On the Complexity of Voting Manipulation under Randomized Tie-Breaking

Svetlana Obraztsova Yair Zick Edith Elkind

Abstract

Computational complexity of voting manipulation is one of the most actively studied topics in the area of computational social choice, starting with the groundbreaking work of Bartholdi et al. [2]. Most of the existing work in this area, including that of Bartholdi et al., implicitly assumes that whenever several candidates receive the top score with respect to the given voting rule, the resulting tie is broken according to a lexicographic ordering over the candidates. In this paper, we explore an equally appealing method of tie-breaking, namely, selecting the winner uniformly at random among all tied candidates. We show that under this method of breaking ties, all scoring rules, the Bucklin rule and Plurality with Runoff remain easy to manipulate; however, finding a manipulative vote becomes NP-hard for Copeland and Maximin. We extend some of our easiness results to elections with multiple winners. We show that if the number of winners is small, then manipulation is in P for all scoring rules, and it is in P for k-Approval for any number of winners.

1 Introduction

Whenever a group of agents have to make a joint decision, the agents' opinions need to be aggregated in order to identify a suitable course of action. This applies both to human societies and to groups of autonomous agents; the entities that the agents need to select from vary from political leaders to song contest winners and joint plans. The standard way to aggregate preferences is by asking the agents to vote over the available candidates: each agent ranks the candidates, and a *voting rule*, i.e., a mapping from collective rankings to candidates, is used to select the winner.

In most preference aggregation settings, each agent wants his most favorite alternative to win, irrespective of other agents' preferences. Thus, he may try to *manipulate* the voting rule, i.e., to misrepresent his preferences in order to obtain an outcome that he ranks higher than the outcome of the truthful voting. Indeed, the famous Gibbard–Satterthwaite theorem [9, 15] shows that whenever the agents have to choose from 3 or more alternatives, every reasonable voting rule is manipulable, i.e., for some collection of voter's preferences some voter can benefit from lying about his ranking. This is bad news, as the manipulator may exercise undue influence over the election outcome, and a lot of research effort has been invested in identifying voting rules that are more resistant to manipulation than others, as measured by the fraction of manipulable profiles or the algorithmic complexity of manipulation (see [7] for an overview).

Many common voting rules operate by assigning scores to candidates, so that the winner is the candidate with the highest score. Now, in elections with a large number of voters and a small number of candidates there is usually only one candidate that obtains the top score. However, this does not necessarily hold when the alternative space is large, as may be the case when, e.g., agents in a multiagent system use voting to decide on a joint plan of action [6]. If, nevertheless, a single outcome needs to be selected, such ties have to be broken. In the context of manipulation this means that the manipulator should take the tie-breaking rule into account when choosing his actions. Much of the existing literature on voting manipulation circumvents the issue by assuming that the manipulator's goal is to make some distinguished candidate p one of the election winners, or, alternatively, the unique winner. The former assumption can be interpreted as a tie-breaking rule that is favorable to the manipulator, i.e., given a tie that involves p, always selects p as the winner; similarly, the latter assumption corresponds to a tie-breaking rule that is adversarial to the manipulator. In fact, most of

the existing algorithms for finding a manipulative vote work for any tie-breaking rule that selects the winner according to a given ordering on the candidates; the two cases considered above correspond to this order being, respectively, the manipulator's preference order or its inverse.

However, till recently, an equally appealing approach to tie-breaking, namely, selecting the winner among all tied candidates uniformly at random, has been rarely studied. Two exceptions to this pattern that we are aware of are [11] and [5]; however, [11] does not deal with manipulation at all, while [5] considers a very different model of manipulation. Perhaps one of the reasons for this is that under randomized tie-breaking the outcome of the election is a random variable, so it is not immediately clear how to compare two outcomes: is having your second-best alternative as the only winner preferable to the lottery in which your top and bottom alternatives have equal chances of winning?

We deal with this issue by augmenting the manipulator's preference model: we assume that the manipulator assigns a numeric utility to all candidates, and his goal is to vote so as to maximize his expected utility, where the expectation is computed over the random choices of the tie-breaking procedure; this approach is standard in the social choice literature (see, e.g., [10]) and has also been used in [5]. We show that in this model all scoring rules are easy to manipulate; this is also the case for Bucklin (both for its classic and simplified versions). On the other hand, we prove that manipulation under randomized tie-breaking is hard for Maximin and Copeland. We complement these hardness results by identifying a natural assumption on the manipulator's utility function that makes Maximin easy to manipulate. We also analyze the complexity of manipulation for three voting rules that compute the winners using a multi-step procedure, namely, Plurality with Runoff, STV, and Ranked Pairs. Thus, we provide an essentially complete picture of the complexity of manipulating common voting rules under randomized tie-breaking (see Table 1 in the end of the paper). Finally, we explore the complexity of manipulation when voters need to choose several winners. We show that for the k-Approval voting rule, multi-winner manipulation is in P; moreover, if the number of winners to be selected is small (i.e., bounded by a constant), then manipulating an election under any scoring rule is also in P.

Some of the results that appear in this paper were previously published in [14] and [13]; however, the material in Section 5 is new.

2 Preliminaries

An *election* is given by a set of candidates $C = \{c_1, \ldots, c_m\}$ and a vector $\mathcal{R} = (R_1, \ldots, R_n)$, where each R_i, $i = 1, \ldots, n$, is a linear order over C; R_i is called the *preference order* (or, *vote*) of voter i. We denote the space of all linear orderings over C by $\mathcal{L}(C)$. The vector $\mathcal{R} = (R_1, \ldots, R_n)$ is called a *preference profile*. For readability, we will sometimes denote R_i by $\succ_i$. When $a \succ_i b$ 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: $r(c_j, R_i) = |\{c \in C \mid c \succ_i c_j\}| + 1$.

A *voting rule* $\mathcal{F}$ is a mapping that, given a preference profile $\mathcal{R}$ over C, outputs a candidate $c \in C$; we write $c = \mathcal{F}(\mathcal{R})$. Many classic voting rules, such as the ones defined below, are, in fact, *voting correspondences*, i.e., they map a preference profile $\mathcal{R}$ to a non-empty subset S of C. Voting correspondences can be transformed into voting rules using *tie-breaking rules*.

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 $c \in S$. We say that 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 $a \in S \setminus \{c\}$.

A *composition* of a voting correspondence $\mathcal{F}$ and a tie-breaking rule T is a voting rule $T \circ \mathcal{F}$ that, given a preference profile $\mathcal{R}$ over C, outputs $T(\mathcal{R}, \mathcal{F}(\mathcal{R}))$. Clearly, $T \circ \mathcal{F}$ is a voting rule and $T \circ \mathcal{F}(\mathcal{R}) \in \mathcal{F}(\mathcal{R})$.

Voting Rules We now describe the voting rules (correspondences) considered in this paper. For all rules that assign scores to the candidates the winners are the candidates with the highest scores.

Scoring rules: Any vector $\alpha = (\alpha_1, \ldots, \alpha_m) \in \mathbb{R}^m$ with $\alpha_1 \geq \ldots \geq \alpha_m$ defines a *scoring rule* $\mathcal{F}_\alpha$. Under this rule, each voter grants α_i points to the candidate it ranks in the i-th position; the score of a candidate is the sum of the scores it receives from all voters. The vector α is called a *scoring vector*. A well-known example of a family of scoring rules is *Borda*, given by $\alpha^m = (m-1, \ldots, 1, 0)$; another example is k-Approval, where a candidate gets one point for each voter that ranks him in the top k positions. 1-Approval is also known as *Plurality*.

Bucklin: Let k^* be the smallest k such that the k-approval score of some $c \in C$ is at least $\lfloor n/2 \rfloor + 1$; we say that k^* is the *Bucklin winning round*. Given a candidate $c \in C$, his *Bucklin score* is his k^*-approval score. Under the *simplified Bucklin* rule, candidates whose Bucklin score is at least $\lfloor n/2 \rfloor + 1$ are the winners, while under *Bucklin*, winners are those with the highest Bucklin score.

Copeland: We say that a candidate a wins 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. Given a rational value $\alpha \in [0, 1]$, under the Copeland$^\alpha$ rule each candidate gets 1 point for each pairwise election he wins and α points for each pairwise election he ties.

Maximin: For every pair of candidates $c, d \in C$, we set $s(c, d) = |\{i \mid c \succ_i d\}|$. The Maximin score of a candidate $c \in C$ is given by $\min_{d \in C \setminus \{c\}} s(c, d)$; that is, c's Maximin score is the number of votes he gets in his worst pairwise election.

Plurality with Runoff and STV: Under the STV rule, the election proceeds in rounds; in each round, the candidate with the lowest Plurality score is eliminated, and candidates' scores are recomputed. The winner is the candidate that survives till the last round. Plurality with Runoff can be thought of as a compressed version of STV: we first select two candidates with the highest Plurality scores, and then output the winner of the pairwise election between them. Note that these definitions are somewhat ambiguous, as several candidates may have the lowest/highest Plurality score; we will comment on this issue in Section 4.

3 The Model

Given a preference profile $\mathcal{R}$ over a candidate set C and a preference order L over C, let $(\mathcal{R}_{-i}, L)$ be the preference profile obtained from $\mathcal{R}$ by replacing R_i with L. We say that a voter $i \in \{1, \ldots, n\}$ can successfully *manipulate* an election $(C, \mathcal{R})$ with respect to a voting rule $\mathcal{F}$ if $\mathcal{F}(\mathcal{R}_{-i}, L) \succ_i \mathcal{F}(\mathcal{R})$. We will now explain how to extend this definition to voting correspondences under the assumption that ties are broken uniformly at random.

Given a voting correspondence $\mathcal{F}$ and an election $(C, \mathcal{R})$, suppose that $\mathcal{F}(C, \mathcal{R}) = S$, where $|S| > 1$. Suppose that we select the winner uniformly at random, i.e., every candidate in S has the same chance of being selected. In this case, knowing the manipulator's preference ordering is not sufficient to determine his optimal strategy. For example, suppose that the manipulator prefers a to b to c, and by voting strategically he can change the output of $\mathcal{F}$ from b to $\{a, c\}$. It is not immediately clear if this manipulation is beneficial. Indeed, if the manipulator strongly prefers a, but is essentially indifferent between b and c, then the answer is probably positive, but if he strongly dislikes c and slightly prefers a to b, the answer is likely to be negative (of course, this also depends on the manipulator's risk attitude).

Thus, to model this situation appropriately, we need to know the utilities that the manipulator assigns to all candidates. Under the assumption of risk neutrality, the manipulator's utility for a set of candidates is equal to his expected utility when a candidate is drawn from this set uniformly at

random, or, equivalently, to his *average* utility for a candidate in this set. Since we are interested in computational issues, we assume that all utilities are positive integers given in binary.

Formally, given a set of candidates C, we assume that the manipulator is endowed with a utility function $u : C \to \mathbb{N}$. This function can be extended to sets of candidates by setting $u(S) = \frac{1}{|S|}\sum_{c\in S} u(c)$ for any $S \subseteq C$.

3.1 Single-Winner Elections

We now define the manipulation problem in the single-winner case. As all voting rules considered in this paper are anonymous, we can fix any voter as the manipulator. In what follows, it will be convenient to assume that the manipulator is voter n.

Definition 3.1. *An instance of the $\mathcal{F}$-RANDMANIPULATION problem is a tuple (E, u, q), where $E = (C, \mathcal{R})$ is an election, $u : C \to \mathbb{N}$ is the manipulator's utility function such that $u(c) \geq u(c')$ if and only if $c \succ_n c'$, and q is a non-negative rational number. It is a "yes"-instance if there exists a vote L such that $u(\mathcal{F}(\mathcal{R}_{-n}, L)) \geq q$ and a "no"-instance otherwise.*

The optimization version of $\mathcal{F}$-RANDMANIPULATION is defined similarly. We remark that $\mathcal{F}$-RANDMANIPULATION is in NP for any polynomial-time computable voting correspondence $\mathcal{F}$: it suffices to guess the manipulative vote L, determine the set $S = \mathcal{F}(\mathcal{R}_{-n}, L)$, and compute the average utility of the candidates in S.

3.2 Multi-Winner Elections

There are settings where voters elect more than one candidate. In that case, ℓ members of C will be named the winners, or the *elected committee*. There are many voting rules that are designed specifically for this setting and aim to select the candidates that best represent the voters (see e.g. Chamberlin and Courant [3]). However, in this paper, we will focus on using scoring rules for the purpose of manipulation; this approach is reasonable when voting is used to select the finalists of a contest, or to allocate grants or fellowships (see, for example the work by Meir et al. [12]). Given a scoring rule, if we want to elect a committee of size ℓ, it is natural to choose the ℓ candidates with the highest score. However, we may still need to break ties. Suppose, for instance, that $\ell = 3$ and we have two candidates whose score is 10 and two candidates whose score is 9. Clearly, the candidates whose score is 10 should be elected no matter what, but we need to choose one of the candidates whose score is 9, e.g., by tossing a fair coin. We will now explain how to formalize this approach.

Fix a scoring rule $\mathcal{F}_\alpha$ with $\alpha = (\alpha_1, \ldots, \alpha_m)$ and an election $E = (C, \mathcal{R})$ with $|C| = m$. Given a candidate $c \in C$, let s_c denote c's score in E under $\mathcal{F}_\alpha$. We say that candidates c and c' are on the same *level* if $s_c = s_{c'}$. There are $p \leq m$ levels, denoted $H_1, \ldots, H_p$; we set $s(H_j)$ to be the score of the candidates in H_j, and assume that $s(H_1) > \ldots > s(H_p)$. Let $W_j = \cup_{q=1}^{j} H_q$. If $|W_j| \leq \ell$, then the tie-breaking rule does not apply to W_j. Formally, let $j_0 = \max\{j \mid |W_j| \leq \ell\}$ and set $\mathcal{W} = W_{j_0}$. The set $\mathcal{W}$ is called the *confirmed set*: these are the candidates who will definitely be in the elected committee. The set $\mathcal{P} = H_{j+1}$ is called the *pending set*: these are the candidates to which we must apply the tie-breaking rule. Note that $|H_1| > \ell$ implies $\mathcal{W} = \emptyset$ and $\mathcal{P} = H_1$, and $|\mathcal{W}| = \ell$ implies $\mathcal{P} = \emptyset$. For single-winner elections ($\ell = 1$) we obtain $\mathcal{P} = \emptyset$ if $|H_1| = 1$, and $\mathcal{P} = H_1$ otherwise. The randomized tie-breaking rule operates by choosing $\ell' = \ell - |\mathcal{W}|$ candidates from the set $\mathcal{P}$ uniformly at random.

We assume that the manipulator's utility is additive, i.e., if a committee $S \subseteq C$ is elected, his utility is given by $\sum_{c\in S} u(c) = |S|u(S)$. Let $T_s(\mathcal{P})$ denote the random variable that takes values in the space of all s-subsets of $\mathcal{P}$, with each subset being equally likely. Given a variable ξ, let $\mathbb{E}[\xi]$ denote its expectation. Then the manipulator's utility in E (under truthful voting) is

$$\sum_{c\in\mathcal{W}} u(c) + \mathbb{E}[\sum_{c\in T_{\ell'}(\mathcal{P})} u(c)],$$

where $\ell' = \ell - |\mathcal{W}|$.

We are now ready to define the computational problem that is associated with manipulating a multi-winner election under randomized tie-breaking with respect to a scoring rule $\mathcal{F}_\alpha$, $\alpha = (\alpha_1, \dots \alpha_m)$. We will refer to this problem as $\mathcal{F}_\alpha$-RANDMULTIMANIPULATION. An instance of this problem is given by an election $E = (C, \mathcal{R})$ with $|C| = m$, a committee size ℓ, the manipulator's utility function $u : C \to \mathbb{N}$, which satisfies $u(c) \geq u(c')$ if and only if $c \succ_n c'$, and a non-negative rational number q. It is a "yes"-instance if there exists a vote L such that the manipulator's utility in $(C, (\mathcal{R}_{-n}, L))$ is at least q and a "no"-instance otherwise.

4 Single-Winner Elections

We begin by analyzing the family of scoring rules. We observe that for any scoring rule, manipulation is easy under randomized tie-breaking.

Theorem 4.1. *For any scoring vector $\alpha = (\alpha_1, \dots, \alpha_m)$ $\mathcal{F}_\alpha$-*RANDMANIPULATION *is in* P.

Theorem 4.1 can be obtained as a corollary of Theorem 5.2 in Section 5 (see [14] for a direct proof). It implies that for scoring rules, assuming that ties are broken uniformly at random does not increase the complexity of manipulation compared to lexicographic tie-breaking.

Similarly, both the classic and the simplified versions of the Bucklin rule can be manipulated in polynomial time; the proof is omitted due to space constraints.

Theorem 4.2. Bucklin-RANDMANIPULATION *and* simplified Bucklin-RANDMANIPULATION *are in* P.

In contrast, if we break ties uniformly at random, manipulation under Maximin becomes NP-hard. In fact, our hardness result holds even for a fairly simple utility function: Let w be the Maximin winner prior to the manipulators vote; if we set $u(w) = 0$, $u(c) = 1$ for $c \in C \setminus \{w\}$, then Maximin-RANDMANIPULATION becomes NP-complete. The proof is omitted due to space constraints.

Theorem 4.3. Maximin-RANDMANIPULATION *is* NP*-complete.*

While Maximin-RANDMANIPULATION is NP-complete in general, there is an efficient algorithm for this problem assuming that the manipulator's utility function has special structure. Specifically, recall that in the model of [2] the manipulator's goal is to make a specific candidate p a winner. This suggests that the manipulator's utility can be modeled by setting $u(p) = 1$, $u(c) = 0$ for all $c \in C \setminus \{p\}$. We will now show that for such utilities, Maximin-RANDMANIPULATION is in P.

Theorem 4.4. *If the manipulator's utility function is given by $u(p) = 1$, $u(c) = 0$ for $c \in C \setminus \{p\}$, Maximin-*RANDMANIPULATION *is in* P.

Proof. Consider an election $E = (C, \mathcal{R})$ with the candidate set $C = \{c_1, \dots, c_m\}$ and recall that n is the manipulating voter. In this proof, we denote by $s(c_i)$ the Maximin score of a candidate $c_i \in C$ in the election $E' = (C, \mathcal{R}')$, where $\mathcal{R}' = \mathcal{R}_{-n}$. Let $s = \max_{c_i \in C} s(c_i)$.

For any $c_i \in C$, the manipulator's vote increases the score of c_i either by 0 or by 1. Thus, if $s(p) < s - 1$, the utility of the manipulator will be 0 irrespective of how he votes.

Now, suppose that $s(p) = s - 1$. The manipulator can increase the score of p by 1 by ranking p first. Thus, his goal is to ensure that after he votes (a) no other candidate gets $s + 1$ point and (b) the number of candidates in $C \setminus \{p\}$ with s points is as small as possible. Similarly, if $s(p) = s$, the manipulator can ensure that p gets $s + 1$ points by ranking him first, so his goal is to rank the remaining candidates so that in $C \setminus \{p\}$ the number of candidates with $s + 1$ points is as small as possible. We will now describe an algorithm that works for both of these cases.

We construct a directed graph G with the vertex set C that captures the relationship among the candidates. Namely, we have an edge from c_i to c_j if there are $s(c_j)$ votes in $\mathcal{R}'$ that rank c_j above

c_i. Observe that, by construction, each vertex in G has at least one incoming edge. We say that c_i is a *parent* of c_j in G whenever there is an edge from c_i to c_j. We remark that if the manipulator ranks one of the parents of c_j above c_j in his vote, then c_j's score does not increase. We say that a vertex c_i of G is *purple* if $s(c_i) = s(p) + 1$, *red* if $s(c_i) = s(p)$ and $c_i \neq p$, and *green* otherwise; note that by construction p is green. Observe also that if $s(p) = s$, there are no purple vertices in the graph. We will say that a candidate c_j is *dominated* in an ordering L (with respect to G) if at least one of c_j's parents in G appears before c_j in L. Thus, our goal is to ensure that the set of dominated candidates includes all purple candidates and as many red candidates as possible.

Our algorithm is based on a recursive procedure $\mathcal{A}$, which takes as its input a graph H with a vertex set $U \subseteq C$ together with a coloring of U into green, red and purple; intuitively, U is the set of currently unranked candidates. It returns "no" if the candidates in U cannot be ranked so that all purple candidates in U are dominated by other candidates in U with respect to H. Otherwise, it returns an ordered list L of the candidates in U in which all purple candidates are dominated, and a set S consisting of all red candidates in U that remain undominated in L with respect to H.

To initialize the algorithm, we call $\mathcal{A}(G)$. The procedure $\mathcal{A}(H)$ is described below (Algorithm 1). We claim that $\mathcal{A}(G)$ outputs "no" if and only if no matter how the manipulator votes, some candidate in $C \setminus \{p\}$ gets $s(p) + 2$ points. Moreover, if $\mathcal{A}(G) = (L, S)$ and the set S contains r red candidates, then for any vote of the manipulator that ensures that all candidates in $C \setminus \{p\}$ have at most $s(p) + 1$ points there are at least r red candidates with $s(p) + 1$ points. We will split the proof of this claim into several lemmas, whose proofs are omitted.

Lemma 4.5. *At any point in the execution of the algorithm, if $\mathcal{A}(H) = (L, S)$, then each candidate in $U \setminus S$ is dominated in H.*

We are now ready to prove that our algorithm correctly determines whether the manipulator can ensure that no candidate gets more than $s(p) + 1$ points.

Lemma 4.6. *The algorithm outputs "no" if and only if for any vote L there is a purple candidate that is undominated.*

It remains to show that the set S output by the algorithm contains as few candidates as possible.

Lemma 4.7. *At any point in the execution of Algorithm 1, if $\mathcal{A}(H) = (L, S)$, then in any ordering of the candidates in U in which each purple vertex in U is dominated, at least $|S|$ red vertices in U are undominated.*

Combining Lemma 4.6 and Lemma 4.7, we conclude that Algorithm 1 outputs (L, S), then L is the optimal vote for the manipulator and if Algorithm 1 outputs "no", then the manipulator's utility is 0 no matter how he votes. Also, it is not hard to see that Algorithm 1 runs in polynomial time. □

We remark that Theorem 4.4 has recently been extended to utility functions that assign utility of 1 to a constant number of candidates and utility of 0 to all other candidates (see [17]).

For the Copeland rule, RANDMANIPULATION is also NP-hard. To show this, we give a reduction from the INDEPENDENT SET problem [8] (proof omitted due to space constraints).

Theorem 4.8. Copeland$^\alpha$-RANDMANIPULATION *is* NP*-complete for any rational* $\alpha \in [0, 1]$.

Some common voting rules, such as, e.g., STV, do not assign scores to candidates; rather, they are defined via multi-step procedures. When one computes the winner under such rules, ties may have to be broken during each step of the procedure. A natural approach to winner determination under such rules is to use the *parallel universes tie-breaking* [4]: a candidate c is an election winner if the intermediate ties can be broken so that c is a winner after the final step. Thus, any such rule defines a voting correspondence in the usual way, and hence the corresponding RANDMANIPULATION problem is well-defined. In this paper we consider three rules in this class, namely, Plurality with Runoff, STV, and Ranked Pairs (we use the definition in [4]).

Algorithm 1: $\mathcal{A}(H)$

$L \leftarrow \emptyset$;
if H *contains* p **then**
  $L \leftarrow [p]$; $H \leftarrow H \setminus \{p\}$;
while H *contains a candidate* c *that is green or has a parent that has already been ranked in the input graph* H **do**
  $L \leftarrow L ::[c]$; $H \leftarrow H \setminus \{c\}$;
`// ::  is the string append operation.`
if $H = \emptyset$ **then**
  return $(L, \emptyset)$;
if *there is a purple candidate in* H *with no parents in* H **then**
  return "no";
if *there is a red candidate* c *in* H *with no parents in* H **then**
  $H' \leftarrow H$ with c colored green;
  $OUT \leftarrow \mathcal{A}(H')$;
  if $OUT =$*"no"* **then**
    return "no"
  $(L', S') \leftarrow OUT$;
  return $(L::L', S' \cup \{c\})$.
Let T be some cycle in H;
`// At this point in the algorithm, each vertex of` H `has a parent, thus there is a cycle in` H
Collapse T;
`// i.e., (a) replace` T `with a single vertex` t`, and (b) for each` $y \notin T$`, add an edge` (t, y) `if` H `contained an edge` (x, y) `for some` $x \in T$ `and add an edge` (y, t) `if` H `contained a vertex` z `with` $(y, z) \in H$
if T *contains at least one red vertex* **then**
  color t red;
else
  color t purple;
$H' \leftarrow H$ after the contraction;
$OUT \leftarrow \mathcal{A}(H')$;
if $OUT =$ *"no"* **then**
  return "no";
$(L', S') \leftarrow OUT$;
if $t \in S'$ **then**
  `//` t `must be red, so` T `contains a red vertex`
  Let c be some red vertex in T;
  Let $\hat{L}$ be an ordering of the vertices in T that starts with c and follows the edges of T;
  Let L'' be the list obtained from L' by replacing t with $\hat{L}$;
  return $(L::L'', (S' \setminus \{t\}) \cup \{c\})$.
else
  `//` $t \notin S$`, so by Lemma 4.5` t `is dominated in` H'
  Let a be a parent of t that precedes it in L';
  Let c be some child of a that appears in T;
  Let $\hat{L}$ be an ordering of the vertices in T that starts with c and follows the edges of T;
  Let L'' be the list obtained from L' by replacing t wish $\hat{L}$;
  return $(L::L'', S')$.

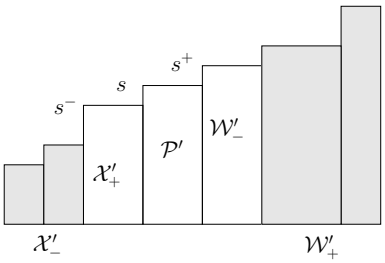

Figure 1: Proof of Theorem 5.1

Proposition 4.9. Plurality with Runoff-RANDMANIPULATION *is in* P.

We omit the proof of Proposition 4.9 due to space constraints; briefly, the main idea of the proof is that the best manipulative vote can be found by placing some candidate first and then ranking the remaining candidates according to their utility.

However, for STV and Ranked Pairs, RANDMANIPULATION is NP-hard. The proof of this fact hinges on an observation that allows us to inherit hardness results from the standard model of voting manipulation. Recall that $\mathcal{F}$-COWINNERMANIPULATION is the computational problem of deciding whether given an election $E = (C, \mathcal{R})$ the manipulator can make a specific candidate $p \in C$ one of the election winners under a voting correspondence $\mathcal{F}$.

Proposition 4.10. *For any voting correspondence $\mathcal{F}$, the problem $\mathcal{F}$-*COWINNERMANIPULATION *many-one reduces to $\mathcal{F}$-*RANDMANIPULATION.

Since for STV and Ranked Pairs COWINNERMANIPULATION is known to be NP-hard (see, respectively, [1] and [16]), we obtain the following corollary.

Corollary 4.11. STV-RANDMANIPULATION *and* Ranked Pairs-RANDMANIPULATION *are* NP-*hard.*

5 Multi-Winner Elections

We now discuss the complexity of manipulating multi-winner elections when ties are broken uniformly at random.

We begin by analyzing the k-Approval voting correspondence. It turns out that k-Approval-RANDMULTIMANIPULATION can be decided in polynomial time.

Theorem 5.1. k-Approval-RANDMULTIMANIPULATION *is in* P.

Proof. Consider the election $(C, \mathcal{R}_{-n})$, and let $\mathcal{P}'$ and $\mathcal{W}'$ be, respectively, the pending set and the confirmed set in this election. Set $\mathcal{X}' = C \setminus (\mathcal{P}' \cup \mathcal{W}')$. Let s^+ be the lowest k-Approval score among the candidates in $\mathcal{W}'$ (set $s^+ = +\infty$ if $\mathcal{W}' = \emptyset$), let s^- be the highest k-Approval score among the candidates in $\mathcal{X}'$ (set $s^- = -\infty$ if $\mathcal{X}' = \emptyset$), and let s be the k-Approval score of the candidates in $\mathcal{P}'$ (if $\mathcal{P}' = \emptyset$, s remains undefined). Let $\mathcal{W}'_- \subseteq \mathcal{W}'$ be the set of candidates whose k-Approval score is s^+, and let $\mathcal{X}'_+ \subseteq \mathcal{X}'$ be the set of candidates whose k-Approval score is s^-; also, set $\mathcal{W}'_+ = \mathcal{W}' \setminus \mathcal{W}'_-$ and $\mathcal{X}'_- = \mathcal{X}' \setminus \mathcal{X}'_+$. Note that $s^- < s^+$, and if $\mathcal{P}' \neq \emptyset$, we have $s^- < s < s^+$.

Let E be the election obtained after the manipulator votes, and suppose that in E the confirmed set is $\mathcal{W}$ and the pending set is $\mathcal{P}$; also, set $\mathcal{X} = C \setminus (\mathcal{W} \cup \mathcal{P})$. We will now argue that, no matter how the manipulator votes, we have $\mathcal{W}'_+ \subseteq \mathcal{W}$ and $\mathcal{X}'_- \subseteq \mathcal{X}$, i.e., points allocated to candidates in $\mathcal{W}'_+ \cup \mathcal{X}'_-$ do not affect the election outcome. Indeed, in E the score of every candidate in $\mathcal{W}'_+$ will

be at least $s^+ + 1$, and there can be at most $|\mathcal{W}'| \leq \ell$ candidates with such score, so every candidate in $\mathcal{W}'_+$ will end up in $\mathcal{W}$. Further, in E the score of every candidate in $\mathcal{X}'_-$ will be at most s^-, and there are at least $|\mathcal{P}'| + |\mathcal{W}'| \geq \ell$ candidates whose score is at least $s^- + 1$, so the score of s^- will be insufficient for being placed in $\mathcal{P}$.

Now, suppose that the manipulator has decided to approve k_w candidates in $\mathcal{W}'_-$. Then, to maximize his utility, he has to approve k_w candidates in $\mathcal{W}'_-$ with the highest utility. A similar argument works for $\mathcal{P}'$ and $\mathcal{X}'_+$. As for the candidates in $\mathcal{W}'_+ \cup \mathcal{X}'_-$, it does not matter which ones he chooses to approve, since, as argued above, his vote will not change the status of these candidates. Thus, the outcome of the election is completely determined by a triple of non-negative integers (k_w, k_p, k_x), where k_w, k_p, and k_x are, respectively, the number of candidates in $\mathcal{W}'_-$, $\mathcal{P}'$, and $\mathcal{X}'_+$ that the manipulator approves. Hence, the manipulator can go over all triples of integers $(k_w, k_p, k_x) \in \{0, \ldots, k\}^3$, and, for each triple, check if it corresponds to a valid vote and compute the expected utility that he obtains from approving k_w highest-utility candidates from $\mathcal{W}'_-$, k_p highest-utility candidates from $\mathcal{P}'$, and k_x highest-utility candidates from $\mathcal{X}'_+$, and distributing the remaining points (if any) among the rest of the candidates. The manipulator can then check if the expected utility from the best such triple is at least q. Clearly, (k_w, k_p, k_x) corresponds to a valid vote if and only if

- $0 \leq k_w \leq |\mathcal{W}'_-|$,
- $0 \leq k_p \leq |\mathcal{P}'|$,
- $0 \leq k_x \leq |\mathcal{X}'_+|$, and
- $0 \leq k - k_w - k_p - k_x \leq |\mathcal{X}'_-| + |\mathcal{W}'_+|$,

and the manipulator's expected utility from any such vote can be computed in time $O(k)$. Thus, the overall running time of our algorithm is $O(k^4)$. Since we can assume that $k \leq m$, this running time is polynomial in the input size. □

We now show that when the size of the committee, ℓ, is bounded by a constant, then $\mathcal{F}_\alpha$-RANDMULTIMANIPULATION is in P for any scoring rule F_α. This immediately implies the single winner case discussed in Section 4.

Theorem 5.2. *$\mathcal{F}_\alpha$-RANDMULTIMANIPULATION is in P when ℓ is bounded by a constant.*

Proof. Fix a scoring rule $\mathcal{F}_\alpha$ with a scoring vector $\alpha = \alpha_1 \geq \ldots \geq \alpha_m$ and an election $(C, \mathcal{R})$ with $|C| = m$, and let $\mathbf{s} = (s_1, \ldots, s_m)$ be the vector of the candidates' scores in $(C, \mathcal{R}_{-n})$. For each $k \leq \ell$ and each subset $\mathcal{W}_k \subseteq C$ of size k, we check if the manipulator can vote so that the confirmed set is $\mathcal{W}_k$. If this is indeed the case, we find the best set of $\ell - k$ pending winners for this choice of $\mathcal{W}_k$; that is, we identify a set $\mathcal{P}_k$ with $|\mathcal{P}_k| > \ell - k$ such that after the manipulator's vote the confirmed set is $\mathcal{W}_k$, the (identical) scores of the candidates in $\mathcal{P}_k$ are strictly less than those of any $c \in \mathcal{W}_k$, and the manipulator's expected utility from $\mathcal{P}_k$ is maximized. Notice that the requirement $|\mathcal{P}_k| > \ell - k$ is necessary; otherwise, $\mathcal{P}_k \cup \mathcal{W}_k$ are the confirmed winners, which contradicts our objective of having $\mathcal{W}_k$ as the confirmed winners. We then compute the manipulator's expected utility from having the candidates in $\mathcal{W}_k$ as the confirmed winners and the candidates in $\mathcal{P}_k$ as the pending winners, and select a triple $(k, \mathcal{W}_k, \mathcal{P}_k)$ that maximizes the manipulator's expected utility.

The candidate set C has at most $\sum_{k=1}^{\ell} \binom{m}{k} \in \mathcal{O}(m^\ell)$ subsets of size at most ℓ; thus, it remains to show that for each subset of size at most ℓ the procedure described in the previous paragraph can be implemented in polynomial time. Fix a $k \leq \ell$ and a set $\mathcal{W}_k$. First, we pick k entries of α; these are the scores that we will assign to candidates in $\mathcal{W}_k$. There are $\binom{m}{k} = \mathcal{O}(m^\ell)$ ways of choosing such a set of scores; we go over all possible choices. We then order the candidates in $\mathcal{W}_k$ by decreasing order of scores under $\mathbf{s}$, and assign the lowest among the selected k scores to the first candidate, the second lowest to the second candidate and so on. If $\mathcal{W}_k$ can be made confirmed winners under some assignment of the k scores selected, then in particular they can be made confirmed winners under

this assignment. Now, let $H_1, \ldots, H_p$ be the levels of the candidates in $C \setminus \mathcal{W}_k$. We renumber the candidates in $C \setminus \mathcal{W}_k$ so that for all $i \in 1, \ldots, p-1$, all candidates in H_i are before the candidates in H_{i+1}. Given a level H_i, we order the candidates in H_i so that if $c, c' \in H_i$ and the manipulator prefers c to c', then c' precedes c. Let $\alpha' = \{\alpha_{i_1}, \ldots, \alpha_{i_{m-k}}\}$ be the remaining $m-k$ scores that the manipulator needs to assign; we assume $\alpha_{i_1} \leq \ldots \leq \alpha_{i_{m-k}}$.

We assign $\alpha_{i_1}, \ldots, \alpha_{i_{|H_1|}}$ to H_1 in that order. Similarly, we assign $\alpha_{i_{|H_1|+1}}, \ldots, \alpha_{i_{|H_1|+|H_2|}}$ to H_2 and so on until all scores are assigned. This assignment, denoted σ_0, ensures that at each level, the manipulator's favorite candidates from that level receive the highest scores. Let Φ be the highest score of any candidate in $C \setminus \mathcal{W}_k$ under σ_0. Observe that for every score assignment to candidates in $C \setminus \mathcal{W}_k$ the score of some candidate in $C \setminus \mathcal{W}_k$ after the manipulator's vote is at least Φ. Thus, if Φ is greater than or equal to the score of some $c \in \mathcal{W}_k$, then $\mathcal{W}_k$ cannot be made confirmed winners using this score assignment, and we proceed to check a different assignment of scores to $\mathcal{W}_k$. Thus, from now one we assume that the score of each candidate in $\mathcal{W}_k$ is greater than Φ. Let $\mathcal{P}_0$ be the set of candidates whose score is Φ after submitting σ_0. We can try to modify σ_0 in order to increase the manipulator's utility, by swapping some candidates in the vote. Note that reassigning scores given to members of $\mathcal{P}_0$ will either result in a non-tied outcome, or decrease the manipulator's expected utility from the set of tied candidates. Indeed, suppose that a candidate $c \in \mathcal{P}_0$ received a score of β and now receives a higher score β'; this increases his score to be strictly more than Φ. If this results in a strictly higher utility for the manipulator, this means that the manipulator can strictly increase his utility by greedily assigning the highest scores in α' to the candidates he prefers the most, with no ties formed. On the other hand, if we assign a lower score to c, this means that some other candidate in a higher level receives a higher score, and the same argument applies. Thus, any swap we make will only involve candidates not in $\mathcal{P}_0$. However, note that the manipulator's utility is unaffected by candidates whose score is less than Φ. Thus, for any candidate c not in $\mathcal{P}_0$, we can just check if there is some score that will give him a total score of Φ. If such $c \in (C \setminus \mathcal{W}_k) \setminus \mathcal{P}_0$ exists, and adding c to $\mathcal{P}_0$ increases the manipulator's expected utility, we can add c to $\mathcal{P}_0$. Having done so for each candidate, we denote the resulting set by $\mathcal{P}_1$. We claim that $\mathcal{P}_1$ is indeed the set of pending candidates we require. However, it is not guaranteed that $|\mathcal{P}_1| > \ell - k$. If it is, then we are done. Otherwise, there are two cases.

Case 1: Given $\mathcal{W}_k$ and the scores we assign $\mathcal{W}_k$, it is impossible to find a score assignment such that $\mathcal{W}_k$ are confirmed winners.

Case 2: Even if there is a set $\mathcal{P}_k$ of pending winners, there is a set $\mathcal{P}'$ of candidates of size exactly $\ell - k$ such that the manipulator's utility from $\mathcal{W}_k \cup \mathcal{P}'$ is at least his expected utility from having $\mathcal{W}$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners.

Observe that both cases imply that if $|\mathcal{P}_1| \leq \ell - k$ we can just move on to another score assignment to $\mathcal{W}_k$, and ignore the current assignment: it is either impossible to have $\mathcal{W}_k$ as the confirmed winners, or there is another candidate set with the same utility that can be made confirmed winners and will be found in some other iteration. We must show that indeed one of these two cases holds.

If neither case holds, there exists a vote σ' such that if the manipulator submits σ', the set of confirmed winners is $\mathcal{W}_k$, the set of pending winners is $\mathcal{P}_k$, and for any set $\mathcal{P}' \subseteq C \setminus \mathcal{W}_k$ such that $|\mathcal{P}'| = \ell - k$ and the set $\mathcal{W}_k \cup \mathcal{P}'$ is a feasible set of winners it holds that the manipulator's expected utility from having $\mathcal{W}$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners is greater than his utility from $\mathcal{W}_k \cup \mathcal{P}'$.

First, consider the case where both confirmed and pending candidates get a total score of more than Φ points. Let $c_{j_1}, \ldots, c_{j_{\ell-k}}$ be the manipulator's most preferred $\ell - k$ candidates in $\mathcal{P}_k$; by assumption, we must have that the manipulator's expected utility from $\mathcal{P}'$ is at most $\sum_{p=1}^{\ell-k} u(c_{j_p})$. Let $\mathcal{S}$ be the set consisting of these $\ell - k$ candidates and $\mathcal{W}_k$. Consider any candidate $c_j \in \mathcal{S}$ and suppose the manipulator grants $\alpha_{j'}$ points to c_j. The score of c_j after the manipulator votes is strictly more than Φ; thus $j' < j$. We set $\mathcal{S}' = \{c_{j'} \in C \mid \alpha_{j'} \text{ is assigned to some } c_j \text{ under } \sigma'\}$.

Now, consider the vote obtained from σ_0 by swapping the votes given to c_j and its corresponding candidate $c_{j'}$. Observe that some candidates can be in two such swaps—once acting as c_j and once as $c_{j'}$—in this case we begin from the swap which uses the candidate as c_j and afterwards we use the candidate who was put on his place for the next swap. All candidates in $C \setminus \mathcal{S} \setminus \mathcal{S}'$ do not have their scores changed, so they still get at most Φ points; more importantly, all candidates in $\mathcal{S}$ now get strictly more than Φ points. Further, all candidates in $\mathcal{S}' \setminus \mathcal{S}$ get less than Φ points. Thus, in this case $\mathcal{S}$ are the confirmed winners and the manipulator's expected utility is at least as high as that from having $\mathcal{W}_k$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners, a contradiction. The other case is when the candidates in $\mathcal{W}_k$ have more than Φ points, but the candidates in $\mathcal{P}_k$ have exactly Φ points. This case is handled similarly; we omit the details due to space constraints. □

6 Conclusions

Implementing a randomized tie-breaking rule proves to be an interesting new direction in computational social choice. Some voting rules (such as scoring rules and Bucklin) remain manipulable when employing randomized tie-breaking; however, computational barriers to manipulation arise for Copeland and Maximin. We also show that the target committee size does not affect the complexity of manipulating k-Approval, and procedures for choosing a constant-size committee that are based on scoring rules are manipulable as well.

While the picture for the single winner case is fairly complete, some problems in the multi-winner case remain open. For example, it is unclear whether $\mathcal{F}_\alpha$-MULTIRANDMANIPULATION remains in P if the size of the committee is unbounded, apart from the special case shown here for k-Approval. Moreover, the effects of randomized tie-breaking on coalitional manipulation are also unclear. While the hardness results shown in our work immediately imply hardness for coalitional manipulation under the same voting rules, the easiness results do not easily generalize.

To conclude, tie-breaking rules strongly influence the manipulability of elections; even when they do not induce hardness of manipulation, the techniques required in order to manipulate under randomized tie-breaking are quite different from those employed for lexicographic tie-breaking. This suggests that the choice of a tie-breaking rule is an important aspect of designing a good voting system and should not be ignored.

	P	NP-hard
Single-Winner ($\ell = 1$)	Plurality w/Runoff Maximin (restricted) Simplified Bucklin Classic Bucklin	Copeland Maximin (general) STV Ranked Pairs
Multi-Winner ($\ell \geq 1$)	Scoring Rules (for constant ℓ) k-Approval	

Table 1: Complexity of RANDMANIPULATION and MULTIRANDMANIPULATION for classic voting rules.

References

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

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

[3] J. R. Chamberlin and P. N. Courant. Representative deliberations and representative decisions: Proportional representation and the borda rule. *The American Political Science Review*, 77(3):718–733, 1983.

[4] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *IJCAI'09*, pages 109–115, 2009.

[5] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *ACM EC'10*, pages 347–356, 2010.

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

[7] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31:53–64, 2010.

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

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

[10] A. Gibbard. Manipulation of schemes that mix voting and chance. *Econometrica*, 45:665–681, 1977.

[11] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. Evaluation of election outcomes under uncertainty. In *AAMAS'08*, pages 959–966, 2008.

[12] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, September 2008.

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

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

[15] M. Satterthwaite. 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, 1975.

[16] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *IJCAI'09*, pages 348–352, 2009.

[17] M. Zuckerman and J. Rosenschein. Manipulation with randomized tie-breaking under Maximin (extended abstract). In *AAMAS'12*, pages 1315–1317, 2012.

Svetlana Obraztsova, Yair Zick, Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
Email: svet0001,yair0001@e.ntu.edu.sg;eelkind@ntu.edu.sg