On Elections with Robust Winners

Dmitry Shiryaev, Lan Yu, Edith Elkind

Abstract

We study the sensitivity of election outcomes to small changes in voters' preferences. We assume that a voter may err by swapping two adjacent candidates in his vote; we would like to check whether the election outcome would remain the same given up to δ errors. We describe polynomial-time algorithms for this problem for all scoring rules as well as for the Condorcet rule. We are also interested in identifying elections that are maximally robust with respect to a given voting rule. We define the robustness radius of an election with respect to a given voting rule as the maximal number of errors that can be made without changing the election outcome; the robustness of a voting rule is defined as the robustness radius of the election that is maximally robust with respect to this rule. We derive bounds on the robustness of various voting rules, including Plurality, Borda, and Condorcet.

1 Introduction

Voting provides a convenient method for preference aggregation in heterogeneous groups of agents: the group members report how they order the available alternatives (from the most preferred one to the least preferred one), and a voting rule is used to select a winner. There is a wide variety of voting rules that can be used for this purpose, with each of these rules encoding a certain approach to aggregating the preferences of the group members. Clearly, for a voting rule to work as intended, it has to be the case that every voter can reliably submit a ranking that fully reflects his opinion of the available alternatives. However, it is not realistic to assume that this is always the case.

Indeed, there are two main reasons for submitting an erroneous vote. First, the voters may be unable to invest sufficient time and resources in investigating the properties of all the available alternatives, and, as a result, they may err by ordering fairly similar alternatives in a way that deviates from the one they would have chosen if they were to study their options in more detail. Second, voters can make mistakes when filling out their ballots; again, while they are unlikely to rank their top alternative last, they may inadvertently swap adjacent alternatives.

Thus, we may wonder if an outcome of a given election would have remained the same if each vote was a perfect reflection of the respective voter's preferences. Of course, the answer to this question depends on the observed election outcome: if the two most successful candidates are close to being tied, it is quite plausible that the error-free outcome would have been different, but if the current winner leads by a significant margin, the election outcome is likely to reflect the true collective opinion. In other words, given an election, it is natural to ask how *robust* its outcome is, given that our perception of the voters' preferences may be noisy.

In this paper, we study this question for several voting rules, namely, the class of all scoring rules and the Condorcet rule, under the assumption that an "elementary" mistake that a voter (or a vote recording device) can make is to swap two adjacent alternatives in the vote; in recording a given vote, several such mistakes can be made consecutively. This approach is motivated by a classic model of noise used in the study of preferences, which is known as the Mallows noise model [7]. However, in contrast to the Mallows model, we do not assume that mistakes follow a particular distribution. Rather, we are interested in the worst-case scenario, i.e., whether the election result could have been different if we were to deviate by δ swaps of adjacent candidates from the observed preference profile. Thus, we measure the distance between elections using the classic *swap distance* [3] (also known as the inversion distance, the bubble-sort distance, or the Kemeny distance), and we ask whether all elections within a given distance bound δ from the observed election E have the same outcome as E.

We remark that this computational problem can be viewed as the destructive version of the well-studied *swap bribery* problem [4] with unit costs. In more detail, in the (constructive version of) the swap bribery problem it is assumed that an external party wants to make a specific candidate the election winner, and bribes the voters to change their preferences; each voter has a price for swapping every pair of candidates in his vote, and the question is whether the external party can achieve its goal given a certain bribery budget. In the destructive version of this problem (which, to the best of our knowledge, has not been considered in the literature), the briber's goal would be to prevent a specific candidate from winning; clearly, this is equivalent to our question under the assumption that all swaps have the same cost.

We are also interested in understanding the structure of elections whose outcome is maximally robust with respect to a given voting rule, i.e., those whose winner is most resilient to swaps of adjacent candidates. Formally for a given voting rule $\mathcal{F}$, we define the *robustness radius* $\mathrm{rob}_{\mathcal{F}}(E, c)$ of an election E with respect to a candidate c as the smallest number of swaps that have to be applied to E to ensure that c is not the (unique) winner of E under $\mathcal{F}$. The *robustness* of a voting rule $\mathcal{F}$ for a given number of voters n and a given number of candidates m is then defined as the maximal robustness radius, over all n-voter m-candidate elections and all candidates in these elections. This quantity measures the maximum resilience of a voting rule to errors in reported preferences and may vary quite substantially from one voting rule to another: for instance, our results show that the Borda rule is considerably more robust than the Condorcet rule.

Our Results We show that our computational problem admits polynomial-time algorithms for all scoring rules and the Condorcet rule. Further, we obtain essentially matching upper and lower bounds on the robustness of several classes of scoring rules, including such prominent scoring rules as Plurality and Borda. Determining the robustness of the Condorcet rule turns out to be more difficult: while we provide non-trivial upper and lower bounds for this quantity, there is still a gap that remains to be closed. Interestingly, we show that an election that is (almost) maximally robust with respect to many scoring rules is provably non-optimal for the Condorcet rule.

Related Work Procaccia et al. [8] also consider robustness of voting rules to swaps of adjacent candidates. However, their approach differs from ours in several important aspects. First, they measure the robustness of a given election as a *fraction* of swaps that leave the outcome unchanged (they also extend this definition to fixed-length chains of swaps), i.e., while our model of noise is adversarial, theirs is random. Second, Procaccia et al. are interested in *minimally* robust elections, while we focus on elections that are *maximally* robust. Indeed, while the goal of Procaccia et al. is to understand which voting rules are most resilient to errors (or, viewed from a different perspective, least sensitive to changes in voters' preferences), and thus a worst-case approach is appropriate, our aim is to understand which features of a preference profile guarantee that a given voting rule will output the desired result, even in the presence of mistakes. Unsurprisingly, our conclusions are also very different from those of Procaccia et al.: in our framework, Borda turns out to be extremely robust, while Plurality is rather fragile, whereas in the model of Procaccia et al. the opposite is true. Finally, we provide efficient algorithms for computing the robustness radius under many voting rules; in contrast, the results of Procaccia et al. are non-algorithmic in nature.

Our work is also closely related to (and shares some of the motivation) with the recent work by Xia [9] on the margin of victory of voting rules. Indeed, Xia explores essentially the same algorithmic question, but for a different model of errors. Namely, he asks if the election results would have remained the same if up to δ voters were to change their vote arbitrarily. Thus, our papers differ in their notion of an elementary error, or, equivalently, in their approach to measuring distance between elections: while the underlying notion of distance for our work is the swap distance, for [9] it is the Hamming distance. In other words, while we study the destructive version of the swap bribery problem [4], paper [9] studies the destructive version of the original bribery problem [5][1].

[1]To be precise, the margin of victory problem studied in [9] differs from destructive bribery with unit costs in its handling of ties, but the two problems are nevertheless very similar; see the discussion in [9].

While our approach is based on a more fine-grained notion of errors than that of [9], we do not claim that it is generally superior: rather, for either approach there is a range of scenarios where it is more suitable than the other. In particular, the swap distance-based model seems more attractive when voters make mistakes due to imperfect introspection or errors in recording their vote, while the Hamming distance-based approach is more appealing when mistakes are due to (potential) malfunctioning of the vote-recording device (which is the motivation put forward in [9]).

We remark that both in our model and in the model of [9] the associated algorithmic question is easy for all scoring rules, but, apart from this, the contribution of the two papers is incomparable: there are several voting rules studied in [9], but not in our work (though we intend to study these voting rules in the future), but, on the other hand, Xia does not consider the Condorcet rule (he does, however, prove NP-hardness results for several voting rules that are refinements of the Condorcet rule). Also, Xia focuses on the algorithmic aspect of the problem only, while a significant (and perhaps the most mathematically interesting) part of our contribution is the study of the combinatorial question of robustness of voting rules; we believe that this question would be just as interesting to study in the model of [9], and propose it as a direction for future work.

The rest of this paper is organized as follows. After introducing our notation and basic definitions in Section 2, we formally define the problems we intend to study (Section 3). Sections 4 and 5 present our results for scoring rules and the Condorcet rule, respectively. We conclude in Section 6.

2 Preliminaries

An *election* is a pair $E = (C, \mathcal{R})$, where C is a set of *candidates*, or *alternatives*, and $\mathcal{R} = (R_1, \ldots, R_n)$ is a *preference profile*, with each R_i, $i = 1, \ldots, n$, being a linear order over C; we will sometimes write $\succ_i$ in place of R_i. We will refer to the elements of $\mathcal{R}$ as *votes*: R_i is is the vote of the i-th voter in the election $(C, \mathcal{R})$. We denote the number of votes in a preference profile $\mathcal{R}$ by $|\mathcal{R}|$. We say that a voter i prefers $a \in C$ to $b \in C$ if $a \succ_i b$. We denote the candidate ranked by voter i in position j by $c(j, R_i)$. Conversely, we denote the position of a candidate c_j in the i-th vote by $\mathrm{pos}(c_j, R_i)$. We will sometimes identify C with the set $[m] = \{1, \ldots, m\}$. We denote the space of all n-voter m-candidate elections by $\mathcal{E}_{n,m}$.

Given an election $E = (C, \mathcal{R})$, a candidate a is said to *win* the pairwise election against b if more than half of the voters prefer a to b; if exactly half of the voters prefer a to b, then a is said to *tie* his pairwise election against b. A candidate $a \in C$ is said to be the *Condorcet winner* of the election $E = (C, \mathcal{R})$ if he beats every other candidate in their pairwise election.

Given two votes R and R' over a set of candidates C, the *swap distance* between R and R', denoted by $d_{\mathrm{swap}}(R, R')$, is the number of swaps of adjacent candidates needed to transform R into R', or, equivalently, the number of pairs $(a, b) \in C \times C$ such that in R candidate a is ranked above candidate b, but in R' candidate b is ranked above candidate a. Given two n-voter elections $E = (C, \mathcal{R})$ and $E' = (C, \mathcal{R}')$ over the same set of candidates C, the *swap distance* between them, denoted by $d_{\mathrm{swap}}(E, E')$, is given by $d_{\mathrm{swap}}(E, E') = \sum_{i=1,\ldots,n} d_{\mathrm{swap}}(R_i, R'_i)$.

A *voting correspondence* (in what follows, we will use the terms *voting correspondence* and *voting rule* interchangeably) is a mapping $\mathcal{F}$ that given an election $E = (C, \mathcal{R})$ outputs a non-empty set of candidates $W = \mathcal{F}(E) \subseteq C$; the candidates in W are called the *winners* of the election E under the voting rule $\mathcal{F}$. We will now define the voting rules that will be considered in this paper.

Scoring rules. Every vector of non-negative reals $\alpha = (\alpha_1, \ldots, \alpha_m)$ that satisfies $\alpha_1 \geq \cdots \geq \alpha_m$ corresponds to a scoring rule $\mathcal{F}_\alpha$, which is defined for m-candidate elections only. Under this rule, each candidate in an election $E = (C, \mathcal{R})$ with $|C| = m$ receives α_i points from every voter that ranks him in position i; the $\mathcal{F}_\alpha$-*score* of a candidate c in E (denoted by $s_\alpha(E, c)$) is the total number of points that c receives in E. The winners under $\mathcal{F}_\alpha$ are the candidates with the highest $\mathcal{F}_\alpha$-score. The vector $(\alpha_1, \ldots, \alpha_m)$ is called the *scoring vector* that corresponds to the scoring rule

$\mathcal{F}_\alpha$. As we are interested in asymptotic complexity results, we will consider families of scoring rules $\{\mathcal{F}_{\alpha^m}\}_{m\geq 1}$, where $\alpha^m = (\alpha_1^m, \dots, \alpha_m^m)$ and $\alpha_1^m \geq \dots \geq \alpha_m^m$. We require these families to be polynomial-time computable, i.e., we assume that for each $m \geq 1$ and each $i = 1, \dots, m$ the number α_i^m is a non-negative integer given in binary, and, moreover, there is a polynomial-time algorithm that can output α_i^m given m and i. There are several prominent voting rules that correspond to families of scoring rules. In particular, Plurality is the family of scoring rules given by $\alpha_1^m = 1$, $\alpha_i^m = 0$ for all $m \geq 1$ and all $i = 2, \dots, m$, Veto is the family of scoring rules given by $\alpha_m^m = 0$, $\alpha_i^m = 1$ for all $m \geq 1$ and all $i = 1, \dots, m-1$, Borda is the family of scoring rules given by $\alpha_i^m = m - i$ for all $m \geq 1$ and all $i = 1, \dots, m$, and k-approval is the family of scoring rules such that for each $m \geq 1$ it holds that $\alpha_i^m = 1$ for $i = 1, \dots, k$ and $\alpha_i^m = 0$ for all $i = k+1, \dots, m$,

The Condorcet rule. Under the Condorcet rule, if the election has a Condorcet winner, he is the (unique) election winner; otherwise, the set of winners is C. We remark that it is more common (see, e.g., [2]) to say that in the latter case the election has no winners. However, in the social choice literature it is standard to require (as we do) that a voting rule outputs a non-empty winner set for every election, so we have modified the definition of the Condorcet rule to satisfy this requirement. Since in this paper we focus on the unique winner variant of our computational problem (see Section 3 for formal definitions), these two definitions are essentially equivalent. However, for the non-unique variant of our problem this is no longer the case; we discuss this issue in detail in Section 5.

In what follows, we abbreviate the Plurality rule to $\mathcal{F}_\mathcal{P}$, the Borda rule to $\mathcal{F}_\mathcal{B}$, k-approval to $\mathcal{F}_k$, and the Condorcet rule to $\mathcal{F}_\mathcal{C}$.

3 Our Model

We will now present the two questions that will be the focus of this paper.

Definition 3.1. *Given a voting rule $\mathcal{F}$, an instance of $\mathcal{F}$-*UC DESTRUCTIVE SWAP BRIBERY *(here "UC" stands for "unit cost") is given by an election $E = (C, \mathcal{R})$, a candidate $c \in C$, and a parameter $\delta \in \mathbb{Z}^+$. It is a "yes"-instance if $\mathcal{F}(E) = \{c\}$, but there exists an election $E' = (C, \mathcal{R}')$ with $d_{\mathrm{swap}}(E, E') \leq \delta$ such that $\mathcal{F}(E') \neq \{c\}$. Otherwise, it is a "no"-instance.*

We remark that in Definition 3.1 we consider the unique winner version of our problem, i.e., we require c to be the unique winner of the original election, and we seek a modified election for which this is no longer the case. Alternatively, one could consider the non-unique winner version of the problem, where c is required to be one of the election winners, and the goal is to find an election in which c is not an election winner at all. It is not hard to verify that the dynamic programming algorithm for scoring rules presented in Section 4 can be modified to work for the non-unique winner version of our problem. However, for the Condorcet rule the relationship between the two variants of the problem is more complicated (see Section 5). We chose to focus on the unique winner version of our problem since it provides a better match for the intuition behind the Condorcet rule.

Definition 3.2. *Given a voting rule $\mathcal{F}$, an election $E = (C, \mathcal{R})$ and a candidate $c \in C$, the* robustness radius *of E with respect to c under $\mathcal{F}$, denoted by $\mathrm{rob}_\mathcal{F}(E, c)$, is the smallest value of δ such that there exists an election $E' = (C, \mathcal{R}')$ with $d_{\mathrm{swap}}(E, E') \leq \delta$ such that $\mathcal{F}(E') \neq \{c\}$.*

Clearly, $\mathrm{rob}_\mathcal{F}(E, c) \geq 0$ and $\mathrm{rob}_\mathcal{F}(E, c) = 0$ if and only if c is not the unique winner of E under $\mathcal{F}$. Moreover, since the swap distance between any pair of n-voter m-candidate elections is at most $\delta_{m,n} = n\frac{m(m-1)}{2}$, we have $\mathrm{rob}_\mathcal{F}(E, c) \leq \delta_{m,n}$ for every $E \in \mathcal{E}_{n,m}$.

Given a voting rule, we would like to understand the structure of the elections that have the maximum robustness radius with respect to this rule. Thus, overloading notation, we define the *robustness of a voting rule $\mathcal{F}$* as a function

$$\mathrm{rob}_\mathcal{F}(m, n) = \max\{\mathrm{rob}_\mathcal{F}(E, c) \mid E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}, c \in C\}.$$

In what follows, we will investigate the complexity of UC DESTRUCTIVE SWAP BRIBERY and prove upper and lower bounds of $\mathrm{rob}_{\mathcal{F}}(m,n)$ for several families of scoring rules as well as the Condorcet rule.

4 Scoring Rules

We start by describing a simple dynamic programming algorithm that efficiently solves UC DESTRUCTIVE SWAP BRIBERY for any polynomial-time computable family of scoring rules. We then describe a simpler and faster algorithm for the Borda rule.

Theorem 4.1. *The problem $\{\mathcal{F}_{\alpha^m}\}_{m\geq 1}$-UC DESTRUCTIVE SWAP BRIBERY is in P for any polynomial-time computable family of scoring rules $\{\mathcal{F}_{\alpha^m}\}_{m\geq 1}$.*

Proof. Fix a scoring vector $\alpha = (\alpha_1, \dots, \alpha_m)$. We will describe an algorithm that given (a) an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$ that has a unique winner c under $\mathcal{F}_\alpha$ and (b) a positive integer δ, determines whether there exists an election E' with $d_{\mathrm{swap}}(E, E') \leq \delta$ such that $\mathcal{F}_\alpha(E') \neq \{c\}$. The running time of our algorithm will be polynomial in n, m, $\log\delta$ and $\log\alpha_1$. Clearly, this implies the statement of the theorem.

Consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$. Suppose that c is the unique winner of E. For each $a \in C \setminus \{c\}$, we will check whether there exists an election E_a with $d_{\mathrm{swap}}(E, E_a) \leq \delta$ such that in E_a the $\mathcal{F}_\alpha$-score of a is at least as high as that of c; we output "yes" if the answer is positive for at least one $a \in C \setminus \{c\}$. Given an election $E' = (C, \mathcal{R}')$ and a candidate $a \in C \setminus \{c\}$, let $\mathrm{def}(E', a) = \max\{0, s_\alpha(E', c) - s_\alpha(E', a)\}$; we will refer to the quantity $\mathrm{def}(E', a)$ as the *deficit* of a in E'. Thus, our goal is find an election E_a within a distance δ from E such that the deficit of a in E_a is 0.

We start by considering a variant of this problem where we are only allowed to modify a single vote $R_i \in \mathcal{R}$. Suppose that we are allowed to make at most d swaps in R_i. Let $z(i, d)$ be the maximum reduction in a's deficit that can be obtained in this manner. Clearly, we cannot benefit from swaps that do not involve a or c. Thus, we should use our d swaps to move a upwards or to move c downwards (or both), and it remains to decide how many swaps to allocate to each of these actions; this can be determined by considering all possible splits. More precisely, for each $d' = 0, \dots, d$, we consider the vote $R_i(d')$ obtained by first shifting c by d' positions downwards in R_i and then shifting a by $d - d'$ positions upwards in the resulting vote; among these $d+1$ votes, we pick one that reduces a's deficit as much as possible, and let $z(i, d)$ be the corresponding reduction in a's deficit.

We are now ready to describe the dynamic programming algorithm for our problem. For each $d = 0, \dots, \delta$ and each $i = 0, \dots, n$, let $N(i, d)$ be the smallest deficit of a over all elections at swap distance at most d from E that differ from E in the first i votes only. The quantities $N(i, d)$ can be computed as follows. Clearly, for every $d = 0, \dots, \delta$, $N(0, d)$ is simply a's deficit in the original election E, which is straightforward to compute. Further, we have

$$N(i,d) = \max\left\{0, \min_{d'=0,\dots,d} (N(i-1, d-d') - z(i, d'))\right\}$$

for all $d = 0, \dots, \delta$ and all $i = 1, \dots, n$. Indeed, we simply have to find an optimal way of splitting d swaps between the i-th vote and the first $i-1$ votes; the best way to use the d' swaps allocated to the i-th vote is given by $z(i, d')$. Thus, the quantities $N(i, d)$ can be computed inductively starting from $i = 0$. Once we have computed $N(n, \delta)$, it remains to check if $N(n, \delta) = 0$; if yes, we have succeeded in finding an election at distance at most δ from E where a's score is at least as high as that of c. $\square$

For some scoring rules, the algorithm given in the proof of Theorem 4.1 can be simplified. In particular, this is the case for the Borda rule. Indeed, under this rule each upwards swap involving a

but not c, as well as each downwards swap involving c but not a, reduces a's deficit by 1; the most "profitable" swaps are the ones that involve both a and c, as they reduce a's deficit by 2. Thus, our optimal strategy is to maximize the number of "super-profitable" swaps. This observation allows us to simplify our algorithm as follows. We first consider the list $\mathcal{R}' \subseteq \mathcal{R}$ of all votes where c is ranked above a. We re-order the votes in this list according to the number of candidates ranked between c and a, from the smallest to the largest (breaking ties arbitrarily). We then process the votes in $\mathcal{R}'$ one by one. In each vote, we swap c downwards until it is swapped with a. If we have processed all votes in $\mathcal{R}'$, and we still have some swaps available, we allocate them arbitrarily to swapping c downwards or swapping a upwards in any vote in $\mathcal{R}$ where this can be done. Clearly, this approach maximizes the number of swaps that reduce the deficit by 2, and is therefore optimal.

We now move on to the study of robustness of scoring rules. We first provide a simple upper bound that applies to all "reasonable" voting rules. We then show that for the Borda rule this bound is essentially tight.

We say that a voting rule $\mathcal{F}$ is *unanimity-consistent* if in every election E where some candidate c is ranked first by all voters it holds that c is a winner of E under $\mathcal{F}$. Note that all voting rules considered in this paper (and, more broadly, all common voting rules) are unanimity-consistent.

Theorem 4.2. *For any unanimity-consistent voting rule $\mathcal{F}$ we have $\mathrm{rob}_{\mathcal{F}}(m,n) \leq \frac{nm}{2}$.*

Proof. Consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$, and let c be a winner of E under $\mathcal{F}$. For every candidate $a \in C \setminus \{c\}$, let r_a be the number of swaps required to get a into the top position in each vote in $\mathcal{R}$; note that by unanimity consistency performing these r_a swaps would make a an election winner. We have

$$\sum_{a \in C \setminus \{c\}} r_a \leq n(1 + 2 + \ldots + (m-1)) = \frac{nm(m-1)}{2}.$$

As $|C \setminus \{c\}| = m-1$, by the pigeonhole principle there exists some $a \in C \setminus \{c\}$ such that $r_a \leq \frac{nm}{2}$. Hence, $\mathrm{rob}_{\mathcal{F}}(m,n) \leq \frac{nm}{2}$. □

Interestingly, for the Borda rule this bound is essentially tight.

Theorem 4.3. *We have $\mathrm{rob}_{\mathcal{F}_\mathcal{B}}(m,n) = \frac{nm}{2} + O(n+m)$.*

Proof. The upper bound follows immediately from Theorem 4.2. For the lower bound, consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$, where $C = \{c_1, \ldots, c_m\}$ and $\mathcal{R}$ consists of $\lfloor n/2 \rfloor$ votes of the form $c_1 \succ c_2 \succ \ldots \succ c_m$ and $\lceil n/2 \rceil$ votes of the form $c_1 \succ c_m \succ \ldots \succ c_2$. In this election c_1 is the unique Borda winner, and his Borda score is $n(m-1)$. On the other hand, consider a candidate c_i with $i > 1$. His Borda score in E is $(m-i)\lfloor \frac{n}{2} \rfloor + (i-2)\lceil \frac{n}{2} \rceil = \frac{nm}{2} + O(n+m)$.

Now, consider a minimal sequence of swaps that transforms E into an election E' where c_i is a Borda winner. Each swap decreases the difference between the score of c_1 and that of c_i by at most one unless this swap involves both c_1 and c_i (in which case it decreases the difference in their scores by 2); however, there can be at most n swaps of the latter type. Therefore, the total number of swaps required to make c_i an election winner is at least $\frac{nm}{2} + O(n+m)$, and therefore $\mathrm{rob}_{\mathcal{F}_\mathcal{B}}(m,n) \geq \frac{nm}{2} + O(n+m)$. □

Next, we consider the k-approval rule with $k \geq m/2$. We will use the following construction. Given a vote R over a candidate set C of size m, we say that R' is obtained from R by the *downwards shift* if $c(1, R') = c(m, R)$ and for each $j = 2, \ldots, m$ it holds that $c(j, R') = c(j-1, R)$. For instance, by applying the downwards shift to the vote $c_1 \succ \ldots \succ c_{m-1} \succ c_m$ we obtain the vote $c_m \succ c_1 \succ \ldots \succ c_{m-1}$. We say that an election $(C, \mathcal{R}) \in \mathcal{E}_{n,m}$ is an (R, n, m)*-typhoon* if $n = m\alpha$ for some $\alpha \in \mathbb{N}$, $R_1 = R$, for each $i = 2, \ldots, m$ the vote R_i is obtained from the vote R_{i-1} by the downwards shift, and for each $j = 1, \ldots, \alpha - 1$ and each $i = 1, \ldots, m$ it holds that

$R_{mj+i} = R_i$. Further, we say that an election $(C, \mathcal{R}) \in \mathcal{E}_{n,m}$ is a (c, R', n, m)-*lidded typhoon* if $c \in C$, $n = (m-1)\alpha$ for some $\alpha \in \mathbb{N}$, R' is a vote over $C \setminus \{c\}$, and $\mathcal{R}$ is obtained from the $(R', n, m-1)$-typhoon by inserting c into the top position of each vote in $\mathcal{R}'$.

Theorem 4.4. *For $k \geq \frac{m}{2}$ we have* $\mathrm{rob}_{\mathcal{F}_k}(m, n) = \frac{n(m-k)^2}{2m} + O(n+m)$.

Proof. For the upper bound, consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$ that has some candidate c as its unique k-approval winner. Consider a candidate $a \in C \setminus \{c\}$. To ensure that c in not the unique winner of E, it suffices to swap a into the top k positions in each vote. Let r_a denote the number of swaps needed to place a into top k positions in every vote. We have

$$\sum_{a \in C \setminus \{c\}} r_a \leq n(1 + 2 + \ldots + (m-k)) = \frac{n(m-k)(m-k+1)}{2}.$$

As $|C \setminus \{c\}| = m - 1$, by the pigeonhole principle there exists some $a \in C \setminus \{c\}$ such that

$$r_a \leq \frac{n(m-k)(m-k+1)}{2(m-1)} = \frac{n(m-k)^2}{2m} + O(n+m),$$

which establishes our upper bound.

For the lower bound, we provide a proof for the case $n = \alpha(m-1)$ for some $\alpha \in \mathbb{N}$. Our proof can be extended to the case where $m-1$ does not divide n; we omit the details due to space constraints.

Let R' be a vote over the candidate set $\{c_2, \ldots, c_m\}$ given by $c_2 \succ \ldots \succ c_m$, and let $(C, \mathcal{R})$ be the $(c_1, R', n, m-1)$-lidded typhoon. Clearly, c_1 is the unique winner of $(C, \mathcal{R})$ under k-approval. Fix a candidate c_i with $i > 1$, and consider a minimal sequence of swaps that makes c_i a k-approval winner. Clearly, the only useful swaps are the ones that shift c_1 out of top k positions or ones that shift c_i into top k positions. Shifting c_i into top k positions requires at most $m-k$ swaps, while shifting c_1 out of top k positions requires k swaps, and by our choice of k we have $k \leq m-k$. Thus, an optimal sequence of swaps that makes c_i a k-approval winner is to shift him into top k positions in every vote. Since c_i appears in each of the bottom $m-k$ positions exactly α times, the total number of swaps required is

$$\alpha\frac{(m-k)(m-k+1)}{2} = n\frac{(m-k)(m-k+1)}{2(m-1)} = \frac{n(m-k)^2}{2m} + O(n+m).$$

We conclude that $\mathrm{rob}_{\mathcal{F}_k}(m, n) \geq n(m-k)^2/(2m) + O(n+m)$. $\square$

For k-approval with $k \leq m/2$, the argument in the proof of Theorem 4.4 no longer applies. Specifically, while we conjecture that lidded typhoons are maximally robust for small values of k as well, it is no longer the case that to make some non-top-ranked candidate a an election winner it is optimal to only perform swaps that shift a into the top k positions. Indeed, for small values of k it may be easier to move the top-ranked candidate out of the top k positions. We will now show that this is indeed the case for the Plurality rule.

Theorem 4.5. *For $m \geq 6$, we have* $n - 1 - \frac{n}{m-1} \leq \mathrm{rob}_{\mathcal{F}_\mathcal{P}}(m, n) \leq n - \lceil \frac{n}{m-1} \rceil$.

Proof. For the upper bound, consider an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$ and suppose that c_1 is the unique Plurality winner of E. Then c_1's Plurality score is at most n. On the other hand, by the pigeonhole principle there exists a candidate $a \in C \setminus \{c\}$ that is ranked in top two positions at least $\lceil \frac{n}{m-1} \rceil$ times. Thus, by using at most $\lceil \frac{n}{m-1} \rceil$ swaps we can ensure that a's Plurality score is at least $\lceil \frac{n}{m-1} \rceil$. Observe that at this point the Plurality score of c is at most $n - \lceil \frac{n}{m-1} \rceil$, so using additional $n - 2\lceil \frac{n}{m-1} \rceil$ swaps, we can reduce its Plurality score to at most $\lceil \frac{n}{m-1} \rceil$. Thus, $\mathrm{rob}_{\mathcal{F}_\mathcal{P}}(m, n) \leq n - \lceil \frac{n}{m-1} \rceil$.

For the lower bound, suppose first that $n = \alpha(m-1)$ for some $\alpha \in \mathbb{N}$. Let $(C, \mathcal{R})$ be the (c_1, R', n, m)-lidded typhoon, where R' is an arbitrary preference order over $C \setminus \{c_1\}$. Among all minimum-length sequences of swaps which ensure that c_1 is not the unique election winner under Plurality, pick one which swaps c_1 out of the top position in the maximum number of votes, and let c_i, $i > 1$, be a winner of the resulting election E'. Let N_1 be the set of voters in E' that rank c_1 first, let N_i be the set of voters in E' that rank c_i first, and let $N' = N \setminus (N_c \cup N_i)$ be the set of all other voters; we have $|N_i| \geq |N_1|$.

We have $N' \neq \emptyset$, since otherwise we would have $|N_i| \geq n/2$, and for $m \geq 6$ the cost of swapping c_i into the top position in $n/2$ votes exceeds n. Therefore, we have $|N_i| = |N_1|$. Indeed, if $|N_i| > |N_1|$, we could shorten our swap sequence by not making the swaps in some vote in N': in the resulting election it would still be the case that $|N_i| \geq |N_1|$. Now, suppose that $|N_i| > \alpha$. Then we had to perform at least two swaps in at least one vote in N_i. Consider a modified sequence of swaps that performs no swaps in this vote (so that it still ranks c_1 first), but swaps c_1 out of the top position in two votes in N_1. The length of this modified sequence is at most that of the original sequence, it also ensures that c_i's Plurality score is at least as high as that of c_1, and it swaps c_1 out of the top position in a higher number of votes, a contradiction with our choice of the swap sequence. It follows that $|N_i| = |N_1| = \alpha$, which implies that the length of our swap sequence is at least $n - \alpha = n - \frac{n}{m-1}$.

It is easy to generalize this argument to the case where $m-1$ does not divide n to obtain a slightly weaker lower bound of $n - 1 - \frac{n}{m-1}$; we omit the details. $\square$

It is instructive to compare the bounds obtained in Theorems 4.3, 4.4, and 4.5. Perhaps not surprisingly, among all k-approval rules with $k \geq m/2$, the $m/2$-approval rule is the most robust, and Veto is the least robust. It is interesting to note that Borda is about four times more robust that $m/2$-approval and $m/2$ times more robust than Plurality; also Plurality is considerably more robust than Veto.

5 The Condorcet Rule

In this section, we show that UC DESTRUCTIVE SWAP BRIBERY remains easy for the Condorcet rule; however, deriving good bounds on $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(m, n)$ requires quite a bit of effort.

Theorem 5.1. *The problem $\mathcal{F}_\mathcal{C}$-UC DESTRUCTIVE SWAP BRIBERY is in* P.

Proof. Consider an instance of $\mathcal{F}_\mathcal{C}$-UC DESTRUCTIVE SWAP BRIBERY given by an election $E = (C, \mathcal{R})$, a candidate $c \in C$ and a non-negative integer δ. Suppose that c is the Condorcet winner of E. Similarly to the proof of Theorem 4.1, for every candidate $a \in C \setminus \{c\}$ we check if there exists an election E_a with $d_{\mathrm{swap}}(E, E_a) \leq \delta$ such that a beats or ties c in their pairwise election. It is not hard to see that we can use essentially the same algorithm as for the Borda rule: that is, we order the votes where a is ranked below c according to the distance between c and a (from the smallest to the largest) and process these votes one by one, shifting c downwards to appear just below a; we do this until we exhaust our swap budget. We return "yes" if in the end of this process a beats or ties c in their pairwise election. $\square$

We remark that the proof of Theorem 5.1 does not extend to the the co-winner version of the $\mathcal{F}_\mathcal{C}$-UC DESTRUCTIVE SWAP BRIBERY problem. Indeed, suppose that c is a co-winner of an election E. Then the nearest election where c is not a co-winner is one where some other candidate is the (unique) Condorcet winner. Thus, given an election E with no Condorcet winners (where, according to our definition of the Condorcet rule, all candidates are the election winners), solving the co-winner version of $\mathcal{F}_\mathcal{C}$-UC DESTRUCTIVE SWAP BRIBERY is essentially the problem of computing the winners of E under the Dodgson rule (recall that under this rule, the winners are the candidates who can be made the Condorcet winners by the smallest number of swaps of adjacent candidates).

The latter problem is known to be computationally hard [1, 6]. In fact, we can use the results of [1, 6] to show that the co-winner version of $\mathcal{F}_\mathcal{C}$-UC DESTRUCTIVE SWAP BRIBERY is computationally hard as well. We do not present the formal proof of this result, as we do not find the co-winner version of $\mathcal{F}_\mathcal{C}$-UC DESTRUCTIVE SWAP BRIBERY intuitively appealing, and therefore we do not think that this hardness result is informative.

We will now present our upper and lower bounds on the robustness of the Condorcet rule.

It will be convenient to prove bounds on $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(m+1,n)$ rather than $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(m,n)$; our results are not affected by this change, since they involve an error term than is linear in $n+m$. First, we will restate the problem of computing $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(m+1,n)$ as an optimization problem. Given a set $S \subseteq \mathbb{N}$, let $L(S)$ denote the sum of the smallest $\lceil \frac{|S|}{2} \rceil$ numbers in S. Then, given an election $(C,\mathcal{R})$ with $|\mathcal{R}| = n$, the quantity $L(\{\mathrm{pos}(c,R_i) \mid i \in [n]\})$ is the sum of the lowest $\lceil n/2 \rceil$ positions in which candidate c appears in $\mathcal{R}$. We can now reformulate our problem as follows.

Lemma 5.2. *We have* $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(m+1,n) = \max_{(C,\mathcal{R})\in\mathcal{E}_{n,m}} \min_{c\in C} L\left(\{\mathrm{pos}(c,R_i) \mid i \in [n]\}\right)$.

Proof. The proof of Theorem 5.1 shows that for every election $E' = (C',\mathcal{R}') \in \mathcal{E}_{n,m+1}$ and every $c_j \in C'$ we have

$$\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(E',c_j) = \min_{c\neq c_j} L\left(\{\max\{0, \mathrm{pos}(c,R'_i) - \mathrm{pos}(c_j,R'_i)\} \mid i \in [n]\}\right).$$

Indeed, to ensure that c_j is not the unique winner of E' under the Condorcet rule, we need to make c_j tie with or lose to some other candidate $c \neq c_j$, i.e., c has to be ranked higher than c_j in at least $\lceil n/2 \rceil$ votes. For each $c \in C' \setminus \{c_j\}$, the number of swaps needed to make c appear above c_j in vote i is $\max\{0, \mathrm{pos}(c,R'_i) - \mathrm{pos}(c_j,R'_i)\}$, and to minimize the total number of swaps for c, we take the $\lceil n/2 \rceil$ votes for which we need the smallest number of swaps. Finally, we choose a candidate $c \in C' \setminus \{c_j\}$ for which the required number of swaps is the smallest.

Now, consider an election $E' = (C',\mathcal{R}') \in \mathcal{E}_{n,m+1}$ and a candidate $c_j \in C'$. Let $E^j = (C',\mathcal{R}^j)$ be the election obtained by moving c_j to the top of each vote in E' (and not changing the relative order of the remaining candidates). We can simplify the expression for $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(E^j,c_j)$, since we have $\mathrm{pos}(c_j,R^j_i) = 1$ and $\mathrm{pos}(c,R^j_i) > \mathrm{pos}(c_j,R^j_i)$ for all $i \in [n]$ and all $c \in C' \setminus \{c_j\}$. Thus, we obtain

$$\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(E^j,c_j) = \min_{c\neq c_j} L(\{\mathrm{pos}(c,R^j_i) - 1 \mid i \in [n]\}).$$

On the other hand, it is not hard to see that $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(E',c_j) \leq \mathrm{rob}_{\mathcal{F}_\mathcal{C}}(E^j,c_j)$. Thus, when computing $\mathrm{rob}_{\mathcal{F}_\mathcal{C}}(m+1,n)$, we only need to consider elections where some candidate c_j is ranked first in every vote; denote the set of all such elections by $\mathcal{E}^j_{m+1,n}$. Note also that the identity of this candidate does not matter. Now, take an election $E^j = (C',\mathcal{R}^j) \in \mathcal{E}^j_{m+1,n}$, let $C = C' \setminus \{c_j\}$ and consider an election $E = (C,\mathcal{R}) \in \mathcal{E}_{n,m}$ obtained by removing c_j from each vote in E^j. Note that any election over C can be obtained in this way.

For every $c \in C$ we have $L(\{\mathrm{pos}(c,R_i) \mid i \in [n]\}) = L(\{\mathrm{pos}(c,R^j_i) - 1 \mid i \in [n]\})$. Consequently,

$$\mathrm{rob}_{\mathcal{F}}(m+1,n) = \max_{(C,\mathcal{R})\in\mathcal{E}_{n,m}} \min_{c\in C} L(\{\mathrm{pos}(c,R_i) \mid i \in [n]\}). \qquad \square$$

From now on, to simplify notation, we identify the candidate set C with $[m]$ and let $s_j = L(\{\mathrm{pos}(j,R_i) \mid i \in [n]\})$ for each candidate $j \in [m]$. By Lemma 5.2, it suffices to find upper and lower bounds on $\max_{E\in\mathcal{E}_{n,m}} \min_{j\in[m]} s_j$. The next theorem provides a lower bound.

Theorem 5.3. *For every $m,n \in \mathbb{N}$ there exists an election $E = (C,\mathcal{R}) \in \mathcal{E}_{n,m}$ such that $s_j \geq \frac{1}{6}mn + O(m+n)$ for every candidate $j \in [m]$.*

Proof. We start by giving the proof for the case $m = 3k$, $n = 6\ell$ for some $k, \ell \in \mathbb{N}$.

For each $j = 1, \dots, k$, we place the candidates j, $m-2j+1$, and $m-2j+2$ in positions j, $m-2j+1$, and $m-2j+2$ in each vote so that each of them appears 2ℓ times in each position:

$$\begin{array}{rccc} j: & j \quad \dots \quad j & m-2j+1 \dots m-2j+1 & m-2j+2 \dots m-2j+2 \\ m-2j+1: & m-2j+2 \dots m-2j+2 & j \quad \dots \quad j & m-2j+1 \dots m-2j+1 \\ m-2j+2: & \underbrace{m-2j+1 \dots m-2j+1}_{2\ell} & \underbrace{m-2j+2 \dots m-2j+2}_{2\ell} & \underbrace{j \quad \dots \quad j}_{2\ell} \end{array}$$

Clearly, this results in a valid profile over $[m]$. For instance, for $m = n = 6$ we obtain the following profile:

$$\begin{pmatrix} 1 & 1 & 5 & 5 & 6 & 6 \\ 2 & 2 & 3 & 3 & 4 & 4 \\ 4 & 4 & 2 & 2 & 3 & 3 \\ 3 & 3 & 4 & 4 & 2 & 2 \\ 6 & 6 & 1 & 1 & 5 & 5 \\ 5 & 5 & 6 & 6 & 1 & 1 \end{pmatrix}$$

In such an election, for every $j \in \{1, \dots, k\}$ we have

$$s_j = j \times 2\ell + (m - 2j + 1) \times \ell = m\ell + \ell = \frac{1}{6}mn + O(m+n).$$

By symmetry, $s_j = s_{m-2j+1} = s_{m-2j+2}$. Therefore, $s_j = \frac{1}{6}mn + O(m+n)$ for all $j \in [m]$.

We will now consider the general case, i.e., we drop the assumption that m is divisible by 3 and n is divisible by 6. First, we fill in the top $3\lfloor \frac{m}{3} \rfloor$ rows and the first $6\lfloor \frac{n}{6} \rfloor$ columns of the profile with $3\lfloor \frac{m}{3} \rfloor$ candidates as described above. Then we complete each of these $6\lfloor \frac{n}{6} \rfloor$ columns by an arbitrary permutation of the remaining candidates. Each remaining column can be an arbitrary vote over $[m]$. It is not difficult to adapt the proof for the special case $m = 3k$, $n = 6\ell$ to show that the theorem holds for this profile. $\square$

Combining Theorem 5.3 with Lemma 5.2, we obtain $\mathrm{rob}_{\mathcal{F}}(m+1, n) \geq \frac{1}{6}mn + O(m+n)$ and hence

$$\mathrm{rob}_{\mathcal{F}}(m, n) \geq \frac{1}{6}(m-1)n + O(m+n) = \frac{1}{6}mn + O(m+n).$$

Now we consider the upper bound.

Theorem 5.4. *For any $E \in \mathcal{E}_{n,m}$ there exists a candidate j such that $s_j \leq \lambda mn + O(m+n)$ for any constant $\lambda > (\sqrt{3}-1)/4$.*

Proof. Fix $\lambda > (\sqrt{3}-1)/4$ and suppose for the sake of contradiction that $s_j > \lambda mn + O(m+n)$ for each $j \in [m]$. Given an election $E = (C, \mathcal{R}) \in \mathcal{E}_{n,m}$, we construct an $m \times n$ matrix $M(\mathcal{R})$ as follows. The j-th row of $M(\mathcal{R})$ lists all n positions in which candidate j occurs in the n votes, in non-decreasing order. Below is an example of a 3×4 profile $\mathcal{R}$ and its corresponding matrix $M(\mathcal{R})$.

$$\mathcal{R} = \begin{pmatrix} 1 & 2 & 3 & 3 \\ 2 & 3 & 2 & 2 \\ 3 & 1 & 1 & 1 \end{pmatrix} \qquad\qquad M(\mathcal{R}) = \begin{pmatrix} 1 & 3 & 3 & 3 \\ 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 3 \end{pmatrix}$$

By the definition of $M(\mathcal{R})$, each number between 1 and m (which denotes a position in a vote) appears exactly n times in $M(\mathcal{R})$. Moreover, s_j is simply the sum of the leftmost $\ell = \lceil \frac{n}{2} \rceil$ entries of the j-th row in $M(\mathcal{R})$. Let S denote the submatrix formed by the first ℓ columns of $M(\mathcal{R})$, and let Σ denote the sum of all entries of S. We will derive upper and lower bounds on Σ. For $\lambda > (\sqrt{3}-1)/4$ the lower bound will exceed the upper bound, leading to a contradiction.

As we have assumed that $s_j > \lambda mn + O(m+n)$, a lower bound is immediate:

$$\Sigma = \sum_{i=1}^{m} s_j > \lambda m^2 n + O(m^2 + mn).$$

The upper bound requires much more work. Let a be the smallest entry of the ℓ-th column of $M(\mathcal{R})$, and let i_0 be the index of its row. All entries to the left of a do not exceed a, so $s_{i_0} \leq \ell a$. On the other hand, our assumption implies $s_{i_0} > \lambda mn + O(m+n)$, so we get a lower bound on a: $a > 2\lambda m + O(\frac{m+n}{n})$.

Note that each entry of $M(\mathcal{R})$ that is not in S is at least a. Therefore, all entries that are smaller than a have to appear in S, and each number between 1 and $a-1$ has to appear exactly n times. The sum of these numbers is

$$\Sigma_1 = \sum_{i=1}^{a-1} i \cdot n = \frac{1}{2} a^2 n + O(mn).$$

Let $\Sigma_2 = \Sigma - \Sigma_1$; Σ_2 is the sum of all entries of S that are greater than or equal to a. We will now derive an upper bound on Σ_2, which will imply an upper bound on Σ.

Let $N_{\geq k}$ denote the number of entries in S that are greater than or equal to k. We will first obtain a general upper bound on $N_{\geq k}$. Observe that entries with value k appear in at least $\lceil \frac{N_{\geq k}}{\ell} \rceil$ rows, and each entry in these rows that does not appears in S is greater than or equal to k. Hence the total number of entries that are greater than or equal to k is at least $N_{\geq k}$ (in S) plus $(n-\ell)\lceil \frac{N_{\geq k}}{\ell} \rceil$ (not in S). On the other hand, there are exactly $(m-k+1)n$ entries that are greater than or equal to k, so we get

$$N_{\geq k} \leq \frac{(m-k+1)n}{1 + \frac{n-\ell}{\ell}} = (m-k+1)\ell.$$

In total there are $m\ell$ entries in S, which include the $n(a-1)$ entries that are smaller than a. We want an upper bound for the sum of the remaining $m\ell - n(a-1)$ entries. To maximize Σ_2, the best way to fill up the remaining entries is to set $N_{\geq k} = (m-k+1)\ell$ by using entries $k = m, m-1, \ldots$ until we run out of entries. More specifically, we put in ℓ entries of value $m, m-1, \ldots, 2a-1$, respectively, and after that the entries left are negligible, since there are at most $a-1$ of them (as $\ell \leq (n+1)/2$) and the order of their sum is $O(m^2)$). Therefore,

$$\Sigma_2 \leq \sum_{i=2a-1}^{m} i \cdot \ell + O(m^2) = \frac{1}{2}(m+2a-1)(m-2a+2)\ell + O(m^2) = \frac{1}{2}(m^2 - 4a^2)\frac{n}{2} + O(m^2 + mn).$$

Combining Σ_1 and Σ_2, we obtain

$$\Sigma = \Sigma_1 + \Sigma_2 \leq \frac{1}{4}(2a^2 + m^2 - 4a^2)n + O(m^2 + mn) = \frac{1}{4}(m^2 - 2a^2)n + O(m^2 + mn),$$

which, by the lower bound on a, can be upper-bounded as

$$\frac{1}{4}(m^2 - 2 \cdot 4\lambda^2 m^2)n + O(m^2 + mn) = \frac{1}{4}(1 - 8\lambda^2)m^2 n + O(m^2 + mn).$$

The lower bound on Σ exceeds this upper bound when $\lambda m^2 n > \frac{1}{4}(1 - 8\lambda^2)m^2 n$, i.e., $8\lambda^2 + 4\lambda - 1 > 0$, which holds for $\lambda > (\sqrt{3} - 1)/4$.

□

Combining Theorem 5.4 with Lemma 5.2, we obtain $\mathrm{rob}_{\mathcal{F}_C}(m+1, n) \leq \lambda mn + O(m+n)$ and hence $\mathrm{rob}_{\mathcal{F}_C}(m, n) \leq \lambda(m-1)n + O(m+n) = \lambda mn + O(m+n)$ for every $\lambda > (\sqrt{3}-1)/4$. Thus, we have

$$\frac{mn}{6} + O(m+n) \leq \mathrm{rob}_{\mathcal{F}_C}(m, n) \leq (\frac{\sqrt{3}-1}{4} + \varepsilon)mn + O(m+n)$$

for every $\varepsilon > 0$. We have $1/6 \approx 0.167$ and $(\sqrt{3}-1)/4 \approx 0.183$, i.e., there is a small gap between our lower and upper bounds. Closing this gap is a natural direction for future work. We remark that our bounds indicate than the Condorcet rule is considerably less robust than the Borda rule, but more robust than $m/2$-approval. Also, it is interesting to note that the lidded typhoon is *not* the most robust election with respect to the Condorcet rule.

6 Conclusions and Future Work

We have introduced the notions of robustness radius of an election and robustness of a voting rule. We have provided efficient algorithms for computing the robustness radius of a given election with respect to scoring rules and the Condorcet rule, and we have provided bounds on the robustness of several voting rules, including Plurality, Borda, k-approval for $k \geq m/2$ and the Condorcet rule. It would be interesting to see if our algorithmic results for destructive swap bribery can be extended to voting rules not considered in this paper (such as, e.g., Copeland and Maximin) and to the general cost version of this problem. Similarly, a natural research direction would be to analyze the robustness of other voting rules.

We remark that the robustness notions introduced in this paper are defined in terms of the swap distance. However, one can define and study them for other distances over elections, such as the Hamming distance or the footrule distance. In particular, one might be able to use the techniques developed by Xia [9] in order to study robustness of voting rules with respect to the Hamming distance.

Acknowledgments This research was supported by National Research Foundation (Singapore) under grant 2009-08. We thank Arkadii Slinko and Piotr Faliszewski for useful discussions.

References

[1] J. Bartholdi, C. Tovey, and M. Trick. Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2):157–165, 1989.

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

[3] M. M. Deza and E. Deza. *Encyclopedia of Distances*. Springer, 2009.

[4] E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *SAGT'09*, pages 299–310, 2009.

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

[6] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact analysis of Dodgson elections: Lewis Carroll's 1876 voting system is complete for parallel access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

[7] C. L. Mallows. Non-null ranking models. *Biometrica*, 44:114–130, 1957.

[8] A. D. Procaccia, J. S. Rosenschein, and G. A. Kaminka. On the robustness of preference aggregation in noisy environments. In *AAMAS'07*, pages 416–422, 2007.

[9] L. Xia. Computing the margin of victory for various voting rules. In *ACM EC'12*, 2012.

Dmitry Shiryaev, Lan Yu, Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: SHIR0010@ntu.edu.sg, YULA0001@ntu.edu.sg, eelkind@ntu.edu.sg