

On the Complexity of Voting Manipulation under Randomized Tie-Breaking

Svetlana Obraztsova Edith Elkind

School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore

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.*, 1989]. Most of the existing work in this area, including that of [Bartholdi *et al.*, 1989], 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. However, till recently, an equally appealing method of tie-breaking, namely, selecting the winner uniformly at random among all tied candidates, has not been considered in the computational social choice literature. The first paper to analyze the complexity of voting manipulation under randomized tie-breaking is [Obraztsova *et al.*, 2011], where the authors provide polynomial-time algorithms for this problem under scoring rules and—under an additional assumption on the manipulator’s utilities—for Maximin. In this paper, we extend the results of [Obraztsova *et al.*, 2011] by showing that finding an optimal vote under randomized tie-breaking is computationally hard for Copeland and Maximin (with general utilities), as well as for STV and Ranked Pairs, but easy for the Bucklin rule and Plurality with Runoff.

1 Introduction

Whenever a group of agents with heterogeneous preferences 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, and the entities that the agents need to select from can vary from political leaders to song contest winners and joint plans. The standard way to aggregate the 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 [Gibbard, 1973; Satterthwaite, 1975] 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 [Faliszewski and Procaccia, 2010] 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 [Ephrati and Rosenschein, 1997]. 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 in the computational social choice literature (two exceptions to this

pattern that we are aware of are [Hazon *et al.*, 2008] and [Desmedt and Elkind, 2010]; however, [Hazon *et al.*, 2008] does not deal with manipulation at all and [Desmedt and Elkind, 2010] 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? A very recent paper [Obraztsova *et al.*, 2011] deals with this issue by augmenting the manipulator’s preference model: it assumes 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., [Gibbard, 1977]) and has also been used in [Desmedt and Elkind, 2010]. [Obraztsova *et al.*, 2011] show that in this setting any scoring rule is easy to manipulate, and so is the Maximin rule (see Section 2 for the definitions), assuming that the manipulator assigns 1 unit of utility to one candidate and utility 0 to all other candidates. However, [Obraztsova *et al.*, 2011] provides no results on the complexity of manipulating Maximin for general utilities, nor does it analyze the complexity of manipulation under randomized tie-breaking for any other voting rules.

In this paper, we pursue the line of enquiry initiated by [Obraztsova *et al.*, 2011]. We answer the open question posed in that paper by showing that for general utilities Maximin is hard to manipulate (Section 3). We then show that this is also the case for the Copeland rule (Section 4). In contrast, for the Bucklin rule the manipulator has a polynomial-time algorithm (Section 5); this is true both for the classic Bucklin rule considered by the social choice theorists and for its simplified version that is sometimes used in the computational social choice literature (see, e.g., [Xia *et al.*, 2009]). In Section 6, we analyze the complexity of our problem for three voting rules that compute the winners using a multi-step procedure, namely, Plurality with Runoff, STV, and Ranked Pairs. Thus, together with the results of [Obraztsova *et al.*, 2011], we obtain 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).

2 Preliminaries

An *election* is given by a set of *candidates* C , $|C| = m$, and a set of *voters* $V = \{v_1, \dots, v_n\}$. Let $\mathcal{L}(C)$ denote the space of all linear orders over C . Each voter v_i is described by an order $R_i \in \mathcal{L}(C)$, also denoted by \succ_i ; this order is called v_i ’s *preference order*. The vector $\mathcal{R} = (R_1, \dots, R_n)$, where $R_i \in \mathcal{L}(C)$ for all $i = 1, \dots, n$, is called a *preference profile*. When $a \succ_i b$ for some $a, b \in C$, we say that voter v_i *prefers* a to b .

A *voting correspondence* \mathcal{F} is a mapping that, given a preference profile \mathcal{R} over C outputs a non-empty subset $S \subseteq C$; we write $S = \mathcal{F}(\mathcal{R})$. If $|\mathcal{F}(\mathcal{R})| = 1$, the mapping \mathcal{F} is called a *voting rule*; if this is the case, we abuse notation and write $\mathcal{F}(\mathcal{R}) = c$ instead of $\mathcal{F}(\mathcal{R}) = \{c\}$. To transform a voting

correspondence into a voting rule, one needs a *tie-breaking rule*, i.e., a mapping T that given a non-empty set of candidates $S \subseteq C$ outputs a candidate $c \in S$: clearly, if \mathcal{F} is a voting correspondence and T is a tie-breaking rule, then the mapping $T \circ \mathcal{F}$ given by $T \circ \mathcal{F}(\mathcal{R}) = T(\mathcal{F}(\mathcal{R}))$ is a voting rule. Throughout most of this paper, we consider the tie-breaking rule that given a set of tied candidates S , chooses an element of S uniformly at random; we will refer to this rule as the *randomized tie-breaking rule*.

Voting rules We will now describe the voting rules (correspondences) considered in this paper. For all rules that assign scores to candidates (i.e., Copeland, Maximin and k -approval), the winners are the candidates with the highest scores (for Copeland and k -approval, the candidate’s score is the total number of points he obtains). We omit the definition of the Ranked Pairs rule, as our hardness proof for this rule does not directly rely on its definition.

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

k -approval, Plurality and Bucklin Under the k -approval rule, a candidate gets one point for each voter that ranks him in the top k positions; 1-approval is also known as Plurality. Let k^* be the smallest value of k such that some candidate’s k -approval score is at least $\lfloor n/2 \rfloor + 1$; we will say that k^* is the *Bucklin winning round*. Under the simplified Bucklin rule, the winners are all candidates whose k^* -approval score is at least $\lfloor n/2 \rfloor + 1$; under the Bucklin rule, the winners are all k^* -approval winners.

Plurality with Runoff and STV Under the STV rule, the election proceeds in rounds. During each round, the candidate with the lowest Plurality score is eliminated, and the candidates’ Plurality 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 6.

Manipulation Given a preference profile \mathcal{R} over a set of candidates C , for any preference order $L \in \mathcal{L}(C)$ we denote by (\mathcal{R}_{-i}, L) the preference profile obtained from \mathcal{R} by replacing R_i with L . We say that a voter v_i can successfully *manipulate* an election (C, V) with a preference profile (R_1, \dots, R_n) with respect to a voting rule \mathcal{F} if $\mathcal{F}(\mathcal{R}_{-i}, L) \succ_i \mathcal{F}(\mathcal{R})$. To define a notion of successful manipulation for voting correspondences with respect to randomized tie-breaking, we follow [Obraztsova *et al.*, 2011] and assume that the manipulator v_i is endowed with a utility function $u : C \rightarrow \mathbb{N}$ that is *consistent* with v_i , i.e., $u(c) \geq u(c')$ if and only if $c \succ_i c'$.

Then, if a voting correspondence \mathcal{F} outputs a set $S \subseteq C$, v_i 's expected utility $\hat{u}(S)$ is given by $\hat{u}(S) = \frac{1}{|S|} \sum_{c \in S} u(c)$. We say that v_i can *successfully manipulate* \mathcal{F} under *randomized tie-breaking* if $\hat{u}(\mathcal{F}(\mathcal{R}_{-i}, L)) > \hat{u}(\mathcal{F}(\mathcal{R}))$ for some $L \in \mathcal{L}(C)$; a vote $L \in \mathcal{L}(C)$ is *optimal* for v_i with respect to \mathcal{F} if $\hat{u}(\mathcal{F}(\mathcal{R}_{-i}, L)) \geq \hat{u}(\mathcal{F}(\mathcal{R}_{-i}, L'))$ for all $L' \in \mathcal{L}(C)$. The corresponding algorithmic problem is defined as follows.

Definition 1. *An instance of the \mathcal{F} -RANDMANIPULATION problem is a tuple $(E, \mathcal{R}, v_i, u, q)$, where $E = (C, V)$ is an election, \mathcal{R} is a preference profile for E , v_i is a voter in V , $u : C \rightarrow \mathbb{N}$ is a utility function such that $u(c) \geq u(c')$ if and only if $c \succ_i c'$, and q is a non-negative rational number. It is a “yes”-instance if there exists a vote L such that $\hat{u}(\mathcal{F}(\mathcal{R}_{-i}, L)) \geq q$ and a “no”-instance otherwise.*

In the optimization version of \mathcal{F} -RANDMANIPULATION, the goal is to find an optimal vote.

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 $S = \mathcal{F}(\mathcal{R}_{-i}(L))$, and compute the average utility of the candidates in S .

3 Maximin

[Obraztsova *et al.*, 2011] show that if the manipulator's utility function is given by $u(p) = 1$, $u(c) = 0$ for $c \in C \setminus \{p\}$, i.e., the manipulator likes one candidate and equally dislikes all other candidates, then Maximin-RANDMANIPULATION is polynomial-time solvable. We will now show that if we “invert” the manipulator's utility, i.e., set $u(w) = 0$, $u(c) = 1$ for $c \in C \setminus \{w\}$, then Maximin-RANDMANIPULATION becomes NP-complete. Observe that if the manipulator v has this utility function, and w is the Maximin winner irrespective of v 's vote, then v 's goal is to maximize the overall number of Maximin winners.

Our hardness proof proceeds by reduction from FEEDBACK VERTEX SET [Garey and Johnson, 1979]. Recall that an instance of FEEDBACK VERTEX SET is given by a directed graph \mathcal{G} with s vertices $\{\nu_1, \dots, \nu_s\}$ and a parameter $t \leq s$; it is a “yes”-instance if it is possible to delete at most t vertices from \mathcal{G} so that the resulting graph contains no directed cycles and a “no”-instance otherwise. It will be convenient to assume that \mathcal{G} contains no directed cycles of length 2. It is easy to see that FEEDBACK VERTEX SET remains NP-hard under this assumption; we omit the proof of this fact due to space constraints.

Theorem 1. *Maximin-RANDMANIPULATION is NP-complete.*

Proof. We have argued that Maximin-RANDMANIPULATION is in NP. For the hardness proof, suppose that we are given an instance (\mathcal{G}, t) of FEEDBACK VERTEX SET, where \mathcal{G} is an s -vertex graph with the vertex set $\{\nu_1, \dots, \nu_s\}$ that has no directed 2-cycles. We will now construct an instance of our problem with $C = \{c_1, c_2, \dots, c_s, w\}$.

By (the proof of) McGarvey's theorem [McGarvey, 1953], there exists an election $E = (C, V)$ with a preference profile $\mathcal{R}' = (R_1, \dots, R_n)$, where n is even, such that

- for $i = 1, \dots, s$, if the indegree of ν_i in \mathcal{G} is at least 1, then exactly $n/2$ voters rank w above c_i ; otherwise, exactly $n/2 + 1$ voter ranks w above c_i .
- if $(\nu_i, \nu_j) \in \mathcal{G}$ (and hence, since \mathcal{G} contains no directed cycles of length 2, $(\nu_j, \nu_i) \notin \mathcal{G}$), exactly $n/2 + 1$ voters rank c_i above c_j .
- if $(\nu_i, \nu_j) \notin \mathcal{G}$ and $(\nu_j, \nu_i) \notin \mathcal{G}$, exactly $n/2$ voters rank c_i above c_j .

Moreover, $\mathcal{R}' = (R_1, \dots, R_n)$ can be constructed in time polynomial in s . We will say that c_i is a *parent* of c_j if exactly $n/2 + 1$ voter ranks c_i above c_j . Observe that in the resulting election the Maximin score of w is $n/2$, and the Maximin score of any other candidate is $n/2 - 1$.

Set $V' = V \cup \{v_{n+1}\}$, and consider the election $E' = (C, V')$ with a preference profile $(\mathcal{R}', L) = (R_1, \dots, R_n, L)$, where L is the manipulator's vote. Since w is the unique Maximin winner before the manipulator votes, and w 's score exceeds the score of any other candidate by 1, a candidate c_i is a winner of (\mathcal{R}', L) if and only if (a) the manipulator ranks c_i above all of her parents and (b) w 's Maximin score does not increase; on the other hand, w will remain the Maximin winner no matter how the manipulator votes.

Let the manipulator's utility be given by $u(w) = 0$, $u(c) = 1$ for any $c \in C \setminus \{w\}$. Under this utility function, the manipulator's utility is 0 if w is the only Maximin winner, 1 if w is not among the Maximin winners, and $r/(r+1)$ if the Maximin winners are w and r candidates from $C \setminus \{w\}$. Let R_{n+1} be some preference order over C that is consistent with u , and set $\mathcal{R} = (R_1, \dots, R_n, R_{n+1})$. We claim that (\mathcal{G}, t) is a “yes”-instance of FEEDBACK VERTEX SET if and only if $(E', \mathcal{R}, v_{n+1}, u, (s-t)/(s-t+1))$ is a “yes”-instance of Maximin-RANDMANIPULATION.

Suppose (\mathcal{G}, t) is a “yes”-instance of FEEDBACK VERTEX SET. Then we can delete t vertices from \mathcal{G} so that the resulting graph \mathcal{G}' is acyclic, and hence can be topologically sorted. Let $\nu_{i_1}, \dots, \nu_{i_{s-t}}$ be the vertices of \mathcal{G}' , listed in the sorted order, i.e., so that any edge of \mathcal{G} is of the form $(\nu_{i_j}, \nu_{i_\ell})$ with $j < \ell$. Consider the vote L obtained by ranking the candidates that correspond to vertices of \mathcal{G}' first, in reverse topological order (i.e., $c_{i_{s-t}}, \dots, c_{i_1}$), followed by the remaining candidates in $C \setminus \{w\}$, followed by w . By construction, each of the first $s-t$ candidates is ranked above all of its parents, so its Maximin score in (\mathcal{R}', L) is $n/2$. On the other hand, w 's score remains equal to $n/2$. Thus, the manipulator's utility in the resulting election is at least $(s-t)/(s-t+1)$.

Conversely, suppose the manipulator submits a vote L' so that in the preference profile $\mathcal{R}_{-(n+1)}(L')$ his utility is at least $(s-t)/(s-t+1)$. We have argued that w is a Maximin winner in $\mathcal{R}_{-(n+1)}(L')$, and therefore $\mathcal{R}_{-(n+1)}(L')$ has at least $s-t+1$ Maximin winners (including w). Let C' be a set of some $s-t$ candidates in $C \setminus \{w\}$ that are Maximin winners in $\mathcal{R}_{-(n+1)}(L')$, and suppose they appear in L' ordered as $c_{i_1}, \dots, c_{i_{s-t}}$. Let \mathcal{G}' be the induced subgraph of \mathcal{G} with the set of vertices $\nu_{i_1}, \dots, \nu_{i_{s-t}}$. Each of the candidates in C' appears in L' before all of its parents. Therefore, in the ordering $\nu_{i_1}, \dots, \nu_{i_{s-t}}$ of the vertices of \mathcal{G}' all arcs are directed from right to left, i.e., \mathcal{G}' contains no directed cycles. Since

\mathcal{G}' has $s - t$ vertices, this means that (\mathcal{G}, t) is a “yes”-instance of FEEDBACK VERTEX SET. \square

4 Copeland

For the Copeland rule, we give an NP-hardness reduction from the INDEPENDENT SET problem [Garey and Johnson, 1979]. An instance of this problem is given by an undirected graph \mathcal{G} and a positive integer t . It is a “yes”-instance if \mathcal{G} contains an independent set of size at least t , i.e., if \mathcal{G} has at least t vertices such that no two of them are connected by an edge; otherwise, it is a “no”-instance.

Our reduction makes use of a technical lemma (proof omitted due to space constraints), which essentially shows that any undirected graph \mathcal{G} can be obtained as a graph of ties in an election whose size is polynomial in the size of \mathcal{G} ; a similar result appears in [Faliszewski *et al.*, 2008] (Lemma 2.4).

Lemma 1. *Let \mathcal{G} be an undirected graph with the vertex set ν_1, \dots, ν_s , $s \geq 3$. Let $d(\nu_i)$ denote the degree of vertex ν_i . Then there exists a directed graph \mathcal{G}' with the vertex set $G \cup Z \cup \{w\}$, where $G = \{g_1, \dots, g_s\}$, $Z = \{z_1, \dots, z_{4s+1}\}$, such that the outdegree d_{out} and the indegree d_{in} of each vertex of \mathcal{G}' satisfy*

- $d_{\text{out}}(w) = 4s + 1, d_{\text{in}}(w) = s;$
- $d_{\text{out}}(g_i) = 4s + 1 - d(\nu_i), d_{\text{in}}(g_i) = s$ for $i = 1, \dots, s;$
- $d_{\text{in}}(z) + d_{\text{out}}(z) = 5s + 1$ and $d_{\text{out}}(z) \leq 3s + 1$ for all $z \in Z,$

\mathcal{G}' contains no 2-cycles, and, furthermore for $i, j \in \{1, \dots, s\}$, g_i and g_j are not connected by an arc in \mathcal{G}' if and only if there is an edge between ν_i and ν_j in \mathcal{G} .

We are now ready to present the main result of this section.

Theorem 2. *Copeland $^\alpha$ -RANDMANIPULATION is NP-complete for any rational $\alpha \in [0, 1]$.*

Proof. Fix $\alpha \in [0, 1]$ We have argued that Copeland $^\alpha$ -RANDMANIPULATION is in NP. For the hardness proof, suppose that we are given an instance (\mathcal{G}, t) of INDEPENDENT SET, where \mathcal{G} is a graph with the vertex set $\{\nu_1, \dots, \nu_s\}$. We will now construct an instance of our problem with a set of candidates $C = G \cup Z \cup \{w\}$, where $G = \{g_1, g_2, \dots, g_s\}$, $Z = \{z_1, \dots, z_{4s+1}\}$.

Given two candidates $x, y \in C$ in an n -voter election, we say that x *safely wins* a pairwise election against y (and y *safely loses* a pairwise election against x) if at least $\lfloor n/2 \rfloor + 2$ voters prefer x to y . For any candidate $x \in C$, let $\text{SW}(x)$ and $\text{SL}(x)$ denote the number of pairwise elections that x safely wins and safely loses, respectively.

Let $d(\nu_i)$ denote the degree of the vertex ν_i in \mathcal{G} . By Lemma 1 and McGarvey’s theorem [McGarvey, 1953], we can construct an election $E = (C, V)$ with a preference profile (R_1, \dots, R_n) that has the following properties:

- $\text{SW}(w) = 4s + 1, \text{SL}(w) = s;$
- $\text{SW}(g_i) = 4s + 1 - d(\nu_i), \text{SL}(g_i) = s$ for $i = 1, \dots, s;$
- $\text{SW}(z) + \text{SL}(z) = 5s + 1$ and $\text{SW}(z) \leq 3s + 1$ for any $z \in Z;$

- there is a tie between two candidates c and c' if and only if $c = g_i, c' = g_j$ for some $i, j \in \{1, \dots, s\}$ and there is an edge between ν_i and ν_j in \mathcal{G} .

Consider an election $E' = (C, V')$ with $V' = V \cup \{v_{n+1}\}$, and a utility function u given by $u(w) = 0, u(z) = 0$ for any $z \in Z, u(g) = 1$ for any $g \in G$. Let \mathcal{R} be a preference ordering that is consistent with u , and set $\mathcal{R} = (R_1, \dots, R_n, R)$. For any $L \in \mathcal{L}(C)$, in the preference profile $\mathcal{R}_{-(n+1)}(L)$ the Copeland $^\alpha$ score of w is $4s + 1$, and the Copeland $^\alpha$ score of each candidate $z \in Z$ is at most $3s + 1$. Moreover, the Copeland $^\alpha$ score of each $g_i \in G$ is at least $4s + 1 - d(\nu_i)$ and at most $4s + 1$; to ensure that g_i ’s score is $4s + 1$, the manipulator must rank g_i above all of the candidates that g_i is tied with in E (note that for $\alpha = 1$ all candidates in G are currently tied with w , but some of them will lose points after the manipulator votes). We claim that (\mathcal{G}, t) is a “yes”-instance of INDEPENDENT SET if and only if $(E', \mathcal{R}, v_{n+1}, u, t/(t+1))$ is a “yes”-instance of Copeland $^\alpha$ -RANDMANIPULATION.

Indeed, let $J = \{\nu_{i_1}, \dots, \nu_{i_t}\}$ be an independent set in \mathcal{G} . Consider a vote L that ranks the candidates g_{i_1}, \dots, g_{i_t} first (in any order), followed by the remaining candidates in $G \cup Z$, followed by w . Clearly, in the resulting election the Copeland $^\alpha$ score of the top t candidates in L is $4s + 1$, so the manipulator’s utility is at least $t/(t+1)$.

Conversely, suppose that for some $L' \in \mathcal{L}(C)$ the manipulator’s utility is at least $t/(t+1)$. Let S' be the set of all candidates in G whose Copeland $^\alpha$ score in $\mathcal{R}_{-(n+1)}(L')$ is $4s + 1$; we have $|S'| \geq t$. As argued above, the manipulator ranks each candidate $g \in S'$ above all candidates that g is tied with in E . This implies that two candidates in S' cannot be tied in E , i.e., S' corresponds to an independent set in \mathcal{G} . \square

5 Bucklin

In this section, we describe a polynomial-time algorithm for Bucklin-RANDMANIPULATION. We will focus on the simplified Bucklin rule, and omit the term “simplified” throughout this section; in the end, we will briefly explain how to extend our algorithm to the classic Bucklin rule. We first need some additional notation. Consider an election $E = (C, V)$ with $|C| = m$ and the preference profile $\mathcal{R} = (R_1, \dots, R_{n+1})$, and suppose that v_{n+1} is the manipulating voter whose utility function is u . Set $V' = V \setminus \{v_{n+1}\}$, $E' = (C, V')$ and $\mathcal{R}' = (R_1, \dots, R_n)$. For any $c \in C$, let $s_k(c)$ denote c ’s k -approval score in \mathcal{R}' . Given a $L \in \mathcal{L}(C)$, let $S(L)$ be the set of Bucklin winners in $(\mathcal{R}_{-(n+1)}, L)$.

Let $\ell = \min\{k \mid s_k(c) \geq \lfloor \frac{n+1}{2} \rfloor + 1 \text{ for some } c \in C\}$, and set $D = \{c \in C \mid s_\ell(c) \geq \lfloor \frac{n+1}{2} \rfloor + 1\}$. Clearly, for any $L \in \mathcal{L}(C)$, if k is the Bucklin winning round in $(\mathcal{R}_{-(n+1)}, L)$, then $k \leq \ell$. For each $i = 1, \dots, m$, let $C_i = \{c \in C \mid s_i(c) = \lfloor \frac{n+1}{2} \rfloor, s_{i-1}(c) < \lfloor \frac{n+1}{2} \rfloor\}$, and set $C_{<i} = \bigcup_{j < i} C_j$ if $i < \ell$ and $C_{\leq \ell} = (\bigcup_{j < \ell} C_j) \setminus D$.

Suppose that $i \leq \ell$. If the manipulator ranks a candidate $c \in C_i$ in position i or higher, and ranks each candidate in $C_{<i}$ in position i or lower, in the resulting election i is the Bucklin winning round, and c is a Bucklin winner. Conversely, if $i \leq \ell$ is the Bucklin winning round in $(\mathcal{R}_{-(n+1)}, L)$ and a candidate c is a Bucklin winner, then one

of the following conditions holds: (a) $c \in C_i$ and c is ranked in position i or higher in L , or (b) $c \in C_{<i}$ and c is ranked in position i in L , or (c) $i = \ell$ and $c \in D$.

For $i \leq \ell$ and $s \leq m$, let $\mathcal{L}_{i,s}$ denote the set of all votes $L \in \mathcal{L}(C)$ such that (a) i is the Bucklin winning round in $(\mathcal{R}_{-(n+1)}, L)$ and (b) $|S(L) \cap C_i| = s$. Also, let $\mathcal{L}_{i,s}^* = \arg \max\{\hat{u}(S(L)) \mid L \in \mathcal{L}_{i,s}\}$ be the set of utility-maximizing votes in $\mathcal{L}_{i,s}$.

We will now explain how to find a vote in $\mathcal{L}_{i,s}^*$. First, we will show that if $L^* \in \mathcal{L}_{i,s}^*$ and the set $S(L^*)$ contains some candidate $c \in C_{<i}$, then c is the top candidate in $C_{<i}$.

Lemma 2. *If $L^* \in \mathcal{L}_{i,s}^*$ for some $i \leq \ell$ and $s \leq m$ and $S(L^*) \cap C_{<i} \neq \emptyset$, then $|S(L^*) \cap C_{<i}| = 1$ and $S(L^*) \cap C_{<i} \in \arg \max\{u(c) \mid c \in C_{<i}\}$.*

Proof. Fix a vote $L^* \in \mathcal{L}_{i,s}^*$, and let c be a candidate in $S(L^*) \cap C_{<i}$. Since i is the Bucklin winning round for L^* and $c \in C_{<i}$, c cannot be ranked in position $i-1$ or higher in L^* . Further, since $c \in S(L^*)$ and i is the Bucklin winning round for L^* , c cannot be ranked in position $i+1$ or lower in L^* (here, for $i = \ell$ it is crucial that the set $C_{<\ell}$ does not contain candidates in D). Hence, c is ranked in position i in L^* , so $|S(L^*) \cap C_{<i}| = 1$. Now, if $c \notin \arg \max\{u(c) \mid c \in C_{<i}\}$, consider the vote L' obtained from L^* by swapping c with some candidate $b \in \arg \max\{u(c) \mid c \in C_{<i}\}$. We have $L' \in \mathcal{L}_{i,s}$. Further, the argument above shows that $b \notin S(L^*)$, so $S(L') = (S(L^*) \setminus \{c\}) \cup \{b\}$ and hence $\hat{u}(S(L')) > \hat{u}(S(L^*))$, a contradiction. \square

Now, we use Lemma 2 to find a vote in $\mathcal{L}_{i,s}^*$.

Lemma 3. *For any $i \leq \ell$ and any $s \leq |C|$, there is a polynomial-time algorithm that checks whether $\mathcal{L}_{i,s}$ is non-empty, and, if so, identifies a vote $L^* \in \mathcal{L}_{i,s}^*$.*

Proof. Let $\mathcal{L}_{i,s}^1$ be the set of all votes L in $\mathcal{L}_{i,s}$ such that $S(L) \cap C_{<i} \neq \emptyset$, and let $\mathcal{L}_{i,s}^2 = \mathcal{L}_{i,s} \setminus \mathcal{L}_{i,s}^1$. We will identify the best vote in $\mathcal{L}_{i,s}^1$ and $\mathcal{L}_{i,s}^2$ and output the better of the two. Observe that either or both of $\mathcal{L}_{i,s}^1$ and $\mathcal{L}_{i,s}^2$ can be empty: if both are empty, then so is $\mathcal{L}_{i,s}$, and if $\mathcal{L}_{i,s}^j$ is empty, but $\mathcal{L}_{i,s}^{3-j}$ is not, we output the best vote in $\mathcal{L}_{i,s}^{3-j}$.

If $C_{<i} \neq \emptyset$, let b_i be some candidate in $\arg \max\{u(c) \mid c \in C_{<i}\}$. By Lemma 2, to find the best vote in $\mathcal{L}_{i,s}^1$, we place b_i in position i . Now, we need to place s candidates from C_i in top $i-1$ positions. Clearly, if $|C_i| < s$ or if $s > i-1$, this is impossible, so $\mathcal{L}_{i,s}^1 = \emptyset$. Otherwise, we pick s candidates in C_i with the highest utility, breaking ties arbitrarily, and rank them in top s positions in the vote. We then fill the remaining $i-1-s$ positions above i with candidates from $C \setminus (C_i \cup C_{<i})$; again, if $|C \setminus (C_i \cup C_{<i})| < i-1-s$, then $\mathcal{L}_{i,s}^1 = \emptyset$. The remaining candidates can be ranked arbitrarily. It is easy to see that the resulting vote L_1 is in $\mathcal{L}_{i,s}^1$, and, moreover, $\hat{u}(S(L_1)) \geq \hat{u}(S(L'))$ for any $L' \in \mathcal{L}_{i,s}^1$.

The procedure for finding the best vote in $\mathcal{L}_{i,s}^2$ is similar. By the same argument as in the previous case, if $|C_i| < s$ or $s > i$ or $|C \setminus C_{<i+1}| < i-s$, then $\mathcal{L}_{i,s}^2$ is empty. Otherwise, we pick s candidates in C_i with the highest utility, rank

them in top s positions in the vote, rank some candidates from $C \setminus (C_i \cup C_{<i})$ in the next $i-s$ positions, and then rank the remaining candidates arbitrarily. The resulting vote L_2 satisfies $\hat{u}(S(L_2)) \geq \hat{u}(S(L'))$ for any $L' \in \mathcal{L}_{i,s}^2$. \square

Using Lemma 3, we can simply find the best vote in $\mathcal{L}_{i,s}$ for all $i = 1, \dots, \ell$, $s = 0, \dots, m$; while for many values of i and s the set $\mathcal{L}_{i,s}$ is empty, we have $\mathcal{L}_{i,s} \neq \emptyset$ for some $i \leq \ell$, $s \leq m$. We obtain the following result.

Theorem 3. *Simplified Bucklin-RANDMANIPULATION is polynomial-time solvable.*

To extend our algorithm to the classic Bucklin rule, observe that if $L \in \mathcal{L}_{i,s}$ for some $i < \ell$, then each Bucklin winner in $(\mathcal{R}_{-(n+1)}, L)$ has the same i -approval score (namely, $\lfloor \frac{n+1}{2} \rfloor + 1$), so any Bucklin winner in $(\mathcal{R}_{-(n+1)}, L)$ is also a simplified Bucklin winner in $(\mathcal{R}_{-(n+1)}, L)$. Thus, only the case $i = \ell$ has to be handled differently. In this case, it matters which candidates in D are ranked in top ℓ positions by the manipulator, as this affects their ℓ -approval score. Despite these additional complications, the best vote in $\bigcup_{s=0}^m \mathcal{L}_{\ell,s}$ with respect to the classic Bucklin rule can be identified efficiently; we omit the details due to space constraints.

Theorem 4. *Bucklin-RANDMANIPULATION is polynomial-time solvable.*

6 Iterative Rules

Some of the 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* [Conitzer *et al.*, 2009]: 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 a usual way, and hence the corresponding RANDMANIPULATION problem is well-defined. In this section, we consider three rules in this class, namely, Plurality with Runoff, STV, and Ranked Pairs (to save space, we omit the definition of the Ranked Pairs rule, as it is not essential for the presentation).

For Plurality with Runoff, RANDMANIPULATION turns out to be in P. The main idea of the proof is that if \mathcal{L}_c is the set of all votes that rank a candidate $c \in C$ first, then the best vote in \mathcal{L}_c ranks all candidates other than c according to their utility; we omit the full proof due to space constraints.

Theorem 5. *Plurality with Runoff-RANDMANIPULATION is polynomial-time solvable.*

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.

Let \mathcal{F} be a voting correspondence. In the \mathcal{F} -COWINNER-MANIPULATION problem, we are given an election $E = (C, V)$ with a preference profile $\mathcal{R} = (R_1, \dots, R_n)$, and a preferred candidate $p \in P$. The question is whether there exists a vote $L \in \mathcal{L}(C)$ such that the preference profile

$\mathcal{R}' = (R_1, \dots, R_n, L)$ satisfies $p \in \mathcal{F}(\mathcal{R}')$. For STV and Ranked Pairs, COWINNERMANIPULATION is known to be NP-hard (see, respectively, [Bartholdi *et al.*, 1989] and [Xia *et al.*, 2009]). It is easy to see that this implies that for these rules RANDMANIPULATION is hard as well; we omit the proof due to space constraints.

Proposition 1. *For any voting correspondence \mathcal{F} , the problems \mathcal{F} -COWINNERMANIPULATION many-one reduces to \mathcal{F} -RANDMANIPULATION.*

Corollary 1. *STV-RANDMANIPULATION and Ranked Pairs-RANDMANIPULATION are NP-hard.*

We remark that it is not clear if these problems are in NP, since the respective winner determination problem is not known to be polynomial-time solvable; in fact, for STV it is known to be NP-hard [Conitzer *et al.*, 2009].

For iterative rules one can also use randomness to break the intermediate ties. The manipulator’s goal is then to maximize the expected utility with respect to the resulting distribution. Generally speaking, this problem is different from RANDMANIPULATION: while the set of candidates that win with non-zero probability is the same in both settings, the probability distribution on these candidates can be different. Nevertheless, we can extend the easiness result for Plurality with Runoff to this model; however, it is not clear if this is the case for the hardness results of Corollary 1.

7 Conclusions and Future Work

We have determined the complexity of finding an optimal manipulation under the randomized tie-breaking rule for several prominent voting rules, namely, Maximin, Copeland $^\alpha$ for any rational $\alpha \in [0, 1]$, two variants of the Bucklin rule, Plurality with Runoff, STV, and Ranked Pairs. Together with the results of [Obraztsova *et al.*, 2011], this provides an essentially complete picture of the complexity of RANDMANIPULATION for commonly studied voting rules (Table 1).

| P | NP-hard |
|----------------------------|---------------------------|
| Scoring rules | Copeland, Thm. 2 |
| Maximin (restricted) | Maximin (general), Thm. 1 |
| simplified Bucklin, Thm. 3 | STV, Cor. 1 |
| classic Bucklin, Thm. 4 | Ranked Pairs, Cor. 1 |
| Plurality w/Runoff, Thm. 5 | |

Table 1: Complexity of RANDMANIPULATION for classic voting rules. The first two results in the left column are due to [Obraztsova *et al.*, 2011].

There is a number of open questions left by our work. For instance, it would be interesting to see whether the easiness results for coalitional manipulation under lexicographic tie-breaking proven by [Zuckerman *et al.*, 2009; Xia *et al.*, 2009] extend to randomized tie-breaking, or whether our algorithmic results hold under a more general definition of randomized tie-breaking, where different candidates may be selected with different probabilities; the latter question includes, in particular, the setting considered in the end of Section 6. Another promising research direction is designing approxima-

tion algorithms for the optimization version of RANDMANIPULATION; while the proof of our hardness result for Copeland can be strengthened to show that this problem does not admit a constant-factor approximation algorithm (we omit the proof due to space constraints; briefly, we modify our construction so that there are many zero-utility necessary winners), it is not clear if this is the case for Maximin, STV, or Ranked Pairs.

Acknowledgments This research was supported by National Research Foundation (Singapore) under grant 2009-08, by NTU SUG, and by the President of RF grant for leading scientific schools support NSh-5282.2010.1.

References

- [Bartholdi *et al.*, 1989] J. J. Bartholdi, C. A. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.
- [Conitzer *et al.*, 2009] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *IJCAI’09*, pp. 109–115, 2009.
- [Desmedt and Elkind, 2010] Y. Desmedt and E. Elkind. Equilibria of plurality voting with abstentions. In *ACM EC’10*, pp. 347–356, 2010.
- [Ephrati and Rosenschein, 1997] E. Ephrati and J. Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20(1–4):13–67, 1997.
- [Faliszewski *et al.*, 2008] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *AAMAS’08*, pp. 983–990, 2008.
- [Faliszewski and Procaccia, 2010] P. Faliszewski and A. Procaccia. AI’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.
- [Garey and Johnson, 1979] M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, 1979.
- [Gibbard, 1973] A. F. Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:597–601, 1973.
- [Gibbard, 1977] A. F. Gibbard. Manipulation of schemes that mix voting and chance. *Econometrica*, 45:665–681, 1977.
- [Hazon *et al.*, 2008] N. Hazon, Y. Aumann, S. Kraus, M. Wooldridge. Evaluation of election outcomes under uncertainty. In *AAMAS’08*, pp. 959–966, 2008.
- [McGarvey, 1953] D. C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.
- [Obraztsova *et al.*, 2011] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *AAMAS’11*, to appear.
- [Satterthwaite, 1975] M. A. Satterthwaite. Strategy-proofness and Arrow’s conditions: existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.
- [Xia *et al.*, 2009] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted manipulation under some common voting rules. In *IJCAI’09*, pp. 348–353, 2009.
- [Zuckerman *et al.*, 2009] M. Zuckerman, A. Procaccia, and J. Rosenschein. Algorithms for the coalitional manipulation problem. *Artificial Intelligence Journal*, 173(2):392–412, 2009.