

# Ties Matter: Complexity of Manipulation when Tie-breaking with a Random Vote

Haris Aziz, Serge Gaspers, Nicholas Mattei, Nina Narodytska, and Toby Walsh

NICTA and UNSW  
Sydney, Australia

{haris.aziz, serge.gaspers, nicholas.mattei, nina.narodytska, toby.walsh}@nicta.com.au

## Abstract

We study the impact on strategic voting of tie-breaking by means of considering the order of tied candidates within a random vote. We compare this to another non-deterministic tie-breaking rule where we simply choose candidate uniformly at random. In general, we demonstrate that there is no connection between the computational complexity of computing a manipulating vote with the two different types of tie-breaking. However, we prove that for some scoring rules, the computational complexity of computing a manipulation can increase from polynomial to NP-hard. We also discuss the relationship with the computational complexity of computing a manipulating vote when we ask for a candidate to be the unique winner, or to be among the set of co-winners.

## Introduction

Voting is a general mechanism for combining preferences in multi-agent systems. One problem with voting is that agents may act strategically, manipulating the result by declaring insincere preferences. Strategic voting, especially when there is only a single manipulating agent or a small manipulating coalition, requires the election to be close. It therefore should come as no surprise that how we break ties is of some importance. Most results assume, either that ties are broken in favour of the manipulators, or that ties are broken against them (see (Faliszewski and Procaccia 2010; Faliszewski, Hemaspaandra, and Hemaspaandra 2010) for a summary of the recent literature). However, to ensure neutrality (all candidates are treated the same) ties are often broken in practice using a non-deterministic mechanism like tossing a coin.

Recently Obraztsova, Elkind and Hazon (2011) have initiated a study of manipulation when a candidate is chosen uniformly at random from the set of co-winners. We refer to this model as *Random Candidate (RC)*. For instance, they prove that *all* scoring rules are polynomial to manipulate in such a situation by a simple greedy method. In this paper, we consider another common method to deal with ties: we select a vote, which in our setting is a strict total order over the candidates, uniformly at random from the profile of submitted votes and select the highest-ranked of the tied candidates from this vote.

Copyright © 2013, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

In this paper, we study the setting in which ties among co-winners are broken by selecting a random vote uniformly at random. We refer to the tie-breaking rule as *RV*. *RV* is used for the Schulze voting method (2003) and is therefore in use by a number of organizations including Debian and Wikimedia. Tideman has also proposed using it for his Ranked-Pairs method (1987). From a theoretical perspective the *RV* method is well-grounded as Tideman argues that such a tie-breaking rule is not influenced by the cloning of candidates (whilst choosing a candidate at random is). When *RV* is used to select a random candidate from the set of all candidates, it coincides with the well-studied Random Ballot or Random Dictator rule (Duggan 1996).

We show that the *RV* method of non-deterministic tie-breaking has quite different properties than *RC*. For instance, several common scoring rules become NP-hard to manipulate when we tie-break with a random vote.

## Formal Background

A *profile* is a set of  $n$  total orders (votes) over  $m$  candidates. A *voting correspondence* is a function mapping a profile onto a set of *co-winners*. A tie-breaking rule may then be used to return the unique winner. Let  $N(i, j)$  be the number of voters preferring  $i$  to  $j$ . We consider some of the most common voting correspondences. These were considered in recent work on tie-breaking with a random candidate (Obraztsova, Elkind, and Hazon 2011; Obraztsova and Elkind 2011).

**Scoring rules:**  $(w_1, \dots, w_m)$  is a vector of scores where the  $i$ th candidate in a vote scores  $w_i$ , then the co-winners are the candidates with highest total score over all the votes. *Plurality* has  $w_1 = 1, w_i = 0$  for  $i > 1$ , *veto* has  $w_m = 0, w_i = 1$  for  $i < m$ , *k-approval* has  $w_i = 1$  for  $i \leq k, 0$  otherwise, and *Borda* has  $w_i = m - i$ .

**Copeland:** The candidates with the highest Copeland score win. Let  $N(i, j)$  be the number of votes where candidate  $i$  is preferred to candidate  $j$ . The Copeland score of candidate  $i$  is  $\sum_{j \neq i} (N(i, j) > \frac{n}{2}) - (N(i, j) < \frac{n}{2})$ . In the second-order Copeland rule, if there is a tie, the winner is the candidate whose defeated competitors have the largest sum of Copeland scores.

**Plurality with runoff:** If one candidate has a majority, he wins. Otherwise everyone but the two candidates with the

most votes are eliminated and the plurality winner wins.

**Single Transferable Vote (STV):** This rule requires up to  $m - 1$  rounds. In each round, the candidate with the least number of voters ranking them first is eliminated until one of the remaining candidates has a majority.

**Bucklin:** The Bucklin winning round is the smallest value  $k$  such that the  $k$ -approval score of at least one candidate exceeds  $\lfloor n/2 \rfloor$ . The Bucklin score of a candidate is his  $k$ -approval score, where  $k$  is the Bucklin winning round. A co-winner is a candidate with the largest Bucklin score. The *simplified Bucklin* procedure is the same except all candidates with score exceeding  $\lfloor n/2 \rfloor$  are co-winners.

We consider the following manipulation problems. We are given  $n - 1$  votes, a preferred candidate  $p$  and a voting correspondence. For the *co-winner manipulation problem*, we wish to decide if we can cast the one remaining vote to make  $p$  a co-winner. For the *unique winner manipulation problem*, we wish to decide if we can cast the one remaining vote to make  $p$  the only winner. For the *non-deterministic manipulation problem*, we are also given a randomized tie-breaking rule (viz. random candidate or a random vote), a probability  $t$  and we wish to decide if we can cast the one remaining vote to make  $p$  the winner with probability at least  $t$ . Note that the unique winner manipulation problem is equivalent to the co-winner manipulation problem with tie-breaking against the manipulator. Similarly, the co-winner manipulation problem is equivalent to the unique winner manipulation problem with tie-breaking in favour of the manipulator.

## Random Vote vs. Random Candidate

There are a number of differences between tie-breaking via a random vote and by choosing a random candidate. Consider Borda voting where half the votes are  $a > b > c$  and the other half are  $c > b > a$ . All three candidates have the same Borda score. Tie-breaking with a random candidate results in  $a$ ,  $b$  or  $c$  being the overall winner with probability  $1/3$ . By contrast, tie-breaking with a random vote results in  $a$  or  $c$  being the overall winner with probability  $1/2$ . One argument in favour of tie-breaking with a random vote is that it discourages strategic voting. If you vote strategically and the result is tied, your vote may result in a worse outcome than voting sincerely. As we shall see, such tie-breaking can also change the computational complexity of computing a strategic vote. In general, tie-breaking with a random vote appears to make manipulation more intractable than tie-breaking with a random candidate. However, the computational complexities of the two problems are unrelated.

**Theorem 1.** *There exists a voting correspondence such that the non-deterministic manipulation problem is NP-complete when tie-breaking with a random vote, but polynomial when tie-breaking with a random candidate, and vice versa.*

**Proof:** Theorem 9 shows that the non-deterministic manipulation problem for Borda when tie-breaking with a random vote is NP-complete. On the other hand, Theorem 4.1 in (Obraztsova, Elkind, and Hazon 2011) proves that the non-deterministic manipulation problem for every scoring rule

(and hence for Borda) is polynomial when tie-breaking with a random candidate.

For the reverse direction, we use a reduction from the NP-complete 1in3SAT problem on positive clauses (Schaefer 1978) in which Boolean variables are represented by non-negative integers and each three literal clause has exactly one true literal. We represent the positive clause  $a \vee b \vee c$  by the vote  $-1 > a > b > c > -2 > \dots$ , and the truth assignment that sets  $x_i$  to true and  $y_j$  to false by  $-2 > x_1 > \dots > x_n > -1 > y_1 > \dots > y_m$ . Consider the following voting correspondence. The rule elects the plurality winner plus potentially one more candidate. If the votes are unanimous that candidate 0 is in last place, all votes but one represent 1in3SAT clauses, and the final vote represents a truth assignment that satisfies one in three of the literals of each clause represented by the other votes, then the rule also elects candidate 0. In all other cases, only the plurality winner is elected.

Consider how we can ensure candidate 0 wins when we tie-break with a random candidate. Suppose the fixed votes represent the clauses in a 1in3SAT problem and put 0 in last place. Then the manipulator can make candidate 0 win with probability 0.5 if and only if his vote has 0 in last place and it represents a satisfying 1in3SAT truth assignment. By comparison, suppose we tie-break with a random vote. There is no advantage for the manipulator to cast a manipulating vote that ensures the plurality winner and 0 are both in the set of winners (as tie-breaking will always favour the plurality winner). Hence, the manipulator simply votes for their preferred candidate to be the plurality winner.  $\square$

## Relation to Unique and Co-winner Problems

The computational complexity of the non-deterministic manipulation problem when tie-breaking with a random candidate can be related to that of both the co-winner and unique winner manipulation problems. First, if the unique winner manipulation problem is NP-complete then so is the corresponding non-deterministic manipulation problem when tie-breaking with a random candidate. To prove this, we simply observe that unique winner manipulation is equivalent to deciding if a candidate can win with probability 1 when tie-breaking at random between the co-winners. Second, Proposition 1 in (Obraztsova and Elkind 2011) shows that if co-winner manipulation is NP-complete then so is the corresponding non-deterministic manipulation problem when tie-breaking with a random candidate. As an aside, we note that the manipulation problem when tie-breaking with a random candidate is thus harder than either the co-winner or unique winner manipulation problems since it is computationally intractable in the case the co-winner manipulation problem is intractable and unique winner is easy, and vice versa.

Interestingly, the same relationships do not hold when tie-breaking with a random vote. The computational complexity of the co-winner or unique winner manipulation problem and the corresponding non-deterministic manipulation problem when tie-breaking with a random vote are unrelated.

**Theorem 2.** *There exists a voting correspondence such that both the co-winner and unique winner manipula-*

tion problems are NP-complete but the corresponding non-deterministic manipulation problem when tie-breaking with a random vote is polynomial, and vice versa.

**Proof:** Consider the following voting correspondence. Let  $a$  and  $b$  be the lexicographically smallest candidates. If every vote has  $a$  in first place and the STV winner among the other candidates is  $b$  then elect both  $a$  and  $b$ . Otherwise elect just  $a$ . Now both the co-winner and unique winner manipulation problems are NP-complete since they are equivalent to deciding if  $b$  can be made an STV winner in the election obtained by removing  $a$ , which is NP-complete (Bartholdi and Orlin 1991). However, the corresponding non-deterministic manipulation problem when tie-breaking with a random vote is polynomial since  $a$  wins with probability 1.

For the reverse, consider the voting correspondence that elects candidate  $b$  plus all candidates in last place. In addition, if every vote has candidate  $a$  in first place and the STV winner among the other candidates is  $b$  then we also elect  $a$ , otherwise we also elect all candidates in first place in any vote except for  $a$ . The co-winner manipulation problem is polynomial since we just need to put the preferred candidate in last place. Similarly, the unique winner manipulation problem is polynomial since there is never a unique winner. However, to decide if  $a$  wins with probability 1 when tie-breaking with random vote, we must put  $a$  in first place and decide if  $b$  can win the STV election in which  $a$  is eliminated. This is NP-complete (Bartholdi and Orlin 1991).  $\square$

On the other hand, if we add some weak assumptions, we can relate the computational complexity of the co-winner or the unique winner manipulation problems with that of the corresponding non-deterministic manipulation problem when tie-breaking with a random vote.

**Theorem 3.** *If the co-winner manipulation problem is NP-complete when the preferred candidate is in first place in at least one vote then the corresponding non-deterministic manipulation problem when tie-breaking with a random vote is NP-complete.*

*Similarly, if the unique winner manipulation problem is NP-complete when the preferred candidate is in last place in at least one vote then the corresponding non-deterministic manipulation problem when tie-breaking with a random vote is NP-complete.*

**Proof:** Suppose the preferred candidate is in first place in at least one vote. The preferred candidate can be made a co-winner if and only if the probability of winning when tie-breaking with a random vote is non-zero.

Suppose the preferred candidate is in last place in at least one vote. If there are multiple winners, the preferred candidate wins after tie-breaking with a random vote with probability less than 1. Hence, the preferred candidate can be made the unique winner if and only if the probability of winning when tie-breaking with a random vote is exactly 1.  $\square$

We can appeal to this result to show that tie-breaking with random vote inherits computational complexity from the corresponding co-winner or unique winner manipulation problems. For example, we can show that manipulation of STV and ranked pairs are NP-hard when tie-breaking with a random vote. This is similar to tie-breaking with a random

candidate where both rules have been shown to be NP-hard to manipulate (Obraztsova and Elkind 2011).

**Theorem 4.** *The non-deterministic manipulation problem for STV and ranked pairs when tie-breaking with a random vote is NP-complete.*

**Proof:** For STV, we note that the reduction used in the proof of Theorem 1 of (Bartholdi and Orlin 1991) puts the preferred candidate in first place in more than one vote. Hence, we can appeal to Theorem 3.

For ranked pairs, we adapt the reduction used in the proof of Theorem 4.2.2 of (Xia 2011) which shows that the unique winner manipulation problem for ranked pairs is NP-complete. We add two additional votes. The first ranks the preferred candidate first and the other candidates in some arbitrary order. The second is the reverse. This ensures that the preferred candidate is in last place in one vote, but leaves the outcome of ranked pairs unchanged.  $\square$

## Plurality and Veto

We focus next on scoring rules. This is a very general setting that was studied in (Obraztsova, Elkind, and Hazon 2011) for tie-breaking with a random candidate. We shall see that tie-breaking with a random vote either leaves the computational complexity unchanged or makes it more intractable compared to tie-breaking with a random candidate. We begin with two of the simplest possible scoring rules: plurality and veto. Not surprisingly, these rules are easy to manipulate even when we tie-break with a random vote.

**Theorem 5.** *The non-deterministic manipulation problem for plurality or veto when tie-breaking with a random vote is polynomial.*

**Proof:** For plurality, we simply vote for the candidate we wish to win. This is the best possible vote. For veto, we can try casting a veto for every candidate, with  $p$  in first place, and compute the probability with which our preferred candidates wins.  $\square$

## $k$ -Approval

We next turn to a slightly more complex scoring rule,  $k$ -approval. For bounded  $k$  and tie-breaking with a random vote, this is also polynomial to manipulate. However, with an unbounded  $k$ , the non-deterministic manipulation problem becomes NP-complete.

**Theorem 6.** *The non-deterministic manipulation problem for  $k$ -approval when tie-breaking with a random vote is polynomial for bounded  $k$ .*

**Proof:** The difficult aspect of constructing a manipulating vote is deciding who else besides the preferred candidate to make a co-winner since we are forced to approve  $k - 1$  candidates besides the preferred one. We can partition the candidates into the following sets:

$L$ : This is the set of necessary losers that, no matter what we do, will not be a co-winner and can be safely approved.

$U$ : This is the set of necessary winners that we can make a unique winner. We cannot approve any of these candidates.

$P$ : This is the set of possible co-winners. If we approve of any of these candidates, they become a co-winner.

	$ X  + 2$ Approvals	Not Approved
$i = 1 \dots  Y $	$N(y_i) \succ p \succ \overline{N(y_i)} \succ w$	$d_1 \dots d_j$
$f_1$	$w \succ x_1 \dots x_{ X } \succ d_1$	$p \succ d_2 \dots d_j$
$f_2$	$w \succ x_2 \dots x_{ X } \succ d_1 \succ p$	$x_1 \succ d_2 \dots d_j$
$\vdots$	$\vdots$	$\vdots$
$f_{ X +1}$	$w \succ x_1 \dots x_{ X -1} \succ d_1 \succ p$	$x_{ X } \succ d_2 \dots d_j$

Table 1: Non-manipulator votes in the proof of Theorem 7.

If  $|L| \geq k - 1$ , approve  $p$  and any subset of  $L$  of size  $k - 1$ . Otherwise, go through all  $\binom{|P|}{k-|L|-1}$  subsets of  $C$  of size  $k$  that include  $\{p\} \cup L$  and exclude  $U$ , and compute the probability that  $p$  wins in each case.  $\square$

**Theorem 7.** *The non-deterministic manipulation problem for  $k$ -approval when tie-breaking with a random vote is NP-complete for unbounded  $k$ .*

**Proof:** We give a reduction from the NP-complete HALL-SET problem (Gaspers et al. 2012): Given a bipartite graph  $G = (X, Y, E)$  and an integer  $z$ , does  $G$  have a Hall set of size  $z$ , i.e., a set  $S \subseteq X$  of size  $z$  such that  $|N(S)| < |S|$ , where  $N(S)$  denotes the neighbourhood of  $S$ ? Intuitively, given an instance of HALL SET, we create an instance of the non-deterministic manipulation problem where the manipulator must select a subset of candidates to place into the co-winner set. The votes in this instance are constructed in such a way that  $p$  will have a probability at least  $t$  of winning the election if and only if the subset selected to be in the co-winner set is a Hall set of size  $z$  in  $G$ .

Given an instance of HALL SET,  $(G = (X, Y, E), z)$  we create an instance of our problem whose candidate set is  $C = D \cup \{p, w\} \cup X$  with  $D = \{d_1, \dots, d_j\}$  and  $j = |X| - z + 1$ . Denote  $X = \{x_1, \dots, x_{|X|}\}$  and  $Y = \{y_1, \dots, y_{|Y|}\}$ . The candidate  $w$  will be a necessary co-winner and the candidates in  $D$  will be necessary losers. The set of voters is  $V = Y \cup \{f_1, \dots, f_{|X|+1}\}$ . We set  $k = |X| + 2$  and  $t = \frac{|Y| - z + 2}{|Y| + |X| + 2}$ . We then construct the votes as shown in Table 1. Here,  $\overline{N(y_i)}$  denotes the set  $X \setminus N(y_i)$ .

Let  $s(x)$  denote the score of candidate  $x$  in the non-manipulator votes. Denoting  $s(w) = T$ , we have:  $s(x_1) = \dots = s(x_n) = T - 1$ ;  $s(p) = T - 1$ ;  $s(d_1) = |X| + 1 < T - 1$ ; and  $s(d_2) = \dots = s(d_j) = 0$ .

We will show that there is a manipulative vote such that the probability that  $p$  wins is at least  $t$  if and only if there is a HALL SET of size  $z$  in the original instance. Suppose there is such a manipulative vote. Without loss of generality, assume  $p$  is ranked first in the manipulative vote, otherwise move  $p$  to the first place, which does not decrease the probability that  $p$  wins. The manipulator cannot approve  $w$  since  $w$  would become the unique winner. Besides  $p$ , the manipulator approves some subset of  $D$ , none of which will be in the co-winner set. Now the manipulator must approve of a set  $A \subseteq X$  of at least  $z$  additional candidates, all of which will be in the co-winner set. There are  $|Y| + |X| + 2$  votes. In the votes  $f_1, \dots, f_{|X|+1}$ , the candidate  $w$  is ranked before  $p$ , in the manipulative vote,  $p$  is ranked first, and the

Blocks	Votes	
$V_Y$	$i = 1..m$	$N(y_i) \succ p \succ \overline{N(y_i)} \succ W \succ D$
$V_B$	$i = 1..n$	$p \succ X \setminus \{x_i\} \succ x_i \succ W \succ D$ $x_1 \succ \dots \succ x_n \succ p \succ W \succ D$
$V_W$	$i = 1..m + n - 1$	$W \succ d_1 \succ p \succ X \succ D \setminus \{d_1\}$
	$i = 1..2$	$W \succ d_2 \succ p \succ X \succ D \setminus \{d_2\}$

Table 2: Non-manipulator votes in the proof of Theorem 8.

number of votes among  $Y$  where some co-winner is ranked before  $p$  is  $|N(A)|$ . Thus, the probability that  $p$  wins when tie-breaking with a random vote is  $\frac{1+|Y|-|N(A)|}{|X|+|Y|+2}$ . For this quantity to be at least  $t$ , it must be that  $|N(A)| \leq z - 1$ . Select any subset  $S$  of  $A$  of size  $z$ . This set  $S$  is a solution to the HALL SET instance since  $S \subset X$ ,  $|S| \leq z$ , and  $|N(S)| \leq |N(A)| < |S| = z$ .

On the other hand, suppose  $G$  has a Hall set  $S$  of size  $z$ . Then, build the manipulative vote by approving  $\{p\} \cup D \cup S$ . The probability that  $p$  wins is  $\frac{1+|Y|-|N(S)|}{|X|+|Y|+2} \geq t$ .  $\square$

We contrast this result with the non-deterministic manipulation problem when tie-breaking with a random vote, which is polynomial for all scoring rules (and thus  $k$ -approval) (Obraztsova, Elkind, and Hazan 2011). Hence tie-breaking with a random vote introduces computational complexity into this manipulation problem.

## Bucklin

Next we consider Bucklin voting. We recall that the co-winner and unique winner manipulation problems for Bucklin are polynomial (Xia et al. 2009). Similarly, the non-deterministic manipulation problem for (simplified) Bucklin when tie-breaking with a random candidate is polynomial (Obraztsova and Elkind 2011). We show here that tie-breaking with a random vote increases the computational complexity of computing a manipulation for Bucklin.

**Theorem 8.** *The non-deterministic manipulation problem for Bucklin and simplified Bucklin when tie-breaking with a random vote is NP-complete.*

**Proof:** (Sketch) We reduce again from HALL-SET. Let  $(G = (X, Y, E), z)$  be an instance for HALL-SET and denote  $X = \{x_1, \dots, x_n\}$  and  $Y = \{y_1, \dots, y_m\}$ .

We construct an election with  $3n + 1 - z$  candidates  $C = X \cup W \cup D \cup \{p\}$ . The set  $W = \{w_1, \dots, w_n\}$  represents dangerous candidates that the manipulator must prevent from being co-winners. The set  $D = \{d_1, \dots, d_{n-z}\}$  contains dummy candidates that are not in a co-winner set regardless of the manipulator's vote. We introduce an order over candidates  $C$ :  $p \succ x_1 \succ \dots \succ x_n \succ w_1 \succ \dots \succ w_n \succ d_1 \succ \dots \succ d_{n-z}$ . Given a vote  $(\dots \succ Z \succ \dots)$ ,  $Z \subset C$  we assume that candidates in  $Z$  are ranked with respect to this order.

We introduce three blocks of votes  $V = V_Y \cup V_B \cup V_W$ ,  $|V| = 2(m + n + 1)$ . Table 2 shows votes for each block. The first set of votes  $V_Y$ ,  $|V_Y| = m$ , encodes the neighborhood for each vertex  $y_i \in Y$ . For each  $y_i$  we introduce a vote  $N(y_i) \succ p \succ \overline{N(y_i)} \succ W \succ D$ , where  $N(y_i)$  ( $\overline{N(y_i)}$ ) con-

tains candidates that represent the neighborhood (the complement of  $N(y_i)$ ) of the vertex  $y_i$  in  $X$ . The second block of votes  $V_B$ ,  $|V_B| = n + 1$ , is a buffer block to make sure that none of the candidates in  $X \cup \{p\}$  gets more than half of the votes in round  $r$ ,  $r < n + 1$ . The third block  $V_W$ ,  $|V_W| = m + n + 1$ , makes sure that dangerous candidates in  $W$  get exactly  $|V|/2$  points in the  $n + 1$ th round.

We sketch the rest of the proof. We show that the probability of  $p$  winning the election is at least  $t = \frac{2(m+n+1)-(z-1)}{2(m+n+1)+1}$  if and only if  $G$  has a Hall set of size  $z$ . The high level structure of the proof is as follows. We make sure that  $p$  does not get more than  $|V|/2$  votes until the  $n + 1$ th round regardless of the manipulator's vote. At the  $(n + 1)$ th round, the scores of candidates in  $X \cup W \cup \{p\}$  is  $|V|/2$  and the manipulator decides the set of the co-winners. We will show that the following properties must hold for  $p$  to win. The manipulator has to give one point to the candidate  $p$  and to at least  $z$  candidates in  $X$ . This corresponds to selection of a set  $S$ . None of the candidates in  $W$  can be a co-winner. Finally, if the selected set  $S$  has less than  $z$  neighbors in  $X$  then  $p$  wins with probability  $t$ .  $\square$

### Borda

We finish our consideration of scoring rules with Borda. We recall that the co-winner and unique winner manipulation problems for Borda are polynomial (Bartholdi, Tovey, and Trick 1989). The generalization of the co-winner manipulation problem in which we have two manipulating agents casting strategic votes is NP-complete (Davies et al. 2011; Betzler, Niedermeier, and Woeginger 2011). We also recall that the non-deterministic manipulation problem when tie-breaking with a random candidate is polynomial (Obraztsova, Elkind, and Hazon 2011).

**Theorem 9.** *The non-deterministic manipulation problem for Borda when tie-breaking with a random vote is NP-complete.*

**Proof:** (Sketch) We reduce from MINSAT (Kohli, Krishnamurti, and Mirchandani 1994): Given a 2-SAT formula with  $n$  variables and  $m$  clauses and a positive integer  $k \leq m$ , is there a truth assignment that satisfies at most  $k$  clauses. We introduce  $n' = 2n + 4$  candidates:  $U = \{x_1, \bar{x}_1, \dots, x_n, \bar{x}_n, p, w_1, w_2, d\}$ . The first  $n$  pairs of candidates correspond to literals: the  $(2i - 1)$ th candidate encodes  $x_i$  and the  $(2i)$ th candidate encodes  $\bar{x}_i$ . Candidates  $w_1$  and  $w_2$  are dangerous candidates and  $d$  is a dummy candidate. The candidate  $p$  is the preferred candidate.

**The first part of the profile,  $P_1$ .** For each clause  $\ell_1 \vee \ell_2$  we introduce a vote  $\ell_1 \succ \ell_2 \succ p \succ w_1 \succ w_2 \succ \dots \succ d$ . This gives  $m$  votes in total. We denote these votes  $P_1$ . Let  $maxscore$  be the maximum score among candidates in  $P_1$  and  $s$  be a number,  $s > maxscore + n'$ .

**The second part of the profile,  $P_2$ .** We construct the second part in such a way that the total scores of candidates are as follows for some integer  $c$ :  $score(p) = s - (n' - 1) + c$ ,  $score(x_i) = score(\bar{x}_i) = s - 2i - 1 + c$ ,  $i = 1, \dots, n$ ,  $score(w_1) = s - 1 + c$ ,  $score(w_2) = s + c$  and  $score(d) < c$ .

Moreover, we ensure that either  $w_1$  or  $w_2$  is ranked ahead of  $p$  in all votes in  $P_2$ . We modify the construction of Mc-

Garvey's theorem (McGarvey 1953), which has been used elsewhere in the computational social choice literature (Xia, Conitzer, and Procaccia 2010; Betzler, and Dorn 2010). We show how to increase the score of a candidate by one more than the other candidates except for the last candidate whose score increases by one less while satisfying our additional restriction on the relative ordering of  $p, w_1$  and  $w_2$ .

To increase the score of  $x_i$  by one more than the score of the other non-dummy candidates we cast two votes:  $w_1 \succ x_i \succ d \succ p \succ x_1 \succ \dots \succ \bar{x}_n \succ w_2$  and  $w_2 \succ \bar{x}_n \succ \dots \succ x_1 \succ p \succ x_i \succ d \succ w_1$ . The score of  $x_i$  increases by  $n'$ , the score of the other candidates except  $d$  by  $n' - 1$ , and the score of  $d$  by  $n' - 2$ . Moreover,  $w_1$  or  $w_2$  are ranked ahead of  $p$ . Similar votes are used for  $\bar{x}_i, p, w_1$ , and  $w_2$ .

The construction makes sure that  $w_1, w_2$  are in the co-winner set  $\mathcal{W}$ . Hence, the probability of  $p$ 's victory does not depend on votes in  $P_2$ . Moreover, exactly one of the candidates in  $\{x_i, \bar{x}_i\}$  is in  $\mathcal{W}$ . This corresponds to an assignment of Boolean variables. If  $x_i \in \mathcal{W}$  then in all votes in  $P_1$  corresponding to clauses that are satisfied by  $x_i$  we have  $x_i$  ranked above  $p$ . Hence, this decreases the probability of  $p$ 's victory. The construction makes sure that the probability of  $p$  winning the election is at least  $t = \frac{m-k}{m+|P_2|}$  if and only if there exists a solution to the MINSAT instance.  $\square$

### Plurality with Runoff

We end our theoretical results with a final rule shown to be polynomial to manipulate when tie-breaking with a random candidate: plurality with runoff (Obraztsova and Elkind 2011). Manipulation remains polynomial when tie-breaking with a random vote. For simplicity, we assume an odd number of voters so that the only tie that can occur is to enter the runoff. Given a tie for the runoff, we chose a random vote, restrict it to the candidates eligible for the runoff and select the top two candidates in this vote.

**Theorem 10.** *The non-deterministic manipulation problem for plurality with runoff when tie-breaking with a random vote is polynomial.*

**Proof:** (Sketch) Observe that the probability that  $p$  wins equals the probability that it reaches the runoff together with some other candidate  $x \in C \setminus \{p\}$  which it defeats. There may be some instances where it is beneficial for the manipulator to cast a vote that ranks some candidate  $x$  ahead of  $p$  in order to make  $x$  enter the runoff stage. Our algorithm enumerates all candidates  $x \in C \setminus \{p\}$ . For each such  $x$ , the algorithm computes a manipulative vote maximizing the probability that  $p$  and  $x$  reach the runoff round under the restriction that  $p$  defeats  $x$  in the runoff. A simple case analysis shows that we need either cast a vote of the form  $p \succ x \succ \dots$  or  $x \succ p \succ \dots$ .  $\square$

### Empirical Comparison

Our experiments compare different tie-breaking methods in practice: random candidate (RC), random vote (RV) and manipulator favoured (MF). We use a single manipulator and Borda voting; since manipulation is NP-hard in this case with RV but polynomial with RC. Our setup is similar to (Walsh 2010): we varied the number of candidates

Inst.	$> 1$ Co-winner	RC = RV	Avg. Pr. Success			
			Pr(MF)	Pr(RC)	Pr(RV)	
IC	104,998	4,096 (3.9%)	4,046 (98.8%)	35.6%	33.3%	33.5%
IAC	104,998	3,441 (3.3%)	3,377 (98.1%)	34.2%	32.3%	32.4%
USP	104,998	2,491 (2.4%)	2,467 (99.0%)	32.6%	31.1%	31.2%
Urn	104,999	2,326 (2.2%)	2,308 (99.2%)	28.0%	26.6%	26.7%

Table 3: Experimental results: the first part shows how often the tie-breaking rule comes into play while the second half shows the average probability of successful manipulations under the different tie-breaking rules.

from  $m = 3, 2^2, 2^3, \dots, 2^7$ ; the number of voters from  $n = 1, 2^1, 2^1 + 1, 2^2, 2^2 + 1, \dots, 2^7, 2^7 + 1$  and tested 1,000 samples at each point. We used brute force search with a one hour time out to find the optimal vote for RV. We failed to find an optimal vote in 7 instances within the allotted time.

We used four statistical cultures to generate our data: the Impartial Culture (IC), where all  $m!$  votes are equally likely; the Impartial Anonymous Culture (IAC), where all distributions over the  $m!$  possible votes are equally likely; the Uniform Single Peaked Culture (USP) where all single-peaked profiles are equally likely; and an Urn model where replacement was set to  $m! - 1$  (once a vote was drawn, it had a 50% chance of being drawn on the next draw). For more detail see, for example, (Tideman and Plassmann 2012).

In our experiments neither RC or RV strictly dominates the other in terms of success probability; in most of our results, the probability curves cross each other one or more times. Table 3 shows how often we tie-break. Of the all the instances generated, few resulted in multiple co-winners. Of these instances, the optimal vote for manipulating under RC was the same as the optimal vote to manipulate under RV in 98.7% of the cases. Over all cases, the probability of success was about the same for RC and RV, which were both strictly dominated by tie-breaking in favour of the manipulator.

The conclusions that can be drawn from our empirical study are mixed. While they imply that tie-breaking comes into play in relatively few cases, they also suggest that in *some* cases, tie-breaking with a random vote is more challenging computationally. This might protect from manipulation some closely contested instances, where candidates are likely to be tied.

## Other Related Work

The importance of tie-breaking can be seen in the earliest literature on computational social choice. For instance, Bartholdi, Tovey and Trick (1989) prove that a single agent can manipulate the Copeland rule in polynomial time using the simple greedy algorithm mentioned earlier when ties are broken in favour of the manipulators, but with the second order tie-breaking rule (used in chess competitions, for example) manipulation becomes NP-hard.

Faliszewski, Hemaspaandra and Schnoor (2008) proved that for variants of Copeland voting, the tie-breaking rule impacts on the computational complexity of manipulation. For example, with weighted votes, if ties are scored 0, then it is NP-hard for a coalition to compute a manipulation that

tie-breaking rule	P	NP-hard	open
random vote	plurality, veto, $k$ -approval ( $k$ fixed), plurality with runoff	$k$ -approval ( $k$ unbounded), Borda, ranked pairs, STV (simplified) Bucklin	Copeland, maximin
random candidate	scoring rules (plurality, veto, $k$ -approval, Borda), plurality with runoff, (simplified) Bucklin	Copeland, (general) maximin, STV, ranked pairs	

Table 4: Comparison of the computational complexity of non-deterministic manipulation problems when tie-breaking with a random vote or a random candidate.

makes a given candidate the unique winner of the election, but polynomial if ties are given any other score.

Obraztsova, Elkind and Hazon (2011) consider a more general model of manipulation where the manipulator has utilities associated with any given candidate being elected, and they wish to maximize their expected utility under the randomized tie-breaking rule. This is equivalent to our model with a simple utility model: assign utility 1 to a preferred candidate and 0 to everyone else. For example, Obraztsova and Elkind (2011) provide a polynomial algorithm for constructing the optimal manipulation for the maximin rule in this setting. This implies that, in the model we consider, the manipulation problem for maximin when tie-breaking with a random candidate is polynomial.

## Conclusions

We have studied tie-breaking by means of considering a random vote and compared it with tie-breaking by choosing a candidate at random. In general, there is no connection between the computational complexity of computing a manipulating vote with the two different types of non-deterministic tie-breaking. However, for common rules like  $k$ -approval, Borda and Bucklin, the computational complexity increases from polynomial to NP-hard. For other rules like plurality, veto and plurality with runoff, it remains polynomial to compute a manipulating vote. For rules like STV and ranked pairs, where computing a manipulation is hard even without tie-breaking, the addition of tie-breaking unsurprisingly does not change the computational complexity of computing a manipulation. This work opens a number of interesting questions. Perhaps the most pressing problem is the computational complexity of manipulating the Copeland and Maximin rules (see Table 4). Both have been shown NP-hard to manipulate when tie-breaking with a random candidate.

## Acknowledgements

The authors are supported by the Australian Government’s Department of Broadband, Communications and the Digital Economy, the Australian Research Council and the Asian Ofce of Aerospace Research and Development through grants AOARD-104123 and 124056. Serge Gaspers acknowledges support from the Australian Research Council, grant DE120101761.

## References

- 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.
- Bartholdi, J.; Tovey, C.; and Trick, M. 1992. How hard is it to control an election. *Mathematical and Computer Modeling* 16(8-9):27–40.
- Betzler, N.; and Dorn B. 2010. Towards a dichotomy of finding possible winners in elections based on scoring rules. *Journal of Computer and System Sciences* 76(8):812–836.
- Betzler, N.; Niedermeier, R.; and Woeginger, G. 2011. Unweighted coalitional manipulation under the Borda rule is NP-hard. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. International Joint Conference on Artificial Intelligence.
- Davies, J.; Katsirelos, G.; Narodytska, N.; and Walsh, T. 2011. Complexity of and algorithms for Borda manipulation. In Burgard, W., and Roth, D., eds., *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press.
- Duggan, J. 1996. A geometric proof of Gibbard’s random dictatorship theorem. *Economic Theory* 7(2):365–369.
- Faliszewski, P., and Procaccia, A. 2010. AI’s war on manipulation: Are we winning? *AI Magazine* 31(4):53–64.
- Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. 2010. Using complexity to protect elections. *Communications of the ACM* 53(11):74–82.
- Faliszewski, P.; Hemaspaandra, E.; and Schnoor, H. 2008. Copeland voting: ties matter. In Padgham, L.; Parkes, D.; Müller, J.; and Parsons, S., eds., *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, 983–990.
- Gaspers, S.; Kim, E. J.; Ordyniak, S.; Saurabh, S.; and Szeider, S. 2012. Don’t be strict in local search! In Hoffmann, J., and Selman, B., eds., *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI 2012)*. AAAI Press.
- Kohli, R.; Krishnamurti, R.; and Mirchandani, P.: 1994. The Minimum Satisfiability Problem. *SIAM J. Discrete Math.* 7(2): 275–283.
- McGarvey, D. 1953. A theorem on the construction of voting paradoxes. *Econometrica* 21(4):608–610.
- Obraztsova, S., and Elkind, E. 2011. On the complexity of voting manipulation under randomized tie-breaking. In Walsh, T., ed., *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*, 319–324. IJCAI/AAAI.
- Obraztsova, S.; Elkind, E.; and Hazon, N. 2011. Ties matter: complexity of voting manipulation revisited. In Sonenberg, L.; Stone, P.; Tumer, K.; and Yolum, P., eds., *10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, 71–78.
- Schaefer, T. J. 1978. The complexity of satisfiability problems. In Lipton, R. J.; Burkhard, W. A.; Savitch, W. J.; Friedman, E. P.; and Aho, A. V., eds., *10th Annual ACM Symposium on Theory of Computing (STOC 1978)*, 216–226.
- Schulze, M. 2003. A new monotonic and clone-independent single-winner election method. *Voting matters* 17(1):9–19.
- Tideman, N., and Plassmann, F. 2012. Modeling the outcomes of vote-casting in actual elections. In Felsenthal, D., and Machover, M., eds., *Electoral Systems: Paradoxes, Assumptions, and Procedures*. Springer.
- Tideman, T. N. 1987. Independence of clones as a criterion for voting rules. *Social Choice and Welfare* 4(3):185–206.
- Walsh, T. 2010. An empirical study of the manipulability of single transferable voting. In *Proceedings of the 19th European Conference on Artificial Intelligence*, 257–262.
- Walsh, T., and Xia, L. 2012. Lot-based voting rules. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, 603–610.
- Xia, L.; Zuckerman, M.; Procaccia, A.; Conitzer, V.; and Rosenschein, J. 2009. Complexity of unweighted coalitional manipulation under some common voting rules. In Boutilier, C., ed., *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, 348–353. International Joint Conference on Artificial Intelligence.
- Xia, L.; Conitzer, V.; and Procaccia, A. 2010. A scheduling approach to coalitional manipulation. In Parkes, D.; Dellarocas, C.; and Tennenholtz, M., eds., *Proceedings 11th ACM Conference on Electronic Commerce (EC-2010)*, 275–284.
- Xia, L. 2011. *Computational Voting Theory: Game-Theoretic and Combinatorial Aspects*. Ph.D. Dissertation, Department of Computer Science, Duke University.