Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

Optimal Manipulation of Voting Rules*

Svetlana Obraztsova
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
svet0001@ntu.edu.sg

Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
eelkind@ntu.edu.sg

Abstract

Complexity of voting manipulation is a prominent research topic in computational social choice. The voting manipulation literature usually assumes that the manipulator is only concerned with improving the outcome of the election from her perspective. However, in practice, the manipulator may also be reluctant to lie, i.e., she may have a preference for submitting a vote that does not deviate too much from her true ranking of the candidates. In this paper, we study the complexity of finding a manipulative vote that achieves the manipulator's goal yet is as close as possible to her true preference order. We analyze this problem for three natural notions of closeness, namely, swap distance, footrule distance, and maximum displacement distance, and a variety of voting rules, such as scoring rules, Bucklin, Copeland, and Maximin. For all three distances, we obtain polynomial-time algorithms for all scoring rules and Bucklin and hardness results for Copeland and Maximin.

1 Introduction

Mechanisms for aggregating the preferences of heterogeneous agents play an important role in the design of multi-agent systems (Ephrati and Rosenschein 1997). Such mechanisms are typically implemented by *voting rules*, i.e., mappings that, given the rankings of the available alternatives by all agents, output an alternative that best reflects the collective opinion. There are many different voting rules that are used for group decision making; see, e.g., (Brams and Fishburn 2002) for an overview.

A weakness shared by all reasonable voting rules is their susceptibility to *manipulation*: for any voting rule over a set of alternatives C, $|C| \geq 3$, that is not a dictatorship, there are voting situations where some voter would be better off if, instead of submitting her true ranking of the alternatives, she submitted a vote that did not quite match her true preferences. This was observed by Gibbard (1973) and, independently, by Satterthwaite (1975) more than 30 years ago, and a lot of research effort since then has been spent on identifying voting rules that are at least somewhat resistant to manipulation.

In their pioneering paper, Bartholdi, Tovey and Trick (1989) proposed to use computational complexity as a roadblock in the way of manipulative behavior: they observed that, in practice, the manipulator needs an efficient method to find a successful manipulative vote, and a voting rule that does not admit such a method may be viewed as being relatively less vulnerable to manipulation. However, most classic voting rules, with the notable exception of STV, turn out to be susceptible to manipulation in this sense (Bartholdi, Tovey, and Trick 1989; Bartholdi and Orlin 1991).

In this paper, we study a refinement of the question asked by Bartholdi, Tovey and Trick. We observe that, while the manipulator is willing to lie about her preferences, she may nevertheless prefer to submit a vote that deviates as little as possible from her true ranking. Indeed, if voting is public (or if there is a risk of information leakage), and a voter's preferences are at least somewhat known to her friends and colleagues, she may be worried that voting non-truthfully can harm her reputation—yet hope that she will not be caught if her vote is sufficiently similar to her true ranking. Alternatively, a voter who is uncomfortable about manipulating an election for ethical reasons may find a lie more palatable if it does not require her to re-order more than a few candidates. Finally, a manipulator may want to express support for candidates she truly likes, even if these candidates have no chances of winning; while she may lie about her ranking, she would prefer to submit a vote where her most preferred candidates are ranked close to the top.

These scenarios suggest the following research question: does a voting rule admit an efficient algorithm for finding a manipulative vote that achieves the manipulator's goals, yet deviates from her true ranking as little as possible? To make this question precise, we need to decide how to measure the discrepancy between the manipulator's true preferences and her actual vote. Mathematically speaking, votes are permutations of the candidate set, and there are several *distances* on permutations that one can use. In our work, we consider what is arguably the two most prominent distances on votes, namely, the *swap distance* (Kendall 1938) (also known as bubble-sort distance, Kendall distance, etc.) and the *footrule distance* (Spearman 1904) (also known as the Spearman dis-

*An earlier version of this paper, which contains many—though not all—of the proofs omitted from the current version, appeared in AAMAS'12. Compared to the AAMAS'12 version, the current version provides a more up-to-date literature survey.

tance), as well as a natural variation of the footrule distance, which we call the *maximum displacement* distance.

In more detail, the swap distance counts the number of candidate pairs that are ranked differently in two preference orderings. Thus, when the manipulator chooses her vote based on the swap distance, she is trying to minimize the number of swaps needed to transform her true ranking into the manipulative vote. We remark that for swap distance, our problem can be viewed as a special case of the swap bribery problem (Elkind, Faliszewski, and Slinko 2009); however, our question is not addressed by existing complexity results for swap bribery (Elkind, Faliszewski, and Slinko 2009; Elkind and Faliszewski 2010; Dorn and Schlotter 2010; Schlotter, Faliszewski, and Elkind 2011) (see Section 7 for a discussion). The footrule distance and the maximum displacement distance are based on computing, for each candidate, the absolute difference between his positions in the two votes; the footrule distance then computes the sum of these quantities, over all candidates, while the maximum displacement distance returns the largest of them. We believe that each of these distances captures a reasonable approach to defining what it means for two votes to be close to each other; therefore, we are interested in analyzing the complexity of our manipulation problem for all of them.

We study our problem for several classic voting rules, namely, Bucklin, Copeland, Maximin, as well as all scoring rules. For all these rules, the algorithm of Bartholdi et al. (1989) finds a successful manipulation if it exists. However, this algorithm does not necessarily produce a vote that is optimal with respect to any of our distance measures: in particular, it always ranks the manipulator's target candidate first, even if this is not necessary to achieve the manipulator's goal. Thus, we need to devise new algorithms—or prove that finding an optimal manipulation is computationally hard.

For all three distances, we obtain the same classification of these rules with respect to the complexity of finding an optimal manipulation: our problem is easy for Bucklin and all polynomial-time computable families of scoring rules (see Section 2 for definitions), but hard for Copeland and Maximin. For swap distance and footrule distance, we strengthen these hardness results to show that our problem is, in fact, hard to approximate up to a factor of $\Omega(\log m)$, where m is the number of candidates.

Our results provide a fairly complete picture of the complexity of finding an optimal manipulative vote for the three distances and four types of voting rules that we consider. Interestingly, they indicate that scoring rules (and the Bucklin rule, which is closely related to a subfamily of scoring rules known as k-approval) are fundamentally easier to manipulate than Copeland and Maximin; we remark that this observation is also suggested by the recent work of Obraztsova et al. (Obraztsova, Elkind, and Hazon 2011; Obraztsova and Elkind 2011) on the complexity of manipulation under randomized tie-breaking. Thus, we believe that, besides being interesting for its own sake, our work contributes to the broad agenda of understanding the intrinsic complexity—and, therefore, practical applicability—of various voting rules.

2 Preliminaries

An *election* is given by a set of candidates $C = \{c_1, \dots, c_m\}$ and a vector $\mathcal{R} = (R_1, \dots, R_n)$, where each R_i, $i = 1, \dots, n$, is a linear order over C; R_i is called the *preference order* (or, *vote*) of voter i. We will sometimes write $\succ_i$ in place of R_i. If $a \succ_i b$ for some $a, b \in C$, we say that voter i *prefers* a to b. We denote by $r(c_j, R_i)$ the *rank* of candidate c_j in the preference order R_i: $r(c_j, R_i) = |\{c \in C \mid c \succ_i c_j\}|+1$. We denote the space of all linear orders over C by $\mathcal{L}(C)$. We denote by $(\mathcal{R}_{-i}, L)$ the preference profile obtained from $\mathcal{R}$ by replacing R_i with L.

A *voting correspondence* $\mathcal{F}$ is a mapping that, given a candidate set C and a preference profile $\mathcal{R}$ over C outputs a non-empty subset of candidates $S \subseteq C$; we write $S = \mathcal{F}(\mathcal{R})$. The candidates in S are called the *winners* of election $(C, \mathcal{R})$. A *voting correspondence* $\mathcal{F}$ is said to be a *voting rule* if it always produces a unique winner, i.e., $|\mathcal{F}(\mathcal{R})| = 1$ for any profile $\mathcal{R}$.

A voting correspondence can be transformed into a voting rule with the help of a *tie-breaking rule*. A tie-breaking rule for an election $(C, \mathcal{R})$ is a mapping $T = T(\mathcal{R}, S)$ that for any $S \subseteq C$, $S \neq \emptyset$, outputs a candidate $c \in S$. A tie-breaking rule T is *lexicographic* with respect to a preference ordering $\succ$ over C if for any preference profile $\mathcal{R}$ over C and any $S \subseteq C$ it selects the most preferred candidate from S with respect to $\succ$, i.e., we have $T(S) = c$ if and only if $c \succ a$ for all $a \in S \setminus \{c\}$. In the context of single-voter manipulation problems, where there is one voter that considers lying about his vote to obtain a better outcome, of particular interest are *benevolent* and *adversarial* tie-breaking rules: the former breaks ties in the manipulator's favor while the latter breaks ties against the manipulator's wishes (i.e., tie-breaking is lexicographic with respect to, respectively, the manipulator's true preference ordering and its inverse). In the traditional computational social choice terminology benevolent and adversarial tie-breaking correspond to, respectively, non-unique and unique winner settings.

Voting rules We will now describe the voting correspondences considered in this paper. All these correspondences assign scores to candidates; the winners are the candidates with the highest scores. In what follows, we will assume that these correspondences are transformed into voting rules by breaking ties adversarially; however, all of our results can be adapted in a straightforward manner to benevolent or, more generally, lexicographic tie-breaking.

Scoring rules Any vector $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{R}^m$ such that $\alpha_1 \geq \cdots \geq \alpha_m$ defines a *scoring rule* $\mathcal{F}_\alpha$ as follows. Each voter grants α_i points to the candidate she ranks in the i-th position; the score of a candidate is the sum of the scores he receives from all voters. The vector α is called a *scoring vector*; we assume without loss of generality that the coordinates of α are nonnegative integers given in binary. We remark that scoring rules are defined for a fixed number of candidates, and therefore do not quite fit our definition of a voting rule. Thus, one needs to consider *families* of scoring rules (one for every possible number of candidates). From the algorithmic perspective, it is natural to restrict our attention to *polynomial-time computable fami-*

lies of scoring rules, where the scoring vector α^m for an m-candidate election can be computed in time $\mathrm{poly}(m)$. Two well-known examples of such families are *Borda*, given by $\alpha = (m-1, \ldots, 1, 0)$, and *$k$-approval*, given by $\alpha_i = 1$ if $i \leq k$, $\alpha_i = 0$ if $i > k$.

Bucklin Given an n-voter election, the *Bucklin winning round* is the smallest value of r such that the r-approval score of at least one candidate exceeds $n/2$. The *Bucklin score* of a candidate $c \in C$ is his r-approval score, where r is the Bucklin winning round.

Copeland A candidate a is said to win a *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. Under the Copeland$^\alpha$ rule, $\alpha \in \mathbb{Q} \cap [0,1]$, each candidate gets 1 point for each pairwise election he wins and α points for each pairwise election he ties.

Maximin The *Maximin score* of a candidate $c \in C$ is given by the number of votes c gets in his worst pairwise election, i.e., $\min_{d \in C \setminus \{c\}} |\{i \mid c \succ_i d\}|$.

Distances A *distance* on a space X is a mapping $d : X \times X \to \mathbb{R}$ that has the following properties for all $x, y, z \in X$: (1) non-negativity: $d(x,y) \geq 0$; (2) identity of indiscernibles: $d(x,y) = 0$ if and only if $x = y$; (3) symmetry: $d(x,y) = d(y,x)$; (4) triangle inequality: $d(x,y) + d(y,z) \geq d(x,z)$.

In this paper, we will be interested in distances over votes, i.e., mapping of the form $d : \mathcal{L}(C) \times \mathcal{L}(C) \to \mathbb{R}$. In fact, since we are interested in asymptotic complexity results, we will consider *families of distances* $(d^m)_{m \geq 1}$, where d^m is a distance over the space of all linear orderings of the set $\{c_1, \ldots, c_m\}$. Specifically, we will consider three such families (in the following definitions, $C = \{c_1, \ldots, c_m\}$ and R and L are two preference orders in $\mathcal{L}(C)$, also denoted as $\succ_R$ and $\succ_L$):

Swap distance. The *swap distance* $\mathrm{d}_{\mathrm{swap}}(L,R)$ is given by

$$\mathrm{d}_{\mathrm{swap}}(L,R) = |\{(c_i, c_j) \mid c_i \succ_L c_j \text{ and } c_j \succ_R c_i\}|.$$

This distance counts the number of swaps of adjacent candidates needed to transform L into R.

Footrule distance. The *footrule distance* $\mathrm{d}_{\mathrm{fr}}(L,R)$ is given by

$$\mathrm{d}_{\mathrm{fr}}(L,R) = \sum_{i=1}^{m} |r(c_i, L) - r(c_i, R)|.$$

This distance calculates by how much each candidate needs to be shifted to transform L into R, and sums up all shifts.

Maximum displacement distance. The *maximum displacement distance* $\mathrm{d}_{\mathrm{md}}(L,R)$ is given by

$$\mathrm{d}_{\mathrm{md}}(L,R) = \max_{i=1,\ldots,m} |r(c_i, L) - r(c_i, R)|.$$

This distance is similar to the footrule distance; the only difference is that instead of summing up all shifts it only considers the maximum shift.

It is not hard to verify that the swap distance, the footrule distance, and the maximum displacement distance fulfill all distance axioms. It is also known (Diakonis and Graham 1977) that the swap distance and the footrule distance are always within a factor of two from each other: we have $\mathrm{d}_{\mathrm{swap}}(L,R) \leq \mathrm{d}_{\mathrm{fr}}(L,R) \leq 2\mathrm{d}_{\mathrm{swap}}(L,R)$ for any space of candidates C and any $L, R \in \mathcal{L}(C)$.

3 Our Model

We will now formally describe our computational problem.

Definition 3.1. *Let $\mathcal{D} = (d^m)_{m \geq 1}$ be a family of integer-valued distances, where d^m is a distance over $\mathcal{L}(\{c_1, \ldots, c_m\})$. Let $\mathcal{F}$ be a voting rule. An instance of $(\mathcal{D}, \mathcal{F})$-*OPTMANIPULATION *is given by an election $(C, \mathcal{R})$ with $C = \{c_1, \ldots, c_m\}$, $\mathcal{R} = (R_1, \ldots, R_n)$, a voter $i \in \{1, \ldots, n\}$, a candidate $p \in C$, and a positive integer k. It is a "yes"-instance if there exists a vote $L \in \mathcal{L}(C)$ such that $\mathcal{F}(C, (\mathcal{R}_{-i}, L)) = \{p\}$ and $d^m(R_i, L) \leq k$, and a "no"-instance otherwise.*

Remark 3.2. The problem $(\mathcal{D}, \mathcal{F})$-OPTMANIPULATION is in NP as long as all distances in $\mathcal{D}$ and the rule $\mathcal{F}$ are polytime computable: one can guess a vote L and check that $\mathcal{F}(C, (\mathcal{R}_{-i}, L)) = \{p\}$ and $d^m(R_i, L) \leq k$. In particular, it is in NP for all distance families and voting rules considered in this paper.

Remark 3.3. We formulated OPTMANIPULATION as a decision problem. However, it also admits a natural interpretation as an optimization problem: in this case, we are given an election $(C, \mathcal{R})$, a voter i and a candidate p, and the goal is to find the smallest value of k such that there exists a vote $L \in \mathcal{L}(C)$ at distance at most k from R_i that satisfies $\mathcal{F}(C, (\mathcal{R}_{-i}, L)) = \{p\}$ (k is assumed to be $+\infty$ if there is no vote L with $\mathcal{F}(C, (\mathcal{R}_{-i}, L)) = \{p\}$). In this version of the problem, one can relax the optimality condition, and ask for an *approximately optimal* manipulative vote: an algorithm is said to be a *ρ-approximation algorithm* for $(\mathcal{D}, \mathcal{F})$-OPTMANIPULATION, $\rho \geq 1$, if, given an instance of the problem for which the correct answer is $k \in \mathbb{R} \cup \{+\infty\}$, it outputs a value k' that satisfies $k \leq k' \leq \rho k$. We will consider the optimization version of OPTMANIPULATION (and prove hardness of approximation results) for Copeland and Maximin under swap distance (Sections 4) and footrule distance (Section 5).

Remark 3.4. In our definition of OPTMANIPULATION, the manipulator wants to make a specific candidate elected; the identity of this candidate is given as a part of the instance description. An alternative approach would be to ask if the manipulator can obtain what he considers a better outcome by submitting a non-truthful vote, i.e., whether there is a vote $L \in \mathcal{L}(C)$ such that $d^m(R_i, L) \leq k$ and $\mathcal{F}(C, (\mathcal{R}_{-i}, L)) \succ_i \mathcal{F}(C, \mathcal{R})$; we will refer to this problem as OPTMANIPULATION$'$. Clearly, an efficient algorithm for OPTMANIPULATION can be used to solve OPTMANIPULATION$'$, by determining the winner w under truthful voting, and then running the OPTMANIPULATION algorithm for all candidates that the manipulator ranks above w. Hence, OPTMANIPULATION is at least as hard

as OPTMANIPULATION′. In what follows, we will provide polynomial-time algorithms for the "harder" problem OPTMANIPULATION. On the other hand, all our NP-hardness results apply to the "easier" problem OPTMANIPULATION′: in fact, in all our hardness proofs the manipulator's goal will be to make his favorite candidate the election winner. Using OPTMANIPULATION as our base problem allows for a direct comparison between the problem of finding the optimal manipulation and the swap bribery problem (see Section 7).

4 Swap Distance

We start by considering optimal manipulability with respect to what is perhaps the best known distance on votes, namely, the swap distance $\mathrm{d}_{\mathrm{swap}}$.

Scoring Rules and Bucklin

The main result of this section is a simple polynomial-time algorithm that solves OPTMANIPULATION for swap distance and an arbitrary scoring rule; we then show that this algorithm can be adapted to work for the Bucklin rule.

An observation that will be important for our analysis of scoring rules in this and subsequent sections is that once we select the position of the manipulator's preferred candidate p, we know his final score. Thus, once p's position is fixed, it remains to rank other candidates so that their scores remain strictly lower than that of p (recall that we use adversarial tie-breaking). More formally, let $s_\alpha(c)$ be the total number of points a candidate c receives from non-manipulators under a voting rule $\mathcal{F}_\alpha$; we will say that a position j is *safe* for a candidate c_ℓ given that p is ranked in position f if $s_\alpha(c_\ell) + \alpha_j < s_\alpha(p) + \alpha_f$. Clearly, for a manipulation to be successful, all candidates other than p should be ranked in positions that are safe for them.

Fix a scoring rule $\mathcal{F}_\alpha$ with $\alpha = (\alpha_1, \ldots, \alpha_m)$. Our algorithm relies on a subroutine $\mathcal{A}$ that given an election $(C, \mathcal{R})$ with $|C| = m$, a voter i, a candidate p, and a position f in i's vote, finds an optimal manipulation for i among all votes that rank p in position f. More formally, let

$$\mathcal{L}^f(\alpha) = \{L \in \mathcal{L}(C) \mid \mathcal{F}_\alpha(C, (\mathcal{R}_{-i}, L)) = \{p\}, r(p, L) = f\};$$

our subroutine outputs $\perp$ if the set $\mathcal{L}^f(\alpha)$ is empty and a vote $\hat{L}$ such that $\mathrm{d}_{\mathrm{swap}}(\hat{L}, R_i) \leq \mathrm{d}_{\mathrm{swap}}(L, R_i)$ for all $L \in \mathcal{L}^f(\alpha)$ otherwise. Given $\mathcal{A}$, we can easily solve $(\mathrm{d}_{\mathrm{swap}}, \mathcal{F}_\alpha)$-OPTMANIPULATION: we run $\mathcal{A}$ for all values of f between 1 and m and output "yes" if at least one of these calls returns a vote $\hat{L}$ with $\mathrm{d}_{\mathrm{swap}}(\hat{L}, R_i) \leq k$. Thus the running time of our algorithm is m times the running time of $\mathcal{A}$. We will now describe $\mathcal{A}$.

For convenience, let us renumber the candidates in C so that $c_m = p$ and $c_1 \succ_i \ldots \succ_i c_{m-1}$. $\mathcal{A}$ proceeds in $m-1$ rounds. In the ℓ-th round, $\ell = 1, \ldots, m-1$, it determines the final position of candidate c_ℓ; we then say that this candidate is *pinned* to that position, and the position becomes *unavailable*. Initially, all candidates are unpinned and all positions are available.

Initialization: We pin p to position f (thus f becomes unavailable), and then fill the remaining positions with the candidates in $C \setminus \{p\}$, in the order of i's preferences, i.e., placing c_1 in the highest available position and c_{m-1} in the lowest available position. In what follows, we will shift the candidates around in order to make p the winner.

Round ℓ, $\ell = 1, \ldots, m-1$ Suppose that in the beginning of the round candidate c_ℓ is ranked in position j. If j is safe for c_ℓ, we pin c_ℓ to position j (which then becomes unavailable) and proceed to the next round. Otherwise, we find the smallest value of h such that position h is available and safe for c_ℓ; if no such value of h can be found, we terminate and return $\perp$. If a suitable value of h has been identified (note that $h > j$), then c_ℓ gets pinned to position h, and all unpinned candidates in positions $j+1, \ldots, h$ are shifted one available position upwards.

If $\mathcal{A}$ does not abort (i.e., return $\perp$), it terminates at the end of the $(m-1)$-st round and returns the vote obtained at that point. Each round involves $O(m)$ score comparisons and shifts, and each comparison can be performed in time $O(\log(n\alpha_1))$. Therefore, the running time of $\mathcal{A}$ can be bounded as $O(m^2 \log(n\alpha_1))$.

The following theorem states that $\mathcal{A}$ is guaranteed to produce the correct answer.

Theorem 4.1. *For any $\alpha = (\alpha_1, \ldots, \alpha_m) \in {\mathbb{Z}_+}^m$ the procedure $\mathcal{A}$ takes an n-voter m-candidate election $(C, \mathcal{R})$, a voter $i \in \{1, \ldots, n\}$, a candidate $p \in C$, and a position $f \in \{1, \ldots, m\}$ as its input, and outputs $\perp$ if $\mathcal{L}^f(\alpha) = \emptyset$ and a vote $\hat{L}$ that satisfies $\mathrm{d}_{\mathrm{swap}}(\hat{L}, R_i) \leq \mathrm{d}_{\mathrm{swap}}(L, R_i)$ for all $L \in \mathcal{L}^f(\alpha)$ otherwise.*

Combining Theorem 4.1 with the bound on the running time, we obtain the following corollary.

Corollary 4.2. *For every polynomial-time computable family $\hat{\mathcal{F}} = (\mathcal{F}^m_\alpha)_{m=1,\ldots}$ of scoring rules, the problem $(\mathrm{d}_{\mathrm{swap}}, \hat{\mathcal{F}})$-*OPTMANIPULATION *is in* P.

For the Bucklin rule, the algorithm is essentially the same; the only difference is in the definition of a safe position.

Theorem 4.3. *The problem $(\mathrm{d}_{\mathrm{swap}}, \mathrm{Bucklin})$-*OPTMANIPULATION *is in* P.

Maximin and Copeland

For both Maximin and Copeland, finding an optimal manipulation with respect to the swap distance turns out to be computationally hard. In fact, we will prove that the optimization versions of these problems (see Remark 3.3) cannot be approximated up to a factor of $\delta \log |C|$ for some $\delta > 0$ unless P=NP; this implies, in particular, that the decision versions of these problems are NP-hard (and hence, by Remark 3.2, NP-complete).

We provide reductions from the optimization version of the SET COVER problem (Garey and Johnson 1979). Recall that an instance of SET COVER is given by a ground set $G = \{g_1, \ldots, g_t\}$ and a collection $\mathcal{S} = \{S_1, \ldots, S_r\}$ of subsets of G. In the optimization version of the problem, the goal is to find the smallest value of h such that G can be covered by h sets from $\mathcal{S}$; we denote this value of h by $h(G, \mathcal{S})$. More formally, we are interested in the smallest value of h such that $G = \cup_{S' \in \mathcal{S}'} S'$ for some collection of

subsets $\mathcal{S}' \subseteq \mathcal{S}$ with $|\mathcal{S}'| = h$. A *ρ-approximation algorithm* for SET COVER is a procedure that, given an instance $(G, \mathcal{S})$ of set cover, outputs a value h' that satisfies $h(G, \mathcal{S}) \leq h' \leq \rho \cdot h(G, \mathcal{S})$. There exists a $\delta > 0$ such that SET COVER does not admit a polynomial-time $\delta \log t$-approximation algorithm unless P=NP (Raz and Safra 1997).

Theorems 4.4 and 4.5 state our hardness-of-approximation results for Maximin and Copeland.

Theorem 4.4. *There exists a $\delta > 0$ s. t. $(\mathrm{d}_{\mathrm{swap}}, \mathrm{Maximin})$-*OPTMANIPULATION *does not admit a polynomial-time $\delta \log |C|$-approximation algorithm unless* P=NP.

Theorem 4.5. *There exists a $\delta > 0$ such that for any $\alpha \in \mathbb{Q} \cap [0, 1]$, $(\mathrm{d}_{\mathrm{swap}}, \mathrm{Copeland}^{\alpha})$-*OPTMANIPULATION *does not admit a polynomial-time $\delta \log |C|$-approximation algorithm unless* P=NP.

5 Footrule Distance

For the footrule distance our analysis turns out to be much easier than for the swap distance: for scoring rules and Bucklin, we design a simple matching-based algorithm, and for Copeland and Maximin we can use the fact that the swap distance and the footrule distance are always within a factor of 2 from each other, as this allows us to inherit the hardness results of the previous section.

Scoring Rules and Bucklin

The overall structure of our argument is similar to the one in Section 4: for any scoring rule $\mathcal{F}_\alpha$ with $\alpha = (\alpha_1, \ldots, \alpha_m)$ we will design a procedure $\mathcal{A}'$ that, given an election $(C, \mathcal{R})$ with $|C| = m$, a voter i, the preferred candidate p, a target position f for the preferred candidate, and a bound k on the distance, constructs a vote L such that (a) $\mathcal{F}(C, (\mathcal{R}_{-i}, L)) = \{p\}$; (b) $r(p, L) = f$; (c) $\mathrm{d}_{\mathrm{fr}}(L, R_i) \leq k$, or returns $\perp$ if no such vote exists. We then run this procedure for $f = 1, \ldots, m$ and return "yes" if at least one of these calls *does not* return $\perp$.

We assume without loss of generality that the manipulator ranks the candidates as $c_1 \succ_i \ldots \succ_i c_m$ (note that this is different from the assumption we made in Section 4), and denote by $s_\alpha(c)$ the score of a candidate $c \in C$ in election $(C, \mathcal{R}_{-i})$ under the voting rule $\mathcal{F}_\alpha$. Let r be the rank of p in i's truthful vote, i.e., $p = c_r$.

$\mathcal{A}'$ proceeds by constructing a bipartite graph G with parts $X = C \setminus \{p\}$ and $Y = \{1, \ldots, m\} \setminus \{f\}$; there is an edge from c_j to ℓ if and only if position ℓ is safe for c_j, i.e., $s_\alpha(c_j) + \alpha_\ell < s_\alpha(p) + \alpha_f$, Each edge has a weight: the weight of the edge (c_j, ℓ) is simply $|j - \ell|$. Clearly, there is a one-to-one correspondence between votes L that rank p in position f and satisfy $\mathcal{F}_\alpha(C, (\mathcal{R}_i, L)) = \{p\}$ and perfect matchings in this graph. Furthermore, the cost of a matching M is x if and only if the corresponding vote L_M satisfies $\mathrm{d}_{\mathrm{fr}}(L_M, R_i) = x + |r - f|$. Thus, it suffices to find a minimum cost perfect matching in G; our algorithm returns the vote L that corresponds to this matching if its cost does not exceed $k - |r - f|$ and $\perp$ otherwise. The graph G can be constructed in time $O(m^2 \log(n\alpha_1))$, and a minimum-cost matching can be found in time $O(m^3)$ (Cormen et al. 2001).

We summarize these observations as follows.

Theorem 5.1. *For every polynomial-time computable family $\hat{\mathcal{F}} = (\mathcal{F}_\alpha^m)_{m=1,\ldots}$ of scoring rules, the problem $(\mathrm{d}_{\mathrm{fr}}, \hat{\mathcal{F}})$-*OPTMANIPULATION *is in* P.

For the Bucklin rule, it suffices to combine the matching-based algorithm given above with the definition of a safe position for the Bucklin rule. We obtain the following corollary.

Corollary 5.2. *The problem $(\mathrm{d}_{\mathrm{fr}}, \mathrm{Bucklin})$-*OPTMANIPULATION *is in* P.

Maximin and Copeland

In Section 2 we have mentioned that for any candidate set C and any pair of votes $L, R \in \mathcal{L}(C)$ we have $\mathrm{d}_{\mathrm{swap}}(L, R) \leq \mathrm{d}_{\mathrm{fr}}(L, R) \leq 2\mathrm{d}_{\mathrm{swap}}(L, R)$ (Diakonis and Graham 1977).

Now, suppose that there exists a ρ-approximation algorithm $\mathcal{A}_{\mathrm{fr}}$ for $(\mathrm{d}_{\mathrm{fr}}, \mathcal{F})$-OPTMANIPULATION for some voting rule $\mathcal{F}$. Consider an instance $(C, \mathcal{R}, i, p)$ of (the optimization version of) this problem, and let

$$\mathcal{L}' = \{L \in \mathcal{L}(C) \mid \mathcal{F}(C, (\mathcal{R}_{-i}, L)) = \{p\}\}.$$

If $\mathcal{L}' \neq \emptyset$, let $k = \min\{\mathrm{d}_{\mathrm{fr}}(L, R_i) \mid L \in \mathcal{L}'\}$. On this instance $\mathcal{A}_{\mathrm{fr}}$ outputs a value k' that satisfies $k \leq k' \leq \rho k$; this value corresponds to a vote $L \in \mathcal{L}'$ that satisfies $\mathrm{d}_{\mathrm{fr}}(L, R_i) = k'$.

Now, for any vote $L' \in \mathcal{L}'$ we have

$$\mathrm{d}_{\mathrm{swap}}(L', R_i) \geq \frac{1}{2}\mathrm{d}_{\mathrm{fr}}(L', R_i) \geq \frac{k}{2}.$$

On the other hand, for L we obtain

$$\mathrm{d}_{\mathrm{swap}}(L, R_i) \leq \mathrm{d}_{\mathrm{fr}}(L, R_i) = k' \leq \rho k.$$

Consider an algorithm $\mathcal{A}_{\mathrm{swap}}$ for $(\mathrm{d}_{\mathrm{swap}}, \mathcal{F})$-OPTMANIPULATION that, given an instance of the problem, runs $\mathcal{A}_{\mathrm{fr}}$ on it and returns the value reported by $\mathcal{A}_{\mathrm{fr}}$. The computation above proves that $\mathcal{A}_{\mathrm{swap}}$ is a 2ρ-approximation algorithm for $(\mathrm{d}_{\mathrm{swap}}, \mathcal{F})$-OPTMANIPULATION (note that $\mathcal{A}_{\mathrm{swap}}$ returns $+\infty$ if and only if $\mathcal{L}' = \emptyset$). Combining this observation with Theorems 4.4 and 4.5, we obtain the following corollaries.

Corollary 5.3. *There exists a $\delta > 0$ such that the problem $(\mathrm{d}_{\mathrm{fr}}, \mathrm{Maximin})$-*OPTMANIPULATION *does not admit a poly-time $\delta \log |C|$-approximation algorithm unless* P=NP.

Corollary 5.4. *There exists a $\delta > 0$ such that for any $\alpha \in \mathbb{Q} \cap [0, 1]$, the problem $(\mathrm{d}_{\mathrm{fr}}, \mathrm{Copeland}^{\alpha})$-*OPTMANIPULATION *does not admit a poly-time $\delta \log |C|$-approximation algorithm unless* P=NP.

6 Max Displacement Distance

Maximum displacement distance is fairly generous to the manipulator. Indeed, the optimal manipulation problems for swap distance and footrule distance become trivial if the maximum distance k is bounded by a constant: in this case, there are only polynomially many possible manipulative votes, and the manipulator can try all of them. In contrast, for the maximum displacement distance, there are exponentially many votes even at distance 2 from the true vote (to see this, cut the manipulator's vote into segments

of length 3; within each segment, the candidates can be shuffled independently). Nevertheless, from the algorithmic perspective maximum displacement distance exhibits essentially the same behavior as swap distance and footrule distance: we can design efficient algorithms for all scoring rules and the Bucklin rule, and derive NP-hardness results for Copeland and Maximin.

Scoring Rules and Bucklin

For scoring rules, we can use a simplified variant of the min-cost matching argument given in Section 5. Again, suppose that we are given a scoring rule $\mathcal{F}_\alpha$ with $\alpha = (\alpha_1, \ldots, \alpha_m)$, an election $(C, \mathcal{R})$ with $|C| = m$, a manipulator i, a preferred candidate p and a distance bound k. We assume that the manipulator ranks the candidates as $c_1 \succ_i \ldots \succ_i c_m$. For each $f = 1, \ldots, m$ we try to find a successful manipulative vote L with $\mathrm{d}_{\mathrm{md}}(L, R_i) \leq k$ that ranks p in position f; in fact, it suffices to consider only values of f that satisfy $|f - r(p, R_i)| \leq k$. For each such f, we construct a bipartite graph G with parts $C \setminus \{p\}$ and $\{1, \ldots, m\} \setminus \{f\}$. In this graph, there is an edge from c_j to ℓ if and only if ℓ is safe for c_j (we use the same definition of a safe position as in Section 5) and $|\ell - j| \leq k$. In contrast to the construction in Section 5, the graph is unweighted. It is immediate that there is a one-to-one correspondence between perfect matchings in G and successful manipulative votes at distance at most k from R_i. Thus, we obtain the following result.

Theorem 6.1. *For every polynomial-time computable family $\hat{\mathcal{F}} = (\mathcal{F}^m_\alpha)_{m=1,\ldots,}$ of scoring rules, the problem $(\mathrm{d}_{\mathrm{md}}, \hat{\mathcal{F}})$-*OPTMANIPULATION *is in* P.

For the Bucklin rule, we use the same approach as in Section 5, i.e., combine the matching-based algorithm with the definition of a safe position for the Bucklin rule. This results in following corollary.

Corollary 6.2. *The problem $(\mathrm{d}_{\mathrm{md}}, \mathrm{Bucklin})$-*OPTMANIPULATION *is in* P.

Maximin and Copeland

For Maximin and Copeland, finding an optimal manipulation with respect to the maximum displacement distance is computationally hard; however, in contrast with our results in Sections 4 and 5, we are only able to show the NP-hardness of the decision version of this problem (rather than inapproximability of its optimization version). The proofs of the following two theorems are based on (somewhat involved) reductions from SET COVER.

Theorem 6.3. $(\mathrm{d}_{\mathrm{md}}, \mathrm{Maximin})$-OPTMANIPULATION *is* NP*-complete.*

Theorem 6.4. *For any $\alpha \in \mathbb{Q} \cap [0, 1]$,* $(\mathrm{d}_{\mathrm{md}}, \mathrm{Copeland}^\alpha)$-OPTMANIPULATION *is* NP*-complete.*

7 Related Work

Our work can be placed in the broader context of *mechanism design with verification* (Green and Laffont 1986; Singh and Wittman 2001; Nisan and Ronen 2001). This research area deals with the design of mechanisms (voting rules, auctions, etc.) for selfish agents in settings where agents cannot misrepresent their private information (type) arbitrarily, but rather are restricted to submit a report that is in some way related to their true type. In some settings (most notably, mechanism design for scheduling problems) imposing natural restrictions on possible misreports enables one to circumvent known impossibility results (Auletta et al. 2006; 2009). We remark, however, that the Gibbard–Satterthwaite theorem has been recently shown to be very robust with respect to restrictions on misreporting: every non-dictatorial voting rule for 3 or more candidates remains manipulable even if we only allow the manipulative votes that only differ by a single swap of adjacent candidates from the manipulator's true preference ranking (Caragiannis et al. 2012). Viewed from the perspective of mechanism design with partial verification, the hardness results in this paper provide a complexity-theoretic separation between the unrestricted manipulation problem for Copeland and Maximin and its version with partial verification (where the permissible misreports are required to be within a certain distance from the manipulator's true ranking). To the best of our knowledge, this is a first result of this type in the mechanism design with partial verification literature.

Optimal Manipulability and Swap Bribery

The problem of finding an optimal manipulation with respect to the swap distance can be viewed as a special case of the swap bribery problem (Elkind, Faliszewski, and Slinko 2009). In the swap bribery model, there is an external party that wants to make a particular candidate the election winner. This party can pay the voters to change their preference orders, with a price assigned to swapping each pair of candidates in each vote. The goal is to decide whether the manipulator can achieve his goal given a budget constraint. Clearly, our problem is a special case of swap bribery, where for one voter each swap has unit cost, and for the remaining voters the prices are set to $+\infty$. Swap bribery is known to be hard, even to approximate, for almost all prominent voting rules, including such relatively simple rules as 2-approval. Thus, the easiness results of Section 4 identify a new family of easy instances of the swap bribery problem, thus complementing the results of (Elkind and Faliszewski 2010; Dorn and Schlotter 2010; Schlotter, Faliszewski, and Elkind 2011). It would be interesting to see if a somewhat more general variant of the swap bribery problem for scoring rules, where only one voter can be bribed but swap bribery prices can be arbitrary, remains tractable; it is not clear if the algorithm given in Section 4 can be adapted to handle this setting.

On the other hand, one may wonder if the hardness results of Section 4 are implied by the existing hardness results for swap bribery. However, this does not seem to be the case: the hardness (and inapproximability) of swap bribery for Copeland and Maximin follows from the hardness results for the possible winner problem (Xia and Conitzer 2011), and the latter problem is easy if all but one voter's preferences are fixed (it can be verified that the algorithm of Bartholdi et al. (1989) works even if the positions of some candidates in the vote are already fixed). Thus, the hardness results for

Copeland and Maximin given in Section 4 strengthen the existing hardness results for swap bribery with respect to these rules.

8 Conclusions and Future Work

We have considered the problem of finding a successful manipulative vote that differs from the manipulators' preferences as little as possible, for three distance measures on votes and four types of voting rules. Our results are summarized in Table 1 (where "NPC" stands for "NP-complete" and "$(\log m)$-inapp." stands for "inapproximable up to a factor of $\Omega(\log m)$").

A natural direction for future work is extending our results to other distances on votes; for instance, it should not be too hard to generalize our results for weighted variants of swap and footrule distances; such distances play an important role in several applications of rank aggregation, and have received considerable attention in the literature (see (Kumar and Vassilvitskii 2010) and references therein). At a more technical level, we remark that for maximum displacement distance we only have NP-hardness results for Copeland and Maximin; it would be interesting to see if this variant of our problem admits efficient approximation algorithms.

	Sc. rules	Bucklin	Copeland	Maximin
$\mathrm{d}_{\mathrm{swap}}$	P	P	$(\log m)$-inapp.	$(\log m)$-inapp.
d_{fr}	P	P	$(\log m)$-inapp.	$(\log m)$-inapp.
d_{md}	P	P	NPC	NPC

Table 1: Summary of results

Acknowledgments This research was supported by National Research Foundation (Singapore) under grant 2009-08 (Edith Elkind) and by Russian Foundation for Basic Research grant 11-01-12135 ofi-m (Svetlana Obraztsova).

References

Auletta, V.; Prisco, R. D.; Penna, P.; Persiano, G.; and Ventre, C. 2006. New constructions of mechanisms with verification. In *ICALP'06*, 596–607.

Auletta, V.; Prisco, R. D.; Penna, P.; and Persiano, G. 2009. The power of verification for one-parameter agents. *Journal of Computer and System Sciences* 75(3):190–211.

Bartholdi, J., and Orlin, J. 1991. Single transferable vote resists strategic voting. *Social Choice and Welfare* 8(4):341–354.

Bartholdi, J.; Tovey, C.; and Trick, M. 1989. The computational difficulty of manipulating an election. *Social Choice and Welfare* 6(3):227–241.

Brams, S., and Fishburn, P. 2002. Voting procedures. In Arrow, K.; Sen, A.; and Suzumura, K., eds., *Handbook of Social Choice and Welfare, Volume 1*. Elsevier. 173–236.

Caragiannis, I.; Elkind, E.; Szegedy, M.; and Yu, L. 2012. Mechanism design: from partial to probabilistic verification. In *ACM EC'12*, to appear.

Cormen, T. H.; Leiserson, C. E.; Rivest, R. L.; and Stein, C. 2001. *Introduction to algorithms*. MIT Press.

Diakonis, P., and Graham, R. 1977. Spearman footrule as a measure of disarray. *Journal of the Royal Statistical Society B (Methodological)* 39(2):262–268.

Dorn, B., and Schlotter, I. 2010. Multivariate complexity analysis of swap bribery. In *IPEC'10*, 107–122.

Elkind, E., and Faliszewski, P. 2010. Approximation algorithms for campaign management. In *WINE'10*, 473–482.

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

Ephrati, E., and Rosenschein, J. 1997. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence* 20(1–4):13–67.

Garey, M., and Johnson, D. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

Gibbard, A. 1973. Manipulation of voting schemes. *Econometrica* 41(4):587–601.

Green, J., and Laffont, J.-J. 1986. Partially verifiable information and mechanism design. *Review of Economic Studies* 53:447–456.

Kendall, M. G. 1938. A new measure of rank correlation. *Biometrika* 30(1-2):81–93.

Kumar, R., and Vassilvitskii, S. 2010. Generalized distances between rankings. In *WWW'10*, 571–580.

Nisan, N., and Ronen, A. 2001. Algorithmic mechanism design. *Games and Economic Behavior* 35:166–196.

Obraztsova, S., and Elkind, E. 2011. On the complexity of voting manipulation under randomized tie-breaking. In *IJCAI'11*, 319–324.

Obraztsova, S.; Elkind, E.; and Hazon, N. 2011. Ties matter: Complexity of voting manipulation revisited. In *AAMAS'11*, 71–79.

Raz, R., and Safra, S. 1997. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *STOC'97*, 475–484.

Satterthwaite, M. 1975. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory* 10(2):187–217.

Schlotter, I.; Faliszewski, P.; and Elkind, E. 2011. Campaign management under approval-driven voting rules. In *AAAI'11*, 726–731.

Singh, N., and Wittman, D. 2001. Implementation with partial verification. *Review of Economic Design* 6:63–84.

Spearman, C. 1904. The proof and measurement of association between two things. *The American Journal of Psychology* 15(1):72–101.

Xia, L., and Conitzer, V. 2011. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research* 41:25–67.