_1 \neq c_2$. Given a vote v, we say that a swap $\gamma = (v, c_1, c_2)$ is *admissible in* v, if $\mathrm{rank}(c_1, v) = \mathrm{rank}(c_2, v) - 1$. Applying this swap means exchanging the positions of c_1 and c_2 in the vote v, we denote by v^γ the vote obtained this way. Given a vote v, a set Γ of swaps is *admissible in* v, if the swaps in Γ can be applied in v in a sequential manner, one after the other, in some order. Note that the obtained vote, denoted by v^Γ, is independent from the order in which the swaps of Γ are applied. We also extend this notation for applying swaps in several votes, in the straightforward way.

In a Swap Bribery instance, we are given V, C, and $\mathcal{E}$ forming an election, a preferred candidate $p \in C$, a cost function c mapping each possible swap to a non-negative integer, and a budget $\beta \in \mathbb{N}$. The task is to determine a set of admissible swaps Γ whose total cost is at most β, such that p is a winner in the election $(V^\Gamma, C, \mathcal{E})$. Such a set of swaps is called a *solution* of the Swap Bribery instance. The underlying decision problem is the following.

Swap Bribery
Given: An election $E = (V, C, \mathcal{E})$, a preferred candidate $p \in C$, a cost function c mapping each possible swap to a non-negative integer, and a budget $\beta \in \mathbb{N}$.
Question: Is there a set of swaps Γ whose total cost is at most β such that p is a winner in the election $(V^\Gamma, C, \mathcal{E})$?

We will also show the connection between Swap Bribery and the Possible Winner problem. In this setting, we have an election where some of the votes may be *partial* orders over C instead of complete linear ones. The question is whether it is possible to extend the partial votes to complete linear orders in such a way that a preferred candidate wins the election. For a more formal definition, we refer to the article by Konczak and Lang [18] who introduced this problem. The corresponding decision problem is defined as follows.

Possible Winner
Given: A set of candidates C, a set of partial votes $V' = (v'_1, \dots, v'_n)$ over C, an election system $\mathcal{E}$, and a preferred candidate $p \in C$.
Question: Is there an extension $V = (v_1, \dots, v_n)$ of V' such that each v_i extends v'_i, and p is a winner in the election $(V, C, \mathcal{E})$?

A special case of Possible Winner is Manipulation (see e.g. [8, 17]). Here, the given set of partial orders consists of two subsets; one subset contains linearly ordered votes and the other one completely unordered votes.

Parameterized complexity, Multivariate complexity. Parameterized complexity is a two-dimensional framework for studying the computational complexity of problems [9, 15, 21]. One dimension is the size of the input I (as in classical complexity theory) and the other dimension is the parameter k (usually a positive integer). A problem is called *fixed-parameter tractable* (FPT) with respect to a parameter k if it can be solved in $f(k) \cdot |I|^{O(1)}$ time, where f is an arbitrary computable function [9, 15, 21]. Multivariate complexity is the natural sequel of the parameterized approach when expanding to multidimensional parameter spaces, see [22]. For example, if we regard two parameters, say k_1 and k_2, then the desired FPT algorithm should run in time $f(k_1, k_2) \cdot |I|^{O(1)}$ for some f.

The first level of (presumable) parameterized intractability is captured by the complexity class W[1]. A *parameterized reduction* reduces a problem instance (I, k) in $f(k) \cdot |I|^{O(1)}$ time to an instance (I', k') such that (I, k) is a yes-instance if and only if (I', k') is a yes-instance, and k' only depends on k but not on $|I|$.

We will use the following W[1]-hard problem [14] for the hardness reduction in this work:

Multicolored Clique
Given: An undirected graph $G = (V_1 \cup V_2 \cup \cdots \cup V_k, E)$ with $V_i \cap V_j = \emptyset$ for $1 \leq i < j \leq k$ where the vertices of V_i induce an independent set for $1 \leq i \leq k$.
Question: Is there a complete subgraph (clique) of G of size k?

We will also make use of a *kernelization* algorithm in this work, which is a standard technique for obtaining fixed-parameter results, see [5, 16, 21]. The idea is to transform the input instance (I, k) in a polynomial time preprocessing step via *data reduction rules* into a "reduced" instance (I', k') such that two conditions hold: First, (I, k) is a yes-instance if and only if (I', k') is a yes-instance, and second, the size of the reduced instance depends on the parameter only, i.e., $|I'| + |k'| \leq g(p)$ for some arbitrary computable function g. The reduced instance (I', k') is then referred to as the *problem kernel.* If in addition g is a polynomial function, we say that the problem admits a *polynomial kernel.* The existence of a problem kernel is equivalent to fixed-parameter tractability of the corresponding problem with respect to the particular parameter [21].

3 Complexity depending on the cost function

In this section, we focus our attention on Swap Bribery for k-approval. We start with the case where all costs are equal to 1, for which we obtain polynomial-time solvability.

Theorem 1. Swap Bribery *for k-approval is polynomial-time solvable, if all costs are* 1.

Proof. Let V be the set of votes and C be the set of candidates. The score of any candidate is an integer between 0 and $|V|$. Our algorithm finds out for each possible s^* with $1 \leq s^* \leq |V|$ whether there is a solution in which the preferred candidate p wins with score s^*.

Given a value s^*, we answer the above question by solving a corresponding minimum cost maximum flow problem. We will define a network $N = (G, s, t, g, w)$ on a directed graph $G = (D, E)$ with a source vertex s and a target vertex t, where g denotes the capacity function and w the cost function defined on E. First, we introduce the vertex sets $A = \{a_{v,c} \mid v \in V, c \in C, \mathrm{rank}(c, v) \leq k\}$, $A' = \{a'_{v,c} \mid v \in V, c \in C\}$ and $B = \{b_c \mid c \in C\}$, and we set $D = \{s, t, x\} \cup A \cup A' \cup B$. We define the arcs E as the union of the sets $E_S = \{sa \mid a \in A\}$, $E_A = \{a_{v,c}a'_{v,c} \mid \mathrm{rank}(c, v) \leq k\}$, $E_{A'} = \{a_{v,c}a'_{v,c'} \mid \mathrm{rank}(c, v) \leq k, \mathrm{rank}(c', v) > k\}$, $E_B = \{a'_{v,c}b_c \mid v \in V, c \in C\}$, $E_X = \{b_c x \mid c \in C, c \neq p\}$, plus the arcs $b_p t$ and xt. We set the cost function w to be 0 on each arc except for the arcs of $E_{A'}$, and we set $w(a_{v,c}a'_{v,c'}) = \mathrm{rank}(c', v) - \mathrm{rank}(c, v)$. We let the capacity g be 1 on the arcs of $E_S \cup E_A \cup E_{A'} \cup E_B$, we set it to be s^* on the arcs of $E_X \cup \{b_p t\}$, and we set $g(xt) = |V|k - s^*$.

The soundness of the algorithm and hence the theorem itself follows from the following observation (for a detailed proof, see the full version): there is a flow of value $|V|k$ on N having total cost at most β if and only if there exists a set Γ of swaps with total cost at most β such that $\mathrm{score}(p, V^\Gamma) = s^*$ and $\mathrm{score}(c, V^\Gamma) \leq s^*$ for any $c \in C, c \neq p$. □

Theorem 1 also implies a polynomial-time approximation algorithm for SWAP BRIBERY for k-approval with approximation ratio δ, if all costs are in $\{1, \delta\}$ for some $\delta \geq 1$.

Proposition 2 shows the connection between SWAP BRIBERY and POSSIBLE WINNER. This result is an easy consequence of a reduction given by Elkind et al. [10]. For the proof of the other direction, see again the full version.

Proposition 2. *The special case of* SWAP BRIBERY *where the costs are in* $\{0, \delta\}$ *for some* $\delta > 0$ *and the budget is zero is equivalent to the* POSSIBLE WINNER *problem.*

As a corollary, SWAP BRIBERY with costs in $\{0, \delta\}$, $\delta > 0$ and budget zero is NP-complete for almost all election systems based on scoring rules [2]. For many voting systems such as k-approval, Borda, and Bucklin, it is NP-complete even for a fixed number of votes [3].

We now turn to the case with two different positive costs, addressing 2-approval.

Theorem 3. *(1)* SWAP BRIBERY *for 2-approval, with costs in* $\{1, 2\}$, *is NP-complete.*
(2) SWAP BRIBERY *for 2-approval, with costs in* $\{1, 2\}$, *is W[1]-hard, if the parameter is the budget* β, *or equivalently, the maximum number of swaps allowed.*

Proof. We present a reduction from the MULTICOLORED CLIQUE problem. Let $F = (V, E)$ with the k-partition $V = V_1 \cup V_2 \cup \cdots \cup V_k$ be the given instance of MULTICOLORED CLIQUE. For each $1 \leq i < j \leq k$ we let $E^{i,j} = \{xy \mid x \in V_i, y \in V_j, xy \in E\}$. We construct an instance I_F of SWAP BRIBERY as follows.

The set C of candidates will be $C = \bigcup_{i \in [k]} (A_i \cup B_i \cup C_i) \cup D \cup G \cup \{p\}$ where $A_i = \{a_v^j \mid j \in [k], v \in V_i\}$, $B_i = \{b_v^j \mid j \in [k], v \in V_i\}$, and $C_i = \{c^{i,j} \mid j \in [k]\}$. (Here and later, we write $[k]$ for $\{1, 2, \ldots, k\}$.) Our preferred candidate is p. The sets $D = \{d_1, d_2, \ldots\}$ and $G = \{g_1, g_2, \ldots\}$ will contain *dummies* and *guards*, respectively. Our budget will be $\beta = 6k^2 - k$. Regarding the indices i and j, we will suppose $i, j \in [k]$ if not stated otherwise.

The set of votes will be $W = W_G \cup W_I \cup W_S \cup W_C$. Votes in W_G will define guards (explained later), votes in W_I will set the initial scores, votes in W_S will represent the selection of $\binom{k}{2}$ edges and k vertices, and finally, votes in W_C will be responsible for checking that the selected edges connect selected vertices. We construct W such that the following will hold for some fixed even integer K (determined later):

$\mathrm{score}(p, W) = K$.
$\mathrm{score}(c^{i,j}, W) = K + 1$ for each i and j,
$\mathrm{score}(q, W) = K$ for each $q \in \bigcup_{i \in [k]} (A_i \cup B_i) \cup G$, and
$\mathrm{score}(d, W) \leq 1$ for each $d \in D$.

We define the cost function c such that each swap has cost 1 or 2. We will define each cost to be 1 if not explicitly stated otherwise. Using that each cost is at least 1, we get that none of the candidates ranked after the position $\beta+2$ in a vote v can receive non-zero score in v without violating the budget. Thus, we can represent votes by listing only their first $\beta+2$ positions. We say that a candidate does not *appear* in some vote, if he is not contained in these positions.

Dummies, guards, and truncation. First, let us clarify the concept of dummy candidates: we will ensure that no dummy can receive more than one score in total, by letting each $d \in D$ appear in exactly one vote. This can be ensured easily by using at most $|W|(\beta+2)$ dummies in total. We will use the sign $*$ to denote dummies in votes.

Now, we define $\beta+2$ guards using the votes W_G. We let W_G contain votes of the form $w_G(h)$ for each $h \in [\beta+2]$, each such vote having multiplicity $K/2$ in W_G. We let $w_G(h) = (g_h, g_{h+1}, g_{h+2}, \dots, g_{\beta+2}, g_1, g_2, \dots g_{h-1})$. Clearly, $\text{score}(g, W_G) = K$ for each $g \in G$, and the total score obtained by the guards in W_G cannot decrease. As we will make sure that our preferred candidate cannot receive more than K scores without exceeding the budget, this yields that in any possible solution, each guard must have score exactly K.

Using guards, we can *truncate* votes at any position $h > 2$ by putting arbitrarily chosen guards at the positions $h, h+1, \dots, \beta+2$. This way we ensure that only candidates on the first $h-1$ positions can receive a score in this vote. We will denote truncation at position h by using a sign † at that position.

Setting initial scores. Using dummies and guards, we define W_I to adjust the initial scores of the relevant candidates as follows. We put the following votes into W_I:

$$(p, *, \dagger) \text{ with multiplicity } K,$$
$$(c^{i,j}, *, \dagger) \text{ with multiplicity } K+1-|E^{i,j}| \text{ for each } i \neq j,$$
$$(c^{i,i}, *, \dagger) \text{ with multiplicity } K+1-|V_i| \text{ for each } i \in [k], \text{ and}$$
$$(q, *, \dagger) \text{ with multiplicity } K-1 \text{ for each } q \in \bigcup_{i \in [k]} (A_i \cup B_i).$$

The preferred candidate p will not appear in any other vote, implying $\text{score}(p, W) = K$.

Selecting edges and vertices. The set W_S consists of the following votes:

$$w_S(i, x) = (*, c^{i,i}, a^i_x, \dagger) \text{ for each } i \in [k] \text{ and } x \in V_i, \text{ and}$$
$$w_S(i, j, x, y) = (c^{i,j}, c^{j,i}, a^j_x, a^i_y, \dagger) \text{ for each } i < j,\ x \in V_i,\ y \in V_j,\ xy \in E.$$

The cost of swapping $c^{i,j}$ with $c^{j,i}$ and the cost of swapping a^i_x with a^j_y in $w_S(i, j, x, y)$ is 2.

Checking incidency. The set W_C will contain the votes

$$w_C(i, x) = (a^i_x, b^{i-1}_x, b^i_x, *, \dagger) \text{ for each } i \in [k] \text{ and } x \in V_i.$$

Here $i-1$ is taken modulo k. In $w_C(i, x)$ we let the cost of swapping a^i_x with b^{i-1}_x and also the cost of swapping b^i_x with the neighboring dummy be 2.

It remains to define K properly. To this end, we let K be the minimum even integer not smaller than the integers $|E^{i,j}|$ for every $1 \leq i < j \leq k$ and $|V_i|$ for each $i \in [k]$. This finishes the construction. Note that the initial scores of the candidates are as claimed above.

Construction time. Observe $|W_G| = (\beta+2)K/2$, $|W_I| = O(Kk|V|)$, $|W_S| = |E| + |V|$, and $|W_C| = |V|$. Hence, the number of votes is polynomial in the size of the input graph F. This also implies that the number of candidates is polynomial as well, and the whole construction takes polynomial time. Note also that β is only a function of k, hence this yields an FPT reduction as well.

Our aim is to show the following: F has a k-clique if and only if the constructed instance is a yes-instance of Swap Bribery. This will prove both (1) and (2).

Direction $\Longleftarrow$. Suppose that I_F is solvable, and there is a set Γ of swaps transforming W into W' with total cost at most β such that p wins in W' according to 2-approval. We also assume w.l.o.g. that Γ is a solution having minimum cost.

As argued above, $\text{score}(p, W') \leq K$ and $\text{score}(g, W') \geq K$ for each $g \in G$ follow directly from the construction. Thus, only $\text{score}(p, W') = \text{score}(g, W') = K$ for each $g \in G$ is possible. Thus, for any $i, j \in [k]$, by $\text{score}(c^{i,j}, W) = K + 1$ we get that $c^{i,j}$ must lose at least one score during the swaps. Considering $c^{i,i}$ (and the optimality of Γ), this means that each $c^{i,i}$ is swapped with a^i_x by Γ in $w_S(i, x)$ for some unique $x \in V_i$. We use the notation $\sigma(i)$ to denote this vertex x, i.e. we let $\sigma(i) = x$. We will show that the vertices $\sigma(1), \sigma(2), \ldots, \sigma(k)$ form a k-clique in F.

Let us denote by Γ_{vs} the set of those swaps in Γ that swap $c^{i,i}$ with $a^i_{\sigma(i)}$ for some $i \in [k]$. Clearly, Γ_{vs} has total cost k.

Let us fix i and j now, assuming $i < j$. Since both $c^{i,j}$ and $c^{j,i}$ have the same score in W_I as in W_I^{Γ}, $c^{i,j}$ must lose a score due to swaps in $w_S(i, j, x_1, y_1)$ for some x_1 and y_1, and similarly, $c^{j,i}$ must lose a score due to swaps in $w_S(i, j, x_2, y_2)$ for some x_2 and y_2. Let $\Gamma_{\text{es}}(i, j)$ be the swaps applied in these two votes. There are three possibilities for $\Gamma_{\text{es}}(i, j)$:

(a) $w_S(i, j, x_1, y_1) = w_S(i, j, x_2, y_2)$, and the swaps in $\Gamma_{\text{es}}(i, j)$ transform the vote $(c^{i,j}, c^{j,i}, a^j_{x_1}, a^i_{y_1}, \dagger)$ into $(a^j_{x_1}, a^i_{y_1}, c^{i,j}, c^{j,i}, \dagger)$ through 4 swaps having total cost 4.

(b) $w_S(i, j, x_1, y_1) \neq w_S(i, j, x_2, y_2)$ and as a result of the swaps in $\Gamma_{\text{es}}(i, j)$, $c^{i,j}$ gets to the third position of $w_S(i, j, x_1, y_1)$, and $c^{j,i}$ gets to the third position of $w_S(i, j, x_2, y_2)$. In this case, $|\Gamma_{\text{es}}(i, j)| \geq 3$ and $c(\Gamma_{\text{es}}(i, j)) \geq 4$.

(c) $w_S(i, j, x_1, y_1) \neq w_S(i, j, x_2, y_2)$ and after the swaps in $\Gamma_{\text{es}}(i, j)$, at least one of $c^{i,j}$ and $c^{j,i}$ is placed on the fourth position in one of the votes $w_S(i, j, x_1, y_1)$ or $w_S(i, j, x_2, y_2)$. This means $|\Gamma_{\text{es}}(i, j)| \geq 4$ and $c(\Gamma_{\text{es}}(i, j)) \geq 5$.

From the above discussion, the cost of the swaps in $\Gamma_{\text{es}}(i, j)$ is at least 4. Moreover, as a result of the swaps in $\Gamma_{\text{es}}(i, j)$, the candidates in $a^j_{x_1}, a^i_{y_1}, a^j_{x_2}, a^i_{y_2}$ receive a total of 2 additional scores with respect to their initial score in W.

Let A^* denote those candidates in $\bigcup_{i \in [k]} A_i$ which receive an additional score as a result of the swaps in Γ_{vs} or in $\Gamma_{\text{es}}(i, j)$ for some $i < j$. The total score gained by the candidates in A^* during these swaps is exactly k^2. Since the initial score of each candidate in A^* is K, we know that the remaining swaps of Γ must force these candidates to lose a total of k^2 scores. Observe that this can only happen through swaps applied in W_C, and moreover, each candidate can lose at most one score with such swaps. This implies $|A^*| = k^2$.

Let Γ_c be the set of swaps in Γ applied in W_C, transforming W_C into a set of votes W'_C. The above discussion yields that $\text{score}(a, W_C) > \text{score}(a, W'_C)$ holds for each $a \in A^*$. Since Γ is a solution, we also obtain that $\text{score}(q, W_C) \leq \text{score}(q, W'_C)$ must hold for each $q \in \bigcup_{i \in [k]} B_i \cup G$. We will prove the following claim below.

Claim. $c(\Gamma_c) \geq 4k^2$, and equality can only be reached if

$$\{a^j_x \mid j \in [k]\} \cap A^* = \emptyset \text{ or } \{a^j_x \mid j \in [k]\} \subseteq A^* \text{ holds for each } x \in V. \tag{1}$$

Using this claim, $c(\Gamma) = c(\Gamma_{\text{vs}}) + \sum_{i<j} c(\Gamma_{\text{vs}}(i, j)) + c(\Gamma_c) \geq k + 4\binom{k}{2} + 4k^2 = 6k^2 - k = \beta$ follows. Thus, equalities must hold everywhere, resulting in the following consequences.

First, (1) implies that A^* is the union of sets of the form $\{a^1_x, a^2_x, \ldots a^k_x\}$ for exactly k vertices x. By $a^i_{\sigma(i)} \in A^*$, this yields $A^* = \bigcup_{i,j \in [k]} \{a^j_{\sigma(i)}\}$. Recall that by our construction of the votes $w_S(i, x)$, we know $\sigma(i) \in V_i$ for each i.

Second, note that $c(\Gamma_{\text{es}}(i, j)) = 4$ shows that case (c) cannot happen for the swaps $\Gamma_{\text{es}}(i, j)$. Moreover, from (1) we have $|A^* \cap A_i| = k$ for each $i \in [k]$, which implies that case (b) can neither happen. Thus, the only possibility is case (a), meaning that the swaps of $\Gamma_{\text{es}}(i,j)$ transform the vote $(c^{i,j}, c^{j,i}, a_x^j, a_y^i, \dagger)$ for some x and y into a vote $(a_x^j, a_y^i, c^{i,j}, c^{j,i}, \dagger)$. However, by the definition of $w_S(i,j,x,y)$ we know $x \in V_i$, $y \in V_j$, and $xy \in E$. But from $A^* = \bigcup_{i,j \in [k]} \{a_{\sigma(i)}^j\}$, we get that only $x = \sigma(i)$ and $y = \sigma(j)$ is possible. Hence, $\sigma(i)$ and $\sigma(j)$ are neighboring for each $i < j$, proving the first direction.

Before proving the other direction, it remains to show our claim. Let us fix some $x \in V$, and let us suppose $\{a_x^j \mid j \in [k]\} \cap A^* \neq \emptyset$. Let $|A^* \cap \{a_x^j \mid j \in [k]\}| = a_x^*$, and let $c(i)$ be the total cost of the swaps in Γ_c applied to $w_C(i,x)$. We are trying to show that $\sum_{i \in [k]} c(i) \geq 4a_x^*$ and equality implies $a_x^* = k$.

Recall that a_x^i appears only in the vote $w_C(i,x) = (a_x^i, b_x^{i-1}, b_x^i, *, \dagger)$ in W_C. We will use 0-1 variables α_i and β_i to denote whether the score of a_x^i and b_x^i, respectively, are changed in $w_C(i,x)$ as a result of the swaps in Γ_c. The following are elementary observations (sometimes we also use that Γ_c is of minimum cost, and we take $i-1$ modulo k):

1. If $\alpha_i = 1$ and $\beta_i = 0$ then $c(i) = 5$. (In this case, $\beta_{i-1} = 0$ must hold.)
2. If $\alpha_i = 0$ and $\beta_i = 1$ then $c(i) = 1$. (In this case, $\beta_{i-1} = 1$ must hold.)
3. If $\alpha_i = 0$, $\beta_i = 0$, and $\beta_{i-1} = 0$ then $c(i) = 0$.
4. If $\alpha_i = 0$, $\beta_i = 0$, and $\beta_{i-1} = 1$ then $c(i) = 3$.
5. If $\alpha_i = 1$, $\beta_i = 1$, and $\beta_{i-1} = 0$ then $c(i) = 3$.
6. If $\alpha_i = 1$, $\beta_i = 1$, and $\beta_{i-1} = 1$ then $c(i) = 4$.
7. If $\beta_i = 0$ and $\beta_{i-1} = 1$, then $\alpha_i = 1$ is not possible.

First, note that if $\beta_i = 1$ for every $i \in [k]$, then $\sum_{i \in [k]} c(i) = 4a_x^* + (k - a_x^*)$ follows directly by 2 and 6 above. Thus, $\sum_{i \in [k]} c(i) \geq 4a_x^*$ holds, and equality indeed implies $a_x^* = k$.

Otherwise, let us call a maximal series of indices $i, i+1, \dots, j$ in $[k]$ a *segment*, if $\beta_i = \beta_{i+1} = \cdots = \beta_{j-1} = 1$ but $\beta_j = 0$. We think of such series in a cyclic manner, so $i > j$ is possible. First, observe that the cycle $1, 2, \dots, k$ can be decomposed into a certain number of segments and a remaining set H of indices h for which $\beta_h = \beta_{h-1} = 0$. Let us write $I^* = \{i \mid a_x^i \in A^*\}$ for the set of indices associated with A^*. From claims 1 and 3, we know $\sum_{h \in H} c(h) = 5|I^* \cap H|$.

Now, consider a segment $i, i+1, \dots, j$, and let S denote the set of its elements. By claims 7 and 4 we get $\alpha_j = 0$ and $c(j) = 3$. Since case 5 above can only apply for i, by an easy calculation we obtain $\sum_{h \in S} c(h) \geq c(j) + \sum_{h \in S \cap I^*} c(h) > 4|S \cap I^*|$. Taking into account all segments together with the set H, we get $\sum_{i \in [k]} c(i) > 4a_x^*$. From this, the claim follows.

Direction $\Longrightarrow$. Let $\sigma(1), \sigma(2), \dots, \sigma(k)$ form a k-clique in F where $\sigma(i) \in V_i$ for each i. It is straightforward to check that the following swaps of total cost β yield a solution for I_F:
1. For each $i \in [k]$, swap $c^{i,i}$ with $a_{\sigma(i)}^i$ in $w_S(i, \sigma(i))$.
2. For each $i < j$, swap both $c^{i,j}$ and $c^{j,i}$ with both $a_{\sigma(i)}^j$ and $a_{\sigma(j)}^i$ in $w_S(i,j,\sigma(i),\sigma(j))$.
3. For each $i, j \in [k]$, swap both $a_{\sigma(i)}^j$ and $b_{\sigma(i)}^{j-1}$ with $b_{\sigma(i)}^j$ and the dummy in $w_C(i, \sigma(i))$. $\square$

Looking into the proof of Theorem 3, we can see that the results hold even if the costs are uniform in the sense that swapping two given candidates has the same price in any vote, and the maximum number of swaps allowed in a vote is at most 4. By applying minor modifications to the given reduction, Theorem 3 can be generalized to hold for the following modified versions as well.

- If all costs are in $\{\delta_1, \delta_2\}$ such that $\delta_2 \geq 2\delta_1 > 0$: we only have to replace costs 1 and 2 with new costs δ_1 and δ_2, respectively.
- If we want p to be the unique winner: we only have to set $\text{score}(p, W) = K + 1$.
- If we use k-approval for some $3 \leq k \leq |C| - 2$ instead of 2-approval: it suffices to insert $k-2$ dummies into the first $k-2$ positions of each vote.

Hence, Theorem 3 shows that Swap Bribery remains hard even if we consider such natural parameters as the maximum number of swaps allowed in a vote, the maximum number of different possible costs, or the maximum ratio of two different costs to have a fixed value.

4 Other parameterizations

In this section, we will consider different kinds of parameterizations. First, we will look at the parameter 'number of candidates'. For this case, the following observation is helpful.

Let $S_m = \{\pi_1, \pi_2, \ldots, \pi_{m!}\}$ be the set of permutations of size m. We say that an election system is *described by linear inequalities*, if for a given set $C = \{c_1, c_2, \ldots, c_m\}$ of candidates it can be characterized by $f(m)$ sets $A_1, A_2, \ldots A_{f(m)}$ (for some computable function f) of linear inequalities over $m!$ variables $x_1, x_2, \ldots, x_{m!}$ in the following sense: if n_i denotes the number of those votes in a given election E that order C according to π_i, then the first candidate c_1 is a winner of the election if and only if for at least one index i, the setting $x_j = n_j$ for each j satisfies all inequalities in A_i.

It is easy to see that many election systems can be described by linear inequalities: any system based on scoring rules, Copeland$^\alpha$ $(0 \leq \alpha \leq 1)$, Maximin, Bucklin, Ranked pairs.

Theorem 4. Swap Bribery *is FPT if the parameter is the number of candidates, for any election system described by linear inequalities.*

Proof. Let $C = \{c_1, c_2, \ldots, c_m\}$ be the set of candidates given, and let $A_1, A_2, \ldots A_{f(m)}$ be the sets of linear inequalities over variables $x_1, \ldots, x_{m!}$ that describe the given election system $\mathcal{E}$. For some $\pi_i \in S_m$, let v_i denote the vote that ranks C according to π_i. We describe the given set V of votes by writing n_i for the multiplicity of the vote v_i in V.

Our algorithm solves $f(m)$ integer linear programs with variables $T = \{t_{i,j} \mid i \neq j, 1 \leq i, j \leq m!\}$. We will use $t_{i,j}$ to denote the number of votes v_i that we transform into votes v_j; we will require $t_{i,j} \geq 0$ for each $i \neq j$. Let V^T denote the set of votes obtained by transforming the votes in V according to the variables $t_{i,j}$ for each $i \neq j$. Such a transformation from V is feasible if $\sum_{j \neq i} t_{i,j} \leq n_i$ holds for each $i \in [m!]$ (inequality $\mathcal{A}$).

By an observation in [10], we can compute the price $c_{i,j}$ of transforming the vote v_i into v_j in $O(m^3)$ time. Transforming V into V^T can be done with total cost at most β, if $\sum_{i,j \in [m!]} t_{i,j} c_{i,j} \leq \beta$ (inequality $\mathcal{B}$).

We can express the multiplicity x'_i of the vote v_i in V^T as $x'_i = n_i + \sum_{j \neq i} t_{j,i} - \sum_{i \neq j} t_{i,j}$. For some $i \in [f(m)]$, let A'_i denote the set of linear inequalities over the variables in T that are obtained from the linear inequalities in A_i by substituting x_i with the above given expression for x'_i. Using the description of $\mathcal{E}$ with the given linear inequalities, we know that the preferred candidate c_1 wins in $(V^T, C, \mathcal{E})$ for some values of the variables $t_{i,j}$ if and only if these values satisfy the inequalities of A'_i for at least one $i \in [f(m)]$. Thus, our algorithm solves Swap Bribery by finding a non-negative assignment for the variables in T that satisfies both the inequalities $\mathcal{A}$, $\mathcal{B}$, and all inequalities in A'_i for some i.

Solving such a system of linear inequalities can be done in linear FPT time, if the parameter is the number of variables [19]. By $|T| = (m! - 1)m!$ the theorem follows. □

Similarly, we can also show fixed-parameter tractability for other problems if the parameter is the number of candidates, for example for Possible Winner (this result was already obtained for several election systems by Betzler et al., [3]), Manipulation (both for weighted and unweighted voters), several variations of Control (this result was already obtained for Llull and Copeland voting by Faliszewski et al., [13]), or Lobbying [7] (here, the parameter would be the number of issues in the election). Since our topic is Swap Bribery, we will not go into detail here.

Finally, we consider a combined parameter and obtain fixed-parameter tractability.

Theorem 5. *If the minimum cost is* 1*, then* Swap Bribery *for k-approval (where k is part of the input) with combined parameter* $(|V|, \beta)$ *admits a kernel with* $O(|V|^2\beta)$ *votes and* $O(|V|^2\beta^2)$ *candidates. Here,* V *is the set of votes and* β *is the budget.*

Proof. Let V, C, $p \in C$, and β denote the set of votes, the set of candidates, the preferred candidate, and the budget given, respectively. The idea of the kernelization algorithm is that not all candidates are interesting for the problem: only candidates that can be moved within the budget β from a zero-position to a one-position or vice versa are relevant.

Let Γ be a set of swaps with total cost at most β. Clearly, as the minimum possible cost of a swap is 1, we know that there are only 2β candidates c in a vote $v \in V$ for which $\text{score}(c, v) \neq \text{score}(c, v^\Gamma)$ is possible, namely, such a c has to fulfill $k - \beta + 1 \leq \text{rank}(c, v) \leq k + \beta$. Thus, there are at most $2\beta|V|$ candidates for which $\text{score}(c, V) \neq \text{score}(c, V^\Gamma)$ is possible; let us denote the set of these candidates by $\widetilde{C}$. Let c^* be a candidate in $C \setminus \widetilde{C}$ whose score is the maximum among the candidates in $C \setminus \widetilde{C}$.

Note that a candidate $c \in C \setminus (\widetilde{C} \cup \{c^*, p\})$ has no effect on the answer to the problem instance. Indeed, if $\text{score}(p, V^\Gamma) \geq \text{score}(c^*, V^\Gamma)$, then the score of c is not relevant, and conversely, if $\text{score}(p, V^\Gamma) < \text{score}(c^*, V^\Gamma)$ then p loses anyway. Therefore, we can disregard each candidate in $C \setminus \widetilde{C}$ except for c^* and p.

The kernelization algorithm constructs an equivalent instance K as follows. In K, nor the budget, nor the preferred candidate will be changed. However, we will change the value of k to be $\beta + 1$, so the kernel instance K will contain a $(\beta + 1)$-approval election[2]. We define the set V_K of votes and the set C_K of candidates in K as follows.

First, the algorithm "truncates" each vote v, by deleting all its positions (together with the candidates in these positions) except for the 2β positions between $k - \beta + 1$ and $k + \beta$. Then again, we shall make use of dummy candidates (see the proof of Theorem 3); we will ensure $\text{score}(d, V^\Gamma) \leq 1$ for each such dummy d. Swapping a dummy with any other candidate will have cost 1 in K. Now, for each obtained truncated vote, the algorithm inserts a dummy candidate in the first position, so that the obtained votes have length $2\beta + 1$. In this step, the algorithm also determines the set $\widetilde{C}$ and the candidate c^*. This can be done in linear time. We denote the votes[3] obtained in this step by V_r. We do not change the costs of swapping candidates of $\widetilde{C} \cup \{c^*, p\}$ in some vote $v \in V_r$.

Next, to ensure that K is equivalent to the original instance, the algorithm constructs a set V_d of votes such that $\text{score}(c, V_r \cup V_d) = \text{score}(c, V)$ holds for each candidate c in $\widetilde{C} \cup \{p, c^*\}$. This can be done by constructing $\text{score}(c, V) - \text{score}(c, V_r)$ newly added votes where c is on the first position, and all the next 2β positions are taken by dummies. This way we ensure $\text{score}(c, V_d) = \text{score}(c, V_d^\Gamma)$ for any set Γ of swaps with total cost at most β.

If D is the set of dummy candidates created so far, then let $C_K = \widetilde{C} \cup \{p, c^*\} \cup D$. To finish the construction of the votes, it suffices to add for each vote $v \in V_r \cup V_d$ the candidates not yet contained in v, by appending them at the end (starting from the $(2\beta+1)$-th position) in an arbitrary order. The obtained votes will be the votes V_K of the kernel.

The presented construction needs polynomial time. Using the above mentioned arguments, it is straightforward to verify that the constructed kernel instance is indeed equivalent to the original one. Thus, it remains to bound the size of K.

Clearly, $|\widetilde{C} \cup \{p, c^*\}| \leq 2|V|\beta + 2$. The number of dummies introduced in the first phase is exactly $|V_r| = |V|$. As the score of any candidate in V is at most $|V|$, the number of votes created in the second phase is at most $(2|V|\beta + 2)|V|$, which implies that the number of dummies created in this phase is at most $(2|V|\beta + 2)|V| \cdot 2\beta$. Therefore, we obtain $|C_K| \leq |V| + (2|V|\beta + 2)(2|V|\beta + 1) = O(|V|^2\beta^2)$, and also $|V_K| \leq (2|V|\beta + 3)|V| = O(|V|^2\beta)$. ◻

[2] We use $\beta + 1$ instead of β to avoid complications with the case $\beta = 0$.

[3] Actually, these vectors are not real votes in the sense that they do not contain each candidate, but at the moment we do not care about this.

Applying similar ideas, a kernel with $(|V|+k)\beta$ candidates is easy to obtain, which might be favorable to the above result in cases where k is small.

5 Conclusion

We have taken the first step towards parameterized and multivariate investigations of Swap Bribery under certain voting systems. We obtained W[1]-hardness for k-approval if the parameter is the budget β, while Swap Bribery could be shown to be in FPT for a very large class of voting systems if the parameter is the number of candidates. This revaluates previous NP-hardness results: Swap Bribery could be computed efficiently if the number of candidates is small, which is a common setting, e.g. in presidential elections.

However, we have shown this via an integer linear program formulation, using a result by Lenstra, which does not provide running times that are suitable in practice. Here, it would be interesting to give combinatorial algorithms that compute an optimal swap bribery. This might be particularly relevant for a scenario described by Elkind et al. [10], where bribery is not necessarily considered as an undesirable thing, like in the case of campaigning.

As Elkind et al. [10] pointed out, it would be nice to characterize further natural polynomial-time solvable cases of Swap Bribery. We provided one such example with Theorem 1 for k-approval in the case where costs are equal to 1. By contrast, already the case of two different costs δ_1, δ_2 with $\delta_2 \geq 2\delta_1 > 0$ becomes NP-complete for k-approval ($2 \leq k \leq m-2$) and W[1]-hard if the parameter is the budget β. We believe that this can be generalized to the case of two different (arbitrary) positive costs.

There are plenty of possibilities to carry on our initiations. First, there are more parameterizations to be looked at, and in particular the study of combined parameters in the spirit of Niedermeier [22], see e.g. [1], is an interesting approach.

Also, we have focused our attention to k-approval, but the same questions could be studied for other voting systems, or for the special case of Shift Bribery which was shown to be NP-complete for several voting systems [10], or other variants of the bribery problem as mentioned in the introduction. For instance, we have only looked at *constructive* swap bribery, but the case of *destructive* swap bribery (when our aim is to achieve that a disliked candidate does *not* win) is worth further investigation as well.

Acknowledgments. We thank Rolf Niedermeier for an inspiring initial discussion on this topic.

References

[1] N. Betzler. On problem kernels for possible winner determination under the k-approval protocol. In *Proc. of 35th MFCS*, 2010.

[2] N. Betzler and B. Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *J. Comput. Syst. Sci.*, In Press, 2010.

[3] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proc. of 21st IJCAI*, pages 53–58, 2009.

[4] N. Betzler and J. Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009.

[5] H. L. Bodlaender. Kernelization: New upper and lower bound techniques. In *IWPEC*, pages 17–37, 2009.

[6] S. J. Brams and P. C. Fishburn. Voting procedures. In *Handbook of Social Choice and Welfare*, volume 1, pages 173–236. Elsevier, 2002.

[7] R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On complexity of lobbying in multiple referenda. *Review of Economic Design*, 11(3):217–224, November 2007.

[8] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *J. ACM*, 54(3):1–33, 2007.

[9] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[10] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proc. of 2nd SAGT*, volume 5814 of *LNCS*, pages 299–310. Springer, 2009.

[11] P. Faliszewski. Nonuniform bribery. In *Proc. 7th AAMAS*, pages 1569–1572, 2008.

[12] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. The complexity of bribery in elections. In *Proc. of 21st AAAI*, pages 641–646, 2006.

[13] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res. (JAIR)*, 35:275–341, 2009.

[14] M. R. Fellows, D. Hermelin, F. A. Rosamond, and S. Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.

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

[16] J. Guo and R. Niedermeier. Invitation to data reduction and problem kernelization. *SIGACT News*, 38(1):31–45, 2007.

[17] E. Hemaspaandra and L. A. Hemaspaandra. Dichotomy for voting systems. *J. Comput. Syst. Sci.*, 73(1):73–83, 2007.

[18] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. of IJCAI-2005 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[19] H. Lenstra. Integer programming with a fixed number of variables. *Math. of OR*, 8:538–548, 1983.

[20] H. Liu, H. Feng, D. Zhu, and J. Luan. Parameterized computational complexity of control problems in voting systems. *Theor. Comput. Sci.*, 410(27-29):2746–2753, 2009.

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

[22] R. Niedermeier. Reflections on multivariate algorithmics and problem parameterization. In *Proc. of 27th STACS*, pages 17–32, 2010.

Britta Dorn
Wilhelm-Schickard-Institut für Informatik, Universität Tübingen
Sand 13, 72076 Tübingen, Germany
Email: bdorn@informatik.uni-tuebingen.de

Ildikó Schlotter
Budapest University of Technology and Economics
H-1521 Budapest, Hungary
Email: ildi@cs.bme.hu