

Winner Robustness via Swap- and Shift-Bribery: Parameterized Counting Complexity and Experiments

Niclas Boehmer¹, Robert Brederick², Piotr Faliszewski³ and Rolf Niedermeier¹

¹Algorithmics and Computational Complexity, TU Berlin, Germany

²Humboldt-Universität zu Berlin, Germany

³AGH University, Poland

niclas.boehmer@tu-berlin.de, robert.brederick@hu-berlin.de, faliszew@agh.edu.pl,
rolf.niedermeier@tu-berlin.de

Abstract

We study the parameterized complexity of counting variants of SWAP- and SHIFT-BRIBERY, focusing on the parameterizations by the number of swaps and the number of voters. Facing several computational hardness results, using sampling we show experimentally that SWAP-BRIBERY offers a new approach to the robustness analysis of elections.

1 Introduction

Consider a university department which is about to hire a new professor. There are m candidates and the head of the department decided to choose the winner by Borda voting. Each faculty member (i.e., each voter) ranked the candidates from the most to the least appealing one, each candidate received $m - i$ points for each vote where he or she was ranked as the i -th best, and the candidate with the highest score was selected. However, after the results were announced, some voters started wondering if, perhaps, some *other* voters accidentally “misranked” some of the candidates (worrying about mistakes in the votes is an old democratic tradition). For instance, if some voter viewed two candidates as very similar, then he or she could have ranked them either way, depending on an impulse. Or, some voter would have ranked two candidates differently if he or she had more information on their merits. It is, thus, natural to ask for the probability of changing the election outcome by making some random swaps. Indeed, this approach was recently pursued by Baumeister and Högbe [2020] and we follow-up on it, but with a different focus (see the discussion of related work).

Specifically, in our model for each $r \in \mathbb{N}$ and each candidate c , we let $P_c(r)$ be the probability that c wins an election obtained by making r random swaps of candidates ranked on adjacent positions in the votes (we refer to such elections as being at swap distance r from the original one). Such values can be quite useful. For example, if for each r we had (some estimate of) the probability that in total there are r accidental swaps in the votes, then we could compute the probability of each candidate’s victory. If it were small for the original winner, then we might want to recount the votes or reexamine the election process. The values $P_c(r)$ are also useful without the distribution of r ’s. For example, we may want to find the smallest number of swaps for which the probability of

the original winner’s victory drops below some value (such as 50%) or for which he or she is no longer the most probable winner. As we show in our experiments, this approach provides new insights on the robustness of election results.

To determine the value $P_c(r)$, we need to divide the number of elections at swap distance r where c wins, by the total number of elections at this distance. While computing the latter is easy—at least in the sense that there is an efficient algorithm for this task—computing the former requires solving the counting variant of the SWAP-BRIBERY problem (denoted #SWAP-BRIBERY). Briefly put, in the decision variant of the problem, we ask if it is possible to ensure that a designated candidate wins a given election by making r swaps of adjacent candidates in the votes (we assume the unit prices setting; see Section 2). In the counting variant, we ask how many ways there are to achieve this effect (using exactly r swaps). Unfortunately, already the decision variant is NP-hard for many voting rules, and we show that the counting one is hard even for Plurality. On the positive side, we can get a good estimate of $P_c(r)$ by sampling (see Footnote 2).

We also consider the SHIFT-BRIBERY problem, a variant of SWAP-BRIBERY where we can only shift the designated candidate forward (in the constructive case) or backward (in the destructive one, where the goal is to ensure that the designated candidate loses). These problems also can be used to evaluate robustness of election results but, to maintain focus, in our experiments we only consider SWAP-BRIBERY. Yet, we include SHIFT-BRIBERY in our complexity analysis because it illustrates some interesting phenomena.

Main contributions. We focus on #SWAP- and #SHIFT-BRIBERY for the Plurality and Borda voting rules (for unit prices). We consider their computational complexity for parameterizations by the number of unit swaps/shifts (which we refer to as the swap/shift radius) and by the number of voters (see Table 1). We also present experiments, where we use #SWAP-BRIBERY to evaluate the robustness of election results. Our main results are as follows:

1. For Plurality, SWAP-BRIBERY is known to be in P, but we show that the counting variant is #W[1]-hard when parameterized by the swap radius.
2. For Borda, hardness results for #SWAP-BRIBERY follow from those for #SHIFT-BRIBERY, which themselves are intriguing: E.g., the destructive variant parameterized by

the shift radius is $\#W[1]$ -hard, but the constructive one is in FPT; in the decision setting the former is easier.¹

- Using sampling, we estimate the candidate’s winning probabilities in elections from a dataset generated by Szufa *et al.* [2020]. One of the high-level conclusions is that the score differences between the election winners and the runners-up can be quite disconnected from their strengths (measured using $\#SWAP$ -BRIBERY).

We defer several proofs (marked \star) and analyses to a full version available at arxiv.org/abs/2010.09678.

Related work. Our work is most closely related to the papers of Hazon *et al.* [2012], Bachrach *et al.* [2010], and Baumeister and Hogebe [2020]. Similarly to our approach, their authors study the complexity of computing the probability that a given candidate wins, provided that the votes may change according to some probability distribution. In particular, Hazon *et al.* [2012] assume that each voter is endowed with an explicitly encoded list of possible votes, each with its own probability of being cast, Bachrach *et al.* [2010] consider elections where the votes are partial and all completions are equally likely, and Baumeister and Hogebe [2020] consider both these models, as well as a third one, which—in terms of computational complexity—is equivalent to our model for swap bribery.

There are two methodological differences between our work and the three above-discussed papers. Foremost, we provide an experimental analysis showing that counting variants of $SWAP$ -BRIBERY are indeed helpful for evaluating the robustness of election winners. In contrast, Bachrach *et al.* [2010] and Baumeister and Hogebe [2020] focus entirely on the complexity analysis, whereas Hazon *et al.* [2012] also provide experiments, but their focus is on the running time and memory consumption of their algorithm.

The second difference is that we focus on a parameterized complexity analysis—with an explicit focus on establishing FPT and $\#W[1]$ -hardness results—which was not done in previous works (except for obtaining XP algorithms for fixed numbers of candidates or voters).

Historically, $SWAP$ - and $SHIFT$ -BRIBERY were introduced by Elkind *et al.* [2009]. Various authors studied these problems for different voting rules (see, e.g., the works of Maushagen *et al.* [2018] and Zhou and Guo [2020] regarding iterative elections), sought approximation algorithms [Elkind and Faliszewski, 2010; Faliszewski *et al.*, 2019], established parameterized complexity results [Dorn and Schlotter, 2012; Brederick *et al.*, 2016a; Knop *et al.*, 2020], considered restricted preference domains [Elkind *et al.*, 2020], and extended the problem in various ways [Brederick *et al.*, 2016b; Kaczmarczyk and Faliszewski, 2019; Baumeister *et al.*, 2019; Yang *et al.*, 2019]. The idea of using $SWAP$ -BRIBERY to measure the robustness of election results is due to Shiryaev *et al.* [2013], but is also closely related to computing the margin of victory [Magrino *et al.*, 2011; Cary, 2011; Xia, 2012; Brill *et al.*, 2020]; recently it was also applied to committee elections [Brederick *et al.*, 2021].

¹We abuse notation here by using P and FPT for also referring to algorithmically positive counting complexity results although formally these classes are only defined for decision problems.

	Plurality		Borda	
	decision	counting	decision	counting
$SWAP$ -BRIBERY	P	$\#P$ -hard $\#W[1]$ -h.(r) FPT(n)	NP-hard FPT(r) $W[1]$ -h.(n)	$\#P$ -hard ? $\#W[1]$ -h.(n)
CONST. SHIFT-BRIBERY	P	P	NP-hard FPT(r) $W[1]$ -h.(n)	$\#P$ -hard FPT(r) $\#W[1]$ -h.(n)
DEST. SHIFT-BRIBERY	P	P	P	$\#P$ -hard $\#W[1]$ -h.(r) $\#W[1]$ -h.(n)

Table 1: (Parameterized) complexity of $SWAP$ - and $SHIFT$ -BRIBERY with unit prices; r and n refer to the parameterizations by the swap/shift radius and by the number of voters, respectively. Results for the counting variants are new (however, see also the work of Baumeister and Hogebe [2020] for results related to the $\#P$ -hardness of PLURALITY $SWAP$ -BRIBERY); results for the decision variants are due to Elkind *et al.* [2009], Brederick *et al.* [2016b], Brederick *et al.* [2016a], and Kaczmarczyk and Faliszewski [2019].

So far, the complexity of counting problems received limited attention in the context of elections. In addition to the works of Hazon *et al.* [2012], Bachrach *et al.* [2010], and Baumeister and Hogebe [2020], we mention two more: Wojtas and Faliszewski [2012] studied counting solutions for control problems, and Kenig and Kimelfeld [2019] followed up on the work of Bachrach *et al.* [2010] and provided approximation algorithms for their setting.

2 Preliminaries

For each integer k , by $[k]$ we mean the set $\{1, \dots, k\}$.

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 v_i has a preference order, which ranks all candidates from the most to the least desired one (we sometimes refer to preference orders as votes). For a voter v_i , we write $v_i: c_1 \succ c_2 \succ \dots \succ c_m$ to indicate that he or she ranks c_1 first, then c_2 , and so on. If we put a subset of candidates in such a description of a preference order, then we mean listing its members in an arbitrary order.

Voting rules. A voting rule \mathcal{R} is a function that, given an election, returns a set of candidates that tie as winners. We focus on Plurality and Borda, which assign scores to the candidates and select those with the highest ones. Under Plurality, each voter gives one point to the top-ranked candidate. Under Borda, each voter gives $|C| - 1$ points to the top-ranked candidate, $|C| - 2$ points to the next one, and so on. We write $\text{score}_E(c)$ to denote the score of candidate c in election E (the voting rule will be clear from the context).

Swap distance. Let u and v be two votes over the same candidate set. The swap distance between u and v , denoted $d_{\text{sw}}(u, v)$, is the length of the shortest sequence of swaps of adjacent candidates whose application transforms u into v . Given elections $E = (C, V)$ and $E' = (C, V')$, where $V = (v_1, \dots, v_n)$ and $V' = (v'_1, \dots, v'_n)$, their swap distance is $\sum_{i=1}^n d_{\text{sw}}(v_i, v'_i)$. By $R(E, r)$, we denote the set of

elections that are at swap distance r from E .

Swap- and Shift-Bribery. Let \mathcal{R} be a voting rule. In the decision variant of the \mathcal{R} SWAP-BRIBERY problem, we are given an election E , a designated candidate p , and a budget r . Further, for each voter and each two candidates c and d , we have a nonnegative price $\pi_v(c, d)$ for swapping them in v 's preference order (a swap is legal if at the time of its application c and d are adjacent). We ask if there is an election E' where p is an \mathcal{R} -winner, such that E' can be obtained from E by performing a sequence of legal swaps of cost at most r . In the counting variant, we ask for the number of such elections, and we require the cost of swaps to be exactly r (the last condition is for our convenience and all our results would still hold if we asked for cost at most r ; the same would be true if instead of counting elections where p won, we would count those where he or she lost). Since we are interested in computing the candidates' probabilities of victory in elections at a given swap distance, we focus on the case where each swap has the same, unit price.

CONSTRUCTIVE SHIFT-BRIBERY is a variant of SWAP-BRIBERY where all swaps must involve the designated candidate, shifting him or her forward. DESTRUCTIVE SHIFT-BRIBERY is defined analogously, except that our goal is to preclude the designated candidate's victory, and we can only shift him or her backward [Kaczmarczyk and Faliszewski, 2019]. Counting variants are defined in a natural way. We focus on the case where each unit shift has a unit price and, in general, speak of shift or swap radius r instead of budget.

Counting complexity. We assume basic familiarity with (parameterized) complexity theory, including the classes P, NP, FPT, and W[1], and reducibility notions.

Let X be a decision problem from NP, where for each instance we ask if there exists some mathematical object with a given property. In its counting variant, traditionally denoted # X , we ask for the number of such objects. For example, in MATCHING we are given an integer k and a bipartite graph G —with vertex set $U(G) \uplus V(G)$ and edge set $E(G)$ —and we ask if G contains a matching of size k (i.e., a set of k edges, where no two edges touch the same vertex). In #MATCHING we ask how many such matchings exist.

The class #P is the counting analog of NP; a problem belongs to #P if it can be expressed as the task of counting the number of accepting computations of a nondeterministic polynomial-time Turing machine. We say that a counting problem #A (polynomial-time) Turing reduces to #B if there exists an algorithm that solves #A in polynomial time, provided that it has oracle access to #B. A problem is #P-hard if every problem from #P Turing reduces to it. While MATCHING is in P, it is well known that #MATCHING is #P-hard and #P-complete [Valiant, 1979].

#W[1] relates to W[1] in the same way as #P relates to NP. As examples of #W[1]-hard problems, we mention counting size- k cliques in a graph, parameterized by k [Flum and Grohe, 2004] and #MATCHING, parameterized by the size of the matching [Curticapean and Marx, 2014]. Formally, #W[1]-hardness is defined using a slightly more general notion of a reduction, but for our purposes polynomial-time Turing reductions (where the parameters in the queried

instances are bounded from above by a function of the parameter in the input instance) will suffice.

3 Algorithms and Complexity Results

In this section, we present our results regarding the complexity of #SWAP- and #SHIFT-BRIBERY. We first consider Plurality, mostly focusing on #SWAP-BRIBERY, and then discuss Borda, mostly focusing on #SHIFT-BRIBERY.

3.1 Plurality and #SWAP-BRIBERY

We start with a hardness result. While there is a polynomial-time algorithm for the decision variant of PLURALITY SWAP-BRIBERY [Elkind *et al.*, 2009], the counting variant is intractable, even with unit prices (#P-hardness for a slightly different but computationally equivalent model is also reported by Baumeister and Hogrebe [2020]).

Theorem 1. PLURALITY #SWAP-BRIBERY is #P-hard and #W[1]-hard when parameterized by the swap radius, even for unit prices.

Proof. We give a reduction from #MATCHING. We will use a swap radius upper-bounded by a function of the desired matching size, so we obtain both #P- and #W[1]-hardness.

Let (G, k) be an instance of #MATCHING, where G is a bipartite graph with vertex set $U(G) \uplus V(G)$ and k is the size of matchings that we are to count. Assume that $U(G) = \{u_1, \dots, u_n\}$, $V(G) = \{v_1, \dots, v_n\}$, and $k \leq n$. To form an election, we let the candidate set be $C := U(G) \uplus V(G) \uplus \{p, a, b\} \uplus X$, where $X := \{x_1, \dots, x_{3k+1}\}$. The candidates in $U(G) \uplus V(G)$ will model the graph, p will be our designated candidate, a and b will control the size of the matching, and the candidates in X will block undesirable swaps. We will have the following scores of the candidates:

$$\begin{aligned} \forall c \in C \setminus \{a, b\}: \text{score}_E(c) &= n, \\ \text{score}_E(a) &= n - k, \text{score}_E(b) = n + k. \end{aligned}$$

We form the following four groups of voters:

1. For each $\{u_i, v_j\} \in E(G)$, there is an *edge voter* e_{ij} with preference order $e_{ij}: u_i \succ v_j \succ X \succ \dots$.
2. For each $j \in [n]$, we have an *a-voter* a_j with preference order $a_j: v_j \succ a \succ X \succ \dots$.
3. For each $i \in [n]$, we have a *b-voter* b_i with preference order $b_i: b \succ u_i \succ X \succ \dots$.
4. Finally, the *score voters* implement the desired Plurality scores. For each candidate $c \in U(G) \cup V(G) \cup \{p\}$, there are exactly as many voters with preference order $c \succ X \succ \dots$ as necessary to ensure that in total c has score n . Similarly, for each $x_i \in X$ there are n voters with preference order $x_i \succ X \setminus \{x_i\} \succ \dots$. There are also $n - k$ voters with preference order $a \succ X \succ \dots$ and k voters with preference order $b \succ X \succ \dots$.

Let E be an election with the above-described candidates and voters. We form an instance I of PLURALITY #SWAP-BRIBERY with this election, unit prices, and swap radius $r := 3k$. Then, we make an oracle query for I and return its answer. In the remainder of the proof, we argue that this answer

is equal to the number of size- k matchings in G . The idea is that to make p a winner, we have to transfer k points from b to a via swaps that correspond to a matching.

Let E' be some election in $R(E, r)$, i.e., an election at swap distance r from E , where p wins. We note that p and the candidates from X have score n in E' (indeed, in elections from $R(E, r)$, p has score at most n and the average score of the candidates in X is at least n). Further, in E' each edge voter, a -voter, and b -voter either ranks on top the same candidate as in E or the candidate that he or she ranked second in E , and each score voter ranks the same candidate on top as in E (otherwise some candidate in X would have score above n). We call this the *top-two rule*.

Since b must have at most n points in E' , by the top-two rule, there must be at least k b -voters in E' that rank members of $U(G)$ on top. Let U_b be the set of these members of $U(G)$. As each member of $U(G)$ can be swapped with b at most once in the b -votes, we have $|U_b| \geq k$.

Compared to E , in E' each member of U_b gets an additional point from the b -voters. Thus, for each $u_i \in U_b$ there must be a voter that ranked u_i on top in E but does not do so in E' . By the top-two rule, this must be an edge voter. Let M be the set of pairs $\{u_i, v_j\}$ such that in E edge voter e_{ij} ranks u_i on top, but in E' he or she ranks v_j on top. Naturally, we must have $|M| \geq |U_b|$.

For each pair $\{u_i, v_j\} \in M$, there must be a voter who swapped v_j out of the top position in E' , because otherwise v_j would have more than n points. By the top-two rule, this must be voter a_j . Let V_a be the set of those members of $V(G)$ that in E' are swapped out of the top positions in the a -votes. It must be that $|V_a| \geq |M|$.

Altogether, we have $|V_a| \geq |M| \geq |U_b| \geq k$ and, in fact, each of these sets must have exactly k elements (because their elements correspond to unique swaps). Further, M is a matching. If it were not, then some member of $U(G) \uplus V(G)$ would appear in two pairs in M , but then we would have to have two a -voters or two b -voters that corresponded to this candidate, which is not possible in our construction.

This way we have shown that for each election in $R(E, r)$ where p wins, there is a corresponding size- k matching. As the other direction is immediate, the proof is complete. \square

A natural way to circumvent such intractability results is to seek FPT algorithms parameterized by the number of candidates or by the number of voters. For the former, one typically expresses SWAP-BRIBERY problems as integer linear programs (ILPs) and invokes the classic algorithm of Lenstra, Jr. [1983], or some more recent one; see, e.g., the work of Knop *et al.* [2020]. Unfortunately, in case of counting there are two issues. First, counting analogues of these algorithms, dating back to the seminal work of Barvinok [1994], have XP running times. However, fortunately, the ILPs used for SWAP-BRIBERY have such a special form that in their case Barvinok's algorithm would run in FPT time for the parameterization by the number of candidates. The second obstacle is more serious. Even though we could count the number of solutions to our ILPs, each of these solutions would potentially correspond to a different number of solutions for SWAP-BRIBERY. Dealing with this problem, so far, remains elusive

and we leave it as an open problem. Yet, for unit prices we do show an FPT algorithm parameterized by the number of voters.

Theorem 2 (★). *For unit prices, PLURALITY #SWAP-BRIBERY parameterized by the number of voters is in FPT.*

The restriction to unit prices in Theorem 2 is necessary. Otherwise, a reduction from the problem of counting linear extensions of a partially ordered set [Brightwell and Winkler, 1991] shows #P-hardness even for a single voter.

Theorem 3 (★). *PLURALITY #SWAP-BRIBERY is #P-hard even for a single voter and unary-encoded prices.*

We conclude with a brief mention of #SHIFT-BRIBERY. Both the constructive and the destructive variant are in P, even with arbitrary unary-encoded prices (for the binary encoding, #P-hardness follows by a reduction from #PARTITION). Our algorithms use dynamic programming over groups of voters with the same candidate as their top choice.

Theorem 4 (★). *For unary-encoded prices, both the constructive and the destructive variant of PLURALITY #SHIFT-BRIBERY are in P.*

3.2 Borda and #SHIFT-BRIBERY

Our results for BORDA #SWAP-BRIBERY follow from those for #SHIFT-BRIBERY, so we focus on the latter problem.

In the decision setting, the constructive variant of BORDA SHIFT-BRIBERY is NP-hard (and is in FPT when parameterized by the shift radius, but is W[1]-hard for the number of voters), whereas the destructive variant is in P. In the counting setting, both variants are #P-hard and #W[1]-hard for the parameterization by the number of voters; the result for the constructive case follows from a proof for the decision variant due to Bredebeck *et al.* [2016b] and for the destructive case, we use a similar approach with a few tricks on top.

Theorem 5 (★). *Both the constructive and the destructive variant of BORDA #SHIFT-BRIBERY are #P-hard and #W[1]-hard when parameterized by the number of voters.*

Surprisingly, when parameterized by the shift radius, the constructive variant is in FPT and the destructive variant is #W[1]-hard. Not only does the problem that was easier in the decision setting now become harder, but also—to the best of our knowledge—it is the first example where a destructive variant of an election-related problem is harder than the constructive one. Yet, SHIFT-BRIBERY is quite special as the two variants differ in the available actions, i.e., either shifting the designated candidate forward or backward (typically, destructive voting problems have the same sets of actions as the constructive ones).

The FPT algorithm for the constructive case relies on the fact that if we can ensure victory of the designated candidate by shifting him or her by r positions forward, then there are at most r candidates that we need to focus on (the others will be defeated irrespective of what exact shifts we make). There are no such bounds in the destructive setting.

Theorem 6 (★). *Parameterized by the shift radius, BORDA #CONSTRUCTIVE SHIFT-BRIBERY is in FPT (for unary-encoded prices), but the destructive variant is #W[1]-hard, even for unit prices.*

Proof sketch (destructive case). We give a polynomial-time Turing reduction from #MATCHING to BORDA #DESTRUCTIVE SHIFT-BRIBERY. Let (G, k) be an instance of #MATCHING, where G is a bipartite graph with vertex set $U(G) \uplus V(G)$, and k is a positive integer. Assume that $U(G) = \{u_1, \dots, u_n\}$, $V(G) = \{v_1, \dots, v_n\}$, and $k \leq n$.

Our reduction proceeds as follows. First, we form the set of relevant candidates $R := \{d, p\} \uplus U(G) \uplus V(G)$, where d is the designated candidate. Moreover, for each relevant candidate $r \in R$, we form a set $D(r)$ of $3k + 1$ dummy ones. We will form an election E , where these candidates will have the following Borda scores (X is some positive integer, whose value depends on the specifics of the construction; we will be counting ways in which d can cease to be a winner by shifting him or her backward by $3k$ positions):

$$\begin{aligned} \text{score}_E(d) &= X + 3k, & \text{score}_E(p) &= X - k + 1, \\ \text{score}_E(u_1) &= \dots = \text{score}_E(u_n) = X - 1, \\ \text{score}_E(v_1) &= \dots = \text{score}_E(v_n) = X - 1, \text{ and} \\ & \text{each dummy candidate has score at most } X - 3k - 1. \end{aligned}$$

Election E contains the following voters:

1. For each edge $e = \{u_i, v_j\}$ of the input graph, there is an *edge voter* v_e with preference order $v_e: d \succ u_i \succ v_j \succ p \succ D(p) \succ \dots$.
2. There is a group of voters who ensure that the scores are as described above. They rank at least $3k$ dummy candidates between each two relevant ones (so shifting d by $3k$ positions back cannot change the score of another relevant candidate).

Next, we form an election F identical to E , except that one of the edge voters ranks p one position lower (so that p 's score in F is $X - k$). Let I_E and I_F be instances of BORDA #DESTRUCTIVE SHIFT-BRIBERY with designated candidate d , shift radius $3k$, unit prices, and elections E and F , respectively. Our reduction queries the oracle for the numbers of solutions for I_E and I_F , subtracts the latter from the former, and outputs this value. We claim that it is exactly the number of size- k matchings in G .

To see why this is the case, consider some solution for I_E . There are two possibilities: Either d passes some member of $U(G) \uplus V(G)$ twice (in which case this candidate gets at least $X + 1$ points, whereas d always gets exactly X points), or d passes each member of $U(G) \uplus V(G)$ at most once. In the latter case, only p can defeat d (all other candidates have at most X points). However, for this to happen, d must pass p exactly k times (with the shift radius of $3k$, d cannot pass p more times). Further, since we assumed that d never passes a member of $U(G) \uplus V(G)$ more than once, the votes where d passes p must correspond to a size- k matching in G . We refer to such solutions as *matching solutions*.

The set of solutions for I_F contains all solutions for I_E except for the matching ones (because in I_F , p ends up with at most X points and not $X + 1$). So, by subtracting the number of solutions for I_F from the number of solutions for I_E , we get exactly the number of size- k matchings in G . \square

For BORDA #SWAP-BRIBERY, we obtain #P-hardness and #W[1]-hardness when parameterized by the number of

voters by noting that the proofs for #SHIFT-BRIBERY still apply in this case. The parameterization by the swap radius remains open, though (the proof of Theorem 6 does not work as many new, hard to control, solutions appear).

Corollary 1. BORDA #SWAP-BRIBERY is #P-hard and #W[1]-hard when parameterized by the number of voters, even for the case of unit prices.

4 Experiments

In the following, we use #SWAP-BRIBERY to analyze the robustness of election winners experimentally. For clarity, in this section we use normalized swap distances, which specify the fraction of all possible swaps in a given election.

We used a dataset of 800 elections, each with 10 candidates and 100 voters, prepared by Szufa *et al.* [2020]. This dataset contains elections generated from various statistical cultures such as the impartial culture, urn, Mallows, tD -Cube, and tD -Sphere model (see our full version for definitions).

For each election E and candidate c , let $P_{E,c}(r)$ be the probability that c wins—under a given voting rule—in an election chosen uniformly at random from $R(E, r)$. Ideally, we would like to compute these values exactly. However, since #SWAP-BRIBERY is #P-hard for both our rules, instead of computing these values exactly, we resorted to sampling. Specifically, for each election (except those with tied winners) and each normalized swap distance $r \in \{0.05, 0.1, \dots, 1\}$ we sampled 500 elections at this distance and for each candidate recorded the proportion of elections where he or she won² (see our full version for a detailed description of the sampling procedure). For each election, we quantify the robustness of its winner by identifying the smallest swap distance r , among the considered ones, for which he or she has a winning probability below 50%. We refer to this value as the *50%-winner threshold*.

We present some of our findings and refer to the full version for more detailed experiments and a deeper analysis. We start by analyzing the relation between the 50%-winner threshold and two other measures of winner robustness, namely, the score difference between the winner and the runner-up (i.e., the candidate ranked in the second place) and the minimum number of swaps of adjacent candidates that are necessary to change the election winner (this is simply the optimal cost of a DESTRUCTIVE SWAP-BRIBERY with unit prices; see the work of Shiryaev *et al.* [2013]³). To this end, let us turn to Figure 1a (for Plurality) and Figure 1b (for Borda), which are both split into a black part (dots) and a red part (pluses). In both parts, each election is represented as a marker of the respective color whose y -coordinate is the score difference between the winner and the runner up, and whose x -coordinate is either:

²By Hoeffding's inequality, the probability that the estimated winning probability for a given candidate deviates by more than 10% from the true one can be upper-bounded by 0.1%.

³Notably, both PLURALITY DESTRUCTIVE SWAP-BRIBERY and BORDA DESTRUCTIVE SWAP-BRIBERY are in P [Shiryaev *et al.*, 2013]. For both problems, it suffices to iterate over all candidates $d \neq c$ (where c is the original winner) and calculate the minimum cost of modifying the election so that d has a higher score than c .

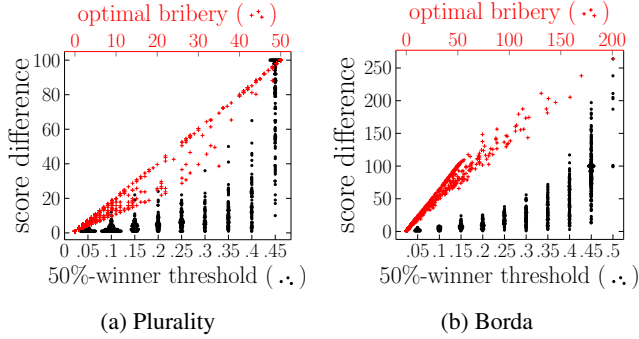


Figure 1: Each election is represented by a red plus and a black dot marker. The y -coordinate of each marker gives the difference between the scores of the winner and the runner-up. The x -coordinate of the black dot gives the 50%-winner threshold (perturbed, if many elections would overlap), while the x -coordinate of the red plus gives the minimum cost of a successful destructive swap bribery.

1. the 50%-winner threshold (for black dots, perturbed a bit if many elections were to take the same place), or
2. the minimum cost of a successful destructive swap bribery (for red pluses).

Finding 1. *The score difference between the winner and the runner-up strongly correlates with the minimum cost of a successful destructive bribery. By contrast, the score difference has a limited predictive value for the 50%-winner threshold.*

Examining the figures, we see that the score difference is very strongly correlated with the minimum cost of a successful destructive swap bribery, but that the correlation between the score difference and the 50%-winner threshold is far less pronounced. Indeed, the same score difference may lead to a wide range of 50%-winner thresholds (e.g., for Plurality a score difference of 10 may lead to the threshold being anything between 0.1 and 0.4). Thus our framework adds a new dimension to the robustness analysis of election winners.

To exemplify this phenomenon, we visualize $P_{E,c}(r)$ under Borda for two particular elections from the 800-elections dataset in Figure 2 (for these two elections, we estimated $P_{E,c}(r)$ for $r \in \{0.0125, 0.025, \dots, 0.5\}$ using 10'000 samples in each case). We want to emphasize that these elections are not artificial extreme examples but were generated both as part of 180 elections generated by the Urn model with different parameters. In fact, there exist several other elections in the dataset exhibiting a similar behavior.

Noticeably, the difference between the Borda score of the red and the blue candidate in both elections is quite similar, i.e., in Election (a) the difference is two and in Election (b) the difference is four. Nevertheless, the robustness of the election winner is quite different. In Election (a), the winner is very sensitive to random swaps: The blue candidate already wins a majority of elections even if only a 0.0125 fraction of all possible swaps are applied (i.e., about half a swap per vote, on average). It is quite surprising that so few random swaps may change the outcome with a fairly high probability. In contrast, in Election (b), the 50%-winner threshold is around swap distance 0.3 and the red candidate even stays the most

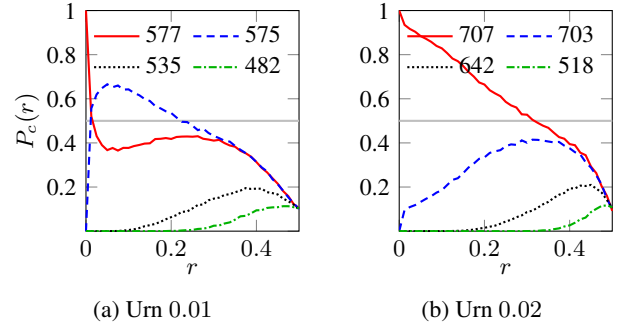


Figure 2: Plots showing $P_{E,c}(r)$ under Borda as a function of r for two selected elections. Each line represents $P_{E,c}(r)$ for one particular candidate. In the legend, the Borda scores of the candidates in the original election are displayed. Only the four candidates with the highest Borda scores are included. Both elections were sampled from the urn model.

probable winner until more than 40% of swaps are performed.

We now turn to the differences between Plurality and Borda when it comes to the robustness of election winners:

Finding 2. *The Borda winner of an election is usually more robust against random swaps than the Plurality winner.*

For Borda, 230 elections out of the 800 considered have a 50%-winner threshold of 0.45, while every other threshold occurs fewer than 90 times. In contrast, for Plurality the distribution is more uniform (with small spikes of around 110 elections at thresholds 0.1 and 0.45). So, Plurality elections are more likely to change results after relatively few swaps than the Borda ones. Two explanations are that (a) under Plurality there can be “strong contenders” who do not win, but who are often ranked close to the first place and, thus, can overtake the original winner after a few swaps, and (b) the Plurality winner has the highest chance of losing points, as he or she is ranked first most often. Under Borda, the candidates usually have similar chances of both gaining and losing a point with a single swap.

5 Conclusions

We have shown that the counting variants of SWAP-BRIBERY have high worst-case complexity, but, nonetheless, are very useful for analyzing the robustness of election winners. In particular, we have observed that the scores of the candidates do not suffice to evaluate their strengths. Establishing the complexity of BORDA #SWAP-BRIBERY parameterized by the swap radius and the complexity of all considered #P-hard problems parameterized by the number of candidates remain as intriguing open problems.

Acknowledgments

NB was supported by the DFG project MaMu (NI 369/19). Work started while PF was visiting TU Berlin based on a Friedrich Wilhelm Bessel Award from the Alexander von Humboldt Foundation. RB was partially supported by the DFG project AFFA (BR 5207/1 and NI 369/15).

References

- [Bachrach *et al.*, 2010] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proceedings of AAAI-2010*, pages 697–702, 2010.
- [Barvinok, 1994] A. Barvinok. A polynomial time algorithm for counting integral points in polyhedra when the dimension is fixed. *Math. Oper. Res.*, 19(4):769–779, 1994.
- [Baumeister and Hoguebe, 2020] D. Baumeister and T. Hoguebe. Complexity of election evaluation and probabilistic robustness: Extended abstract. In *Proceedings of AAMAS-2020*, pages 1771–1773, 2020.
- [Baumeister *et al.*, 2019] D. Baumeister, T. Hoguebe, and L. Rey. Generalized distance bribery. In *Proceedings of AAAI-2019*, pages 1764–1771, 2019.
- [Bredereck *et al.*, 2016a] R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. *Inf. Comput.*, 251:140–164, 2016.
- [Bredereck *et al.*, 2016b] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Complexity of shift bribery in committee elections. In *Proceedings of AAAI-2016*, pages 2452–2458, 2016.
- [Bredereck *et al.*, 2021] R. Bredereck, P. Faliszewski, A. Kaczmarczyk, R. Niedermeier, P. Skowron, and N. Talmon. Robustness among multiwinner voting rules. *Artif. Intell.*, 290:103403, 2021.
- [Brightwell and Winkler, 1991] G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8(3):225–242, 1991.
- [Brill *et al.*, 2020] M. Brill, U. Schmidt-Kraepelin, and W. Suksompong. Refining tournament solutions via margin of victory. In *Proceedings of AAAI-2020*, pages 1862–1869, 2020.
- [Cary, 2011] D. Cary. Estimating the margin of victory for instant-runoff voting. Presented at 2011 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections, 2011.
- [Curticapean and Marx, 2014] R. Curticapean and D. Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *Proceedings of FOCS-2014*, pages 130–139, 2014.
- [Dorn and Schlotter, 2012] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.
- [Elkind and Faliszewski, 2010] E. Elkind and P. Faliszewski. Approximation algorithms for campaign management. In *Proceedings of WINE-2010*, pages 473–482, 2010.
- [Elkind *et al.*, 2009] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of SAGT-2009*, pages 299–310, 2009.
- [Elkind *et al.*, 2020] E. Elkind, P. Faliszewski, S. Gupta, and S. Roy. Algorithms for swap and shift bribery in structured elections. In *Proceedings of AAMAS-2020*, pages 366–374, 2020.
- [Faliszewski *et al.*, 2019] P. Faliszewski, P. Manurangsi, and K. Sornat. Approximation and hardness of shift-bribery. In *Proceedings of AAAI-2019*, pages 1901–1908, 2019.
- [Flum and Grohe, 2004] J. Flum and M. Grohe. The parameterized complexity of counting problems. *SIAM J. Comput.*, 33(4):892–922, 2004.
- [Hazon *et al.*, 2012] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. On the evaluation of election outcomes under uncertainty. *Artif. Intell.*, 189:1–18, 2012.
- [Kaczmarczyk and Faliszewski, 2019] A. Kaczmarczyk and P. Faliszewski. Algorithms for destructive shift bribery. *Auton. Agents Multi-Agent Syst.*, 33(3):275–297, 2019.
- [Kenig and Kimelfeld, 2019] B. Kenig and B. Kimelfeld. Approximate inference of outcomes in probabilistic elections. In *Proceedings of AAAI-2019*, pages 2061–2068, 2019.
- [Knop *et al.*, 2020] D. Knop, M. Koutecky, and M. Mnich. Voting and bribing in single-exponential time. *ACM Trans. Econ. Comput.*, 8(3):12:1–12:28, 2020.
- [Lenstra, Jr., 1983] H. Lenstra, Jr. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.
- [Magrino *et al.*, 2011] T. Magrino, R. Rivest, E. Shen, and D. Wagner. Computing the margin of victory in IRV elections. In *Proceedings of EVT/WOTE-2011*, 2011.
- [Maushagen *et al.*, 2018] C. Maushagen, M. Neveling, J. Rothe, and A.-K. Selker. Complexity of shift bribery in iterative elections. In *Proceedings of AAMAS-2018*, pages 1567–1575, 2018.
- [Shiryayev *et al.*, 2013] D. Shiryayev, L. Yu, and E. Elkind. On elections with robust winners. In *Proceedings of AAMAS-2013*, pages 415–422, 2013.
- [Szufa *et al.*, 2020] S. Szufa, P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. Drawing a map of elections in the space of statistical cultures. In *Proceedings of AAMAS-2020*, pages 1341–1349, 2020.
- [Valiant, 1979] L. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8(2):189–201, 1979.
- [Wojtas and Faliszewski, 2012] K. Wojtas and P. Faliszewski. Possible winners in noisy elections. In *Proceedings of AAAI-2012*, pages 1499–1505, 2012.
- [Xia, 2012] L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of EC-2012*, pages 982–999, 2012.
- [Yang *et al.*, 2019] Y. Yang, Y. Raj Shrestha, and J. Guo. On the complexity of bribery with distance restrictions. *Theor. Comput. Sci.*, 760:55–71, 2019.
- [Zhou and Guo, 2020] A. Zhou and J. Guo. Parameterized complexity of shift bribery in iterative elections. In *Proceedings of AAMAS-2020*, pages 1665–1673, 2020.