Did That Lost Ballot Box Cost Me a Seat? Computing Manipulations of STV Elections

Michelle Blom,[1] **Andrew Conway,**[2] **Peter J. Stuckey,**[3] **Vanessa J. Teague**[1]
[1]University of Melbourne, Australia
[2]Silicon Econometrics Pty. Ltd.
[3]Monash University, Australia

Abstract

Mistakes made by humans, or machines, commonly arise when managing ballots cast in an election. In the 2013 Australian Federal Election, for example, 1,370 West Australian Senate ballots were lost, eventually leading to a costly re-run of the election. Other mistakes include ballots that are misrecorded by electronic voting systems, voters that cast invalid ballots, or vote multiple times at different polling locations. We present a method for assessing whether such problems could have made a difference to the outcome of a Single Transferable Vote (STV) election – a complex system of preferential voting for multi-seat elections. It is used widely in Australia, in Ireland, and in a range of local government elections in the United Kingdom and United States.

1 Introduction

The Single Transferable Vote (STV) is a system of preferential voting for multi-seat elections. In an STV election, each ballot cast is a (potentially partial) ranking over a set of candidates. It is used in Australia to elect candidates to the upper houses of Parliament at federal and state levels, in upper and lower house elections in Ireland, and a range of local government elections in the United Kingdom and United States. The STV counting process is complex, with each ballot assigned a fractional weight (its *value*) that changes over time. Australian STV elections often involve over a hundred candidates, and many 'rounds' of counting in which candidates are elected or eliminated from consideration.

The complexity of STV has led to the development of software for running the count in Australian elections. Ballot scanning and digitisation methods are used to create electronic records of cast ballots. These electronic records are often made publicly available after the outcome of the election has been determined. An election outcome is the subset of candidates $\mathcal{C}$ who were awarded a seat (the winners $\mathcal{W} \subset \mathcal{C}$); the rest have missed out (the losers $\mathcal{L} = \mathcal{C} \setminus \mathcal{W}$).

We present, in this paper, methods that will allow any interested individual, or electoral commission, to answer the following questions about an STV election result.

- Can we find N ballots that, when *added* or *removed* to/from those cast, realises a change in outcome?
- Can we find N ballots that, when *replaced* with ballots expressing alternate rankings, realises a change in outcome?

The outcome of an STV election is changed if at least one candidate in $\mathcal{W}$ is replaced with a candidate in $\mathcal{L}$. Our methods attempt to answer these questions by finding *manipulations* of an STV election profile (the collection of electronic ballot records) that bring about a change in outcome.

We consider the following types of manipulations on an election profile: adding a ballot, with a given preference ordering over the available candidates; removing a ballot; or replacing the preference ordering on a ballot with a different candidate order. To answer the first of the above questions, we must determine if we can find (up to) N ballots that if added (or removed) from the current profile results in at least one current loser replacing a current winner. To answer the second question, we must determine if we can find (up to) N ballots that, if swapped with N different ballots, results in at least one current loser replacing a current winner.

The methods presented in this paper attempt to find the smallest possible outcome-changing manipulation of an election profile in three settings (addition-only; deletion-only; and shift- or replace-only). If the size of the discovered manipulation is $\leq N$ ballots, we can prove that certain errors or problems could have changed an election outcome.

If we can add less than, or equal to, 1,370 ballots to the profile in the 2013 West Australian Senate election, and change the result, we can prove that the loss of 1,370 ballots could have been outcome-changing. If we know that N voters cast invalid ballots in an election,[1] and can find an outcome-changing manipulation that adds $M \leq N$ ballots to the profile, we know that these ballots could have made a difference. If there were N voters who cast more than one ballot in an election, finding an outcome-changing manipulation in which $M \leq N$ ballots are removed indicates that multiple-voting could have influenced the outcome. If

[1]Invalid or informal ballots are excluded from consideration in Australian STV elections – they are not included in the counting process. An example of an informal ballot is one in which less than a required minimum number of candidates have been ranked.

we know that $N\%$ of ballots are likely to be misinterpreted by software systems during a count, finding an outcome-changing manipulation that replaces fewer than $N\%$ of ballots can prove whether this misinterpretation is problematic.

The methods we present are not optimal – there is no guarantee that the manipulation they discover is the 'smallest possible'. Consequently, even though we may not be able to find an outcome-changing manipulation of (up to) N ballots, it does not mean that one does not exist. The smallest number of ballot changes required to alter the outcome of an election is known as its *margin of victory* (MOV). Existing work presents an algorithm for computing a lower bound on the MOV of an STV election (Blom, Stuckey, and Teague 2019). This algorithm is able to find MOV lower bounds for elections of up to a dozen candidates and three to four seats. It cannot be applied to find non-trivial MOV lower bounds (i.e., lower bounds that are greater than 0) for large STV elections involving more than a hundred candidates and up to a dozen seats. The methods we present in this paper find an upper bound on the MOV of an STV election by finding actual manipulations that change the outcome.

We present two heuristic methods for finding candidate manipulations in the three settings described above. The first is denoted *Greedy*, a heuristic that repeatedly simulates shifts, additions, and removals, of varying numbers of ballots between targeted pairs of candidates, to each original loser, and from each original winner, respectively. *Greedy* looks for smaller and smaller shifts, additions, and removals, that realise an outcome change when simulated.

Our second method, *Guided-Greedy*, combines *Greedy* with the MOV lower bound finding algorithm *margin-stv* by Blom, Stuckey, and Teague (2019). We apply *margin-stv* to the tail of an STV election, containing up to 10 or so candidates. The resulting manipulations, since they are based only on a limited view of the election, will typically not change the election outcome when simulated. Applying *Greedy* on top of these manipulations indicates how many additional ballot changes are required to actually realise a change.

Our manipulation finding methods are demonstrated on the 2016 and 2019 Australian Senate (upper house of the Federal Parliament) elections. Background material on STV elections, and related work, is presented in Sections 2 and 3. Our *Greedy* and *Guided-Greedy* heuristics are outlined in Sections 4 and 5, and demonstrated in Section 6.

2 Preliminaries

The outcome of an STV election is a sequence of candidate eliminations and elections. Ballots are transferred between candidates throughout the tallying process, described below, with candidates required to achieve a certain number of votes, called a *quota*, before being elected to a seat.

Definition 1 (STV Election) *An STV election $\mathcal{E}$ is a tuple $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$ where $\mathcal{C}$ is a set of candidates, $\mathcal{B}$ the multi-set of ballots cast, Q the election quota (the number of votes a candidate must attain to be elected to a seat – the Droop quota – Eqn 1), and N the number of seats to be filled.*

$$Q = \left\lfloor \frac{|\mathcal{B}|}{N+1} \right\rfloor + 1 \tag{1}$$

Recall that each ballot in an STV election is a (potentially) partial ranking over a set of candidates. For example, in an election with candidates c_1, c_2, c_3, and c_4, the ballot [c_2, c_1, c_4] expresses a first preference for candidate c_2, a second for c_1, and a third for c_4. We refer the reader to the work of Blom, Stuckey, and Teague (2019) for pseudocode of the STV counting process, and a running example.

The tallying of an STV election proceeds in rounds. In each round, a candidate is either eliminated or elected to a seat. At the start of round i, the set of candidates who remain *standing* is the set of candidates who have not yet been eliminated or elected, denoted $\mathcal{S}_i$. Prior to the first round of counting, $i = 1$, each ballot $b \in \mathcal{B}$ is awarded to its first ranked candidate and is assigned a value of 1, $v_{i=1}(b) = 1$. The first ranked candidate of ballot [c_2, c_1, c_4], for example, is c_2. The sum of the values of the ballots sitting in a candidates pile forms their *tally*. In each round, ballots will move from the tally pile of one candidate to that of others. The value of these ballots—the extent to which they contribute to a candidate's tally—will change over time.

Definition 2 (Tally $t_i(c)$) *The tally of a candidate $c \in \mathcal{C}$ in round i is the sum of the values of the ballots in c's tally pile. These are the ballots for which c is ranked first among the set of candidates still standing, S_i. Let $\mathcal{B}_{i,c}$ denote the subset of ballots sitting in c's tally pile at the start of round i.*

$$t_i(c) = \sum_{b \in \mathcal{B}_{i,c}} v_i(b) \tag{2}$$

Candidates whose tallies exceed (or reach) a *quota* (Eqn 1) are elected to a seat. As each candidate is elected, their *surplus* (the number of votes by which their tally exceeds the quota) is computed, and a subset of their ballots (with a combined value equal to the surplus) is distributed to their next preferred candidate. In the ballot [c_2, c_1, c_4], the next preferred candidate after c_2 is c_1. If no candidate has amassed a quota's worth of votes, the candidate with the smallest tally is eliminated. Upon elimination, all ballots sitting in the candidate's pile are distributed to the next candidate in their ranking who is still standing (i.e., who have not yet been eliminated or elected to a seat) at their current value.

If the number of candidates still standing (i.e., who have not yet been eliminated or elected) equals the number of seats left to be filled, counting stops and these remaining candidates are elected. This is the one instance in which a candidate can be elected to a seat without achieving a quota.

An election order π defines the sequence of elections and eliminations that arise as during the STV counting process.

Definition 3 (Election Order π) *Given an STV election $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, we represent the outcome of the election as an election order π—a sequence of tuples (c, a) where $c \in \mathcal{C}$ and $a \in \{0, 1\}$. The tuple $(c, 1)$ indicates that candidate c is elected to a seat, and $(c, 0)$ that c is eliminated.*

The order $\pi = [(c_1, 1), (c_3, 0), (c_2, 0), (c_4, 1)]$, for example, indicates that candidate c_1 is elected to a seat in the first round of counting, followed by the elimination of candidates c_3 and c_2, and the election of c_4. We denote the subset of

candidates who are elected to a seat in order π as $\mathcal{W}_\pi \subset \mathcal{C}$, and the subset of candidates who are eliminated as $\mathcal{L}_\pi \subset \mathcal{C}$.

Our *Greedy* and *Guided-Greedy* heuristics operate by finding candidate manipulations of a given election $\mathcal{E}$, and simulating these manipulations to determine if they result in a change in outcome. If the original outcome of the election is the order π, we say that a manipulation changes the outcome if, when simulated, a new order is produced, π', for which $\mathcal{W}_{\pi'} \neq \mathcal{W}_\pi$. We define a manipulation as follows.

Definition 4 (Manipulation $\mathcal{M}$) *A manipulation for an election $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$ is a tuple $\mathcal{M} = (\mathcal{B}^+, \mathcal{B}^-)$, where: $\mathcal{B}^+$ denotes a multiset of ballots to add to $\mathcal{B}$; and $\mathcal{B}^-$ a multiset of ballots to remove from $\mathcal{B}$. The result of applying $\mathcal{M}$ to an election $\mathcal{E}$ is a modified election profile $\mathcal{E}' = (\mathcal{C}, \hat{\mathcal{B}}, \hat{Q}, N)$, where: $\hat{\mathcal{B}}$ is the result of removing each ballot in $\mathcal{B}^-$ from $\mathcal{B}$, and then adding each ballot in $\mathcal{B}^+$ to $\mathcal{B}$; and $\hat{Q}$ is quota of the new election $\mathcal{E}'$, computed using the revised number of cast ballots $|\hat{\mathcal{B}}|$, as per Eqn 1.*

In a manipulation generated in the shift-only setting (ballots can only be replaced), $|\mathcal{B}^+| \equiv |\mathcal{B}^-|$. In the add-only setting (ballots can only added), $\mathcal{B}^- \equiv \emptyset$, while in the delete-only setting (ballots can only be removed), $\mathcal{B}^+ \equiv \emptyset$.

Given a candidate manipulation $\mathcal{M}$, *Greedy* and *Guided-Greedy* use a custom STV simulator designed to reflect the precise STV variant used in a given case study. The heuristics refer to this simulator, denoted SIM-STV, as a black box.

2.1 SIM-STV: A Black Box STV Simulator

Several STV variants exist, differing in the way that surpluses are distributed (Weeks 2011). Consider a candidate with 100 ballots sitting in their tally pile, each with a value of 1, and a surplus of 40 votes. The Inclusive Gregory Method, arguably one of the simplest STV variants, redistributes all 100 ballots, each with an assigned 'transfer value' of 0.4 (each ballot is worth 0.4 votes), to their next highest ranked candidate that is 'still standing' (has not yet been elected or eliminated). Across all STV variants, ballots are assigned a transfer value when distributed as part of a surplus.

Unlike IRV for single seat elections where breaking ties and vote formality are the only real ambiguities in the idealized representation, there are a variety of details that different jurisdictions implementing STV have to make explicit rules on (Weeks 2011). The system used for the Senate in Australian Federal elections (Senate 2019) is quite close to the description above, but has some idiosyncrasies such as:

- When ballots are distributed from one candidate's tally pile to another, the total value of those ballots is rounded down to the nearest integer. This practice causes a number of votes to be lost over the course of the tallying process.
- When a candidate is elected, and their surplus distributed, all distributed ballots are assigned the same transfer value, regardless of prior transfer values assigned to each ballot.
- When a candidate is eliminated, their ballots with different transfer values are distributed in different distributions, each with their own rounding down.
- There is a gratuitous subsection 13(A) that allows simultaneous elimination of a couple of low candidates. This has been used by the Australian Electoral Commission (AEC) in some elections but not others. It was written with the aim of never changing the results, but can actually make changes due to effects caused by rounding.

All the manipulations we generate in our experiments are validated on the full federal rules using SIM-STV (Conway 2019), and ballot data published by the AEC, standardized at https://vote.andrewconway.org/.

3 Related Work

The size of the smallest manipulation of an election that changes its outcome is known as its margin of victory (MOV). There is much work on computing the MOV in the one-seat variant of STV, Instant Runoff Voting (IRV) (Magrino et al. 2011; Cary 2011; Sarwate, Checkoway, and Shacham 2013; Blom et al. 2016), but comparatively little for multi-seat STV. Blom, Stuckey, and Teague (2019) present an algorithm that is able to compute a lower bound on the MOV of small STV elections with less than a dozen candidates and four seats. The STV elections held to elect senators in Australian Federal elections can involve more than a hundred candidates and up to 12 seats. We utilise their *margin-stv* algorithm, in our *Guided-Greedy* heuristic, to generate suggestions for how to manipulate an election.

The *margin-stv* method uses branch-and-bound to explore a tree of possible alternate election outcomes, in which at least one original losing candidate is elected to a seat (Blom, Stuckey, and Teague 2019). Each node in this tree is a partial election order π', with complete election orders forming the leaves of the tree. A mixed-integer program (MIP) is defined to compute a lower bound on the smallest manipulation required to realise an election outcome that starts with π'. These lower bounds are used to prune sections of the tree from consideration. To compute an exact MOV, *margin-stv* would need to solve a mixed-integer non-linear program (MINLP) at each node, the scale of which is beyond modern MINLP solvers. Consequently, *margin-stv* computes a lower bound on the MOV via a linear relaxation of the problem.

Computing a non-trivial lower bound L on the MOV of an STV election is one way of answering the question of whether certain problems could have influenced an election outcome. If the problem involves less than L ballots, we know that it could not have changed the outcome. There are no methods available for computing such lower bounds for large STV elections. We can, as an alternative, provide evidence that a problem of size N could have changed the outcome, by finding an outcome-changing manipulation of size N or smaller. This is the task we focus on in this paper.

4 A Greedy Manipulation Finder

Figure 1 presents a greedy heuristic for finding candidate manipulations of an STV election $\mathcal{E}$ in the shift-only setting.

Consider an election $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$ with an outcome π. *Greedy* steps through each elimination and election that occurs in π. When a candidate c is elected to a seat (Step 4), we consider each original losing candidate $c' \in \mathcal{S} \setminus \mathcal{W}_\pi$ that

has not yet been eliminated. $\mathcal{S}$ denotes the set of candidates still standing at this stage of the count. Steps 6 and 7 find, and refine, a manipulation designed to shift enough ballots currently sitting in c's tally pile to that of c' so that c' has a higher tally than c. The difference in tallies between c and c' at this stage in the count is equal to $\Delta = t_{\mathcal{S}}(c) - t_{\mathcal{S}}(c')$.

We need to shift $\lceil(\Delta/2) + 1\rceil$ votes from c to c'. The MP procedure looks at the subset of ballots cast, $\mathcal{B}$, that are currently residing in c's tally pile, and sorts them in decreasing order of their current value at the stage in the count where $\mathcal{S}$ remain standing. The result is the set $\mathcal{B}_{c,\mathcal{S}}$ (Step 17). Replacing $\lceil\Delta/2\rceil$ ballots in $\mathcal{B}_{c,\mathcal{S}}$ with a ballot that preferences c' first will not necessarily alter the original election outcome. Each ballot in $\mathcal{B}_{c,\mathcal{S}}$ represents a proportion of a vote (between 0 and 1) in c's tally. *Greedy* chooses $\lceil t_{\mathcal{S}}(c') - t_{\mathcal{S}}(c) + 1\rceil$ as a likely upper bound on the number of ballot shifts required to elect c' in place of c.

We label this upper bound $\Delta_{c,c'}$. MP first considers replacing the rankings on the $\Delta_{c,c'}$ highest valued ballots in $\mathcal{B}_{c,\mathcal{S}}$, with a ranking that preferences c' first. This produces a new multiset of ballots, $\hat{\mathcal{B}}$ (Step 20). We simulate the election $\mathcal{E}$ with $\mathcal{B}$ replaced by $\hat{\mathcal{B}}$ (Step 21). If we are able to elect a previous loser (c' or otherwise), we update a running 'best found manipulation' (Step 23). Each time a valid manipulation is found, we reduce the number of ballots we shift by δ (Step 27), where δ is initially set to $\lceil\Delta_{c,c'}/2\rceil$ (Step 26). When we are unable to find a valid manipulation by shifting Δ ballots, we increase the number of ballots to be shifted. MP performs a binary search for the smallest number of ballot shifts required to alter the election outcome.

Note that *Greedy* maintains a record of the smallest successful manipulation found thus far (Steps 9 and 14). The size of this manipulation, m, is used to restrict the candidate manipulations considered by MP (Step 19).

4.1 Addition-Only

Where ballots can only be added to an election profile $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, with outcome π, the algorithm of Figure 1 is altered as follows. For every original loser $c \in \mathcal{L}_\pi$, we add a number of ballots Δ in which c is ranked first to $\mathcal{B}$, forming a new profile $\mathcal{E}' = (\mathcal{C}, \hat{\mathcal{B}}, \hat{Q}, N)$. Adding ballots to the profile increases the quota, $\hat{Q} > Q$. We initialise Δ to a suitably high value, and perform the binary search method of Steps 25 to 32 to find the smallest number of ballots Δ that, when added to $\mathcal{B}$, results in an outcome change upon simulation.

4.2 Deletion-Only

Where ballots can only be removed from an election profile $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, with outcome π, we consider each original winner $c \in \mathcal{W}_\pi$. When candidate c is elected to a seat, candidates $\mathcal{S}$ remain standing and $\mathcal{B}_{c,\mathcal{S}}$ denotes the multiset of ballots sitting in their tally pile, sorted in decreasing order of their current value. We remove a number of ballots Δ from $\mathcal{B}_{c,\mathcal{S}}$, forming a new election profile $\mathcal{E}' = (\mathcal{C}, \hat{\mathcal{B}}, \hat{Q}, N)$. Removing ballots from the profile reduces the quota, $\hat{Q} < Q$. We initialise Δ to the total number of ballots in their pile, and perform the binary search method

proc GREEDY($\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, π, $\mathcal{W}_\pi$)
1 $(m, \mathcal{M}) \leftarrow (\infty, \emptyset)$
2 $\mathcal{S} \leftarrow \mathcal{C}$
3 **for each** (c, a) **in** π **do**
4 **if** $a \equiv 1$ **then**
▷ Find ballot shift from c's current tally pile to that of $c' \in \mathcal{S}$, $c' \notin \mathcal{W}_\pi$, to give c' a seat.
5 **for each** $c' \in \mathcal{S} \setminus \mathcal{W}_\pi$ **do**
6 $\Delta_{c,c'} \leftarrow \lceil t_{\mathcal{S}}(c) - t_{\mathcal{S}}(c') + 1\rceil$
7 $(\Delta', \mathcal{M}') \leftarrow$ MP($\Delta_{c,c'}$, c, c', $\mathcal{S}$, $\mathcal{E}$, $\mathcal{W}_\pi$, m)
8 **if** $\Delta' \neq \infty$ **and** $\Delta' < m$ **then**
9 $(m, \mathcal{M}) \leftarrow (\Delta', \mathcal{M}')$
else
▷ Find ballot shift from some $c' \in \mathcal{W}_\pi$, $c' \in \mathcal{S}$, to c so that c' is eliminated in place of c.
10 **for each** $c' \in \mathcal{W}_\pi \bigcap \mathcal{S}$ **do**
11 $\Delta_{c',c} \leftarrow \lceil t_{\mathcal{S}}(c') - t_{\mathcal{S}}(c) + 1\rceil$
12 $(\Delta', \mathcal{M}') \leftarrow$ MP($\Delta_{c',c}$, c', c, $\mathcal{S}$, $\mathcal{E}$, $\mathcal{W}_\pi$, m)
13 **if** $\Delta' \neq \infty$ **and** $\Delta' < m$ **then**
14 $(m, \mathcal{M}) \leftarrow (\Delta', \mathcal{M}')$
15 $\mathcal{S} \leftarrow \mathcal{S} \setminus \{c\}$
16 **return** $(m, \mathcal{M})$

proc MP(Δ, c, c', $\mathcal{S}$, $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, $\mathcal{W}_\pi$, m_{best})
17 $\mathcal{B}_{c,\mathcal{S}} \leftarrow$ Ballots in c's tally pile in the round where candidates $\mathcal{S}$ remain, sorted by decreasing value.
18 $(m, \mathcal{M}) \leftarrow (\infty, \emptyset)$
19 $\Delta' \leftarrow \min(\Delta, m_{best})$
20 $\mathcal{M}', \hat{\mathcal{B}} \leftarrow$ Manipulation in which each of the first Δ' ballots in $\mathcal{B}_{c,\mathcal{S}}$ is replaced in $\mathcal{B}$ with with $[c']$.
▷ Simulate $\mathcal{E}$ with the new ballot set $\hat{\mathcal{B}}$.
21 $\pi' \leftarrow$ **SIM-STV**($\mathcal{E} = (\mathcal{C}, \hat{\mathcal{B}}, Q, N)$)
22 **if** $\mathcal{W}_{\pi'} \neq \mathcal{W}_\pi$ **then**
▷ We have found a valid manipulation.
23 $(m, \mathcal{M}) \leftarrow (\Delta', \mathcal{M}')$
24 $\delta \leftarrow \Delta'$
25 **repeat**
26 $\delta \leftarrow \lceil \frac{\delta}{2} \rceil$
27 $\Delta' \leftarrow \Delta' - \delta$
28 $\pi' \leftarrow$ Perform Steps 20-21 with new Δ'.
29 **if** $\mathcal{W}_{\pi'} \neq \mathcal{W}_\pi$ **then**
▷ We have found a valid manipulation.
30 $(m, \mathcal{M}) \leftarrow (\Delta', \mathcal{M}')$
else
31 $\Delta' \leftarrow \Delta + \delta$
32 **until** $\delta \equiv 1$
33 **return** $(m, \mathcal{M})$

Figure 1: *Greedy* heuristic for finding a candidate manipulation $\mathcal{M}$ of a STV election $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, with an original outcome π and winning candidates $\mathcal{W}_\pi \subset \mathcal{C}$.

of Steps 25 to 32 in Figure 1 to find the smallest number of ballots Δ that, when removed, results in an outcome change.

4.3 Electing a Desired Candidate

In the shift- and addition-only settings, we can apply a variation of *Greedy* to find manipulations in which a desired candidate $c' \in \mathcal{L}_\pi$ is elected to a seat. In the addition-only context, rather than consider *every* original losing candidate $c \in \mathcal{L}_\pi$, we consider only the candidate of interest $c' \in \mathcal{L}_\pi$. In the shift-only setting, Steps 5 to 9 in Figure 1 are altered to consider only a shift of ballots from the winning candidate c to our candidate, c'. Steps 10 to 14 are only performed when we are processing the elimination of c', $(c', 0)$, as we step through the original outcome π in the for loop of Step 3.

5 Guided-Greedy Manipulation

Guided-Greedy uses the *margin-stv* MOV lower bound calculation of Blom, Stuckey, and Teague (2019). Note that this uses a simpler model of STV than that actually used in Australian Federal Senate elections, but since any resulting manipulation is checked using SIM-STV the results are valid. *Guided-Greedy*, outlined in Figure 2, involves three phases.

Phase 1 (Steps 2 and 3): Simulate the original election $\mathcal{E}$ until K candidates remain. The result is a smaller K-candidate election, $\mathcal{E}_K$, with fractional-valued ballots, $\hat{\mathcal{B}}$. We apply *margin-stv* to $\mathcal{E}_K$ to compute a manipulation set $\mathbf{M}$.

Phase 2 (Steps 4 to 11): For each manipulation in $\mathbf{M}$, we apply the manipulation to $\mathcal{E}$, forming a manipulated profile, $\mathcal{E}'$. If, upon simulation of $\mathcal{E}'$, the election outcome has not changed, we apply *Greedy* to $\mathcal{E}'$ to determine how many additional ballot shifts/additions/removals are required.

Phase 3 (Step 12): The manipulation requiring the smallest number of ballot shifts/additions/removals is returned.

To find smaller manipulations, we apply varying proportions of the candidate manipulations found in Phase 1 (from 100% down to 0.5%) to $\mathcal{E}$. In some instances, (i) the manipulations in $\mathbf{M}$ make more ballot changes than necessary due to their limited view of the election (especially in the add/delete only setting), and (ii) some redundant ballot changes are made when taking the union of the candidate manipulation and manipulation suggested by *Greedy*.

When applying $N\%$ of a manipulation $\mathcal{M} = (\mathcal{B}^+, \mathcal{B}^-)$ to an election, we need to decide which ballots in $\mathcal{B}^+$ to add and which ballots in $\mathcal{B}^-$ to remove. Each manipulation is designed to elect a certain loser $l \in \mathcal{L}_\pi$ in place of a certain winner $w \in \mathcal{W}_\pi$. Experimentation has shown that it is best to sort $\mathcal{B}^-$ in decreasing order of their value at the point when w is elected. We then select the first $N\%$ of these ballots to remove. Sorting $\mathcal{B}^+$ has not proved beneficial in our experiments – we simply add the first $N\%$ of ballots in $\mathcal{B}^+$.

If we are interested in realising the election of a specific candidate $l \in \mathcal{L}_\pi$, we run *margin-stv* in a setting where all alternate outcomes in which l is not awarded a seat are ignored. All manipulations found by *margin-stv* award l with a seat. We then apply *Greedy* as described in Section 4.3.

Greedy is restricted to shifting ballots between two candidates, and adding/removing ballots to/from a single candidate. As *Guided-Greedy* starts from a manipulation found by *margin-stv*, it is able to form manipulations in which ballots may shift between multiple pairs of candidates, or be added/removed to/from multiple candidates.

proc GUIDEDGREEDY($\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, π, $\mathcal{W}_\pi$, K)
1 $(m_b, \mathcal{M}_b) \leftarrow (\infty, \emptyset)$
2 $\mathcal{E}_K, \pi_K, \mathcal{W}_{\pi_K} \leftarrow$ Simulate $\mathcal{E}$ until K candidates remain, the set $\mathcal{S}$, producing a new STV election $\mathcal{E}_K = (\mathcal{S}, \hat{\mathcal{B}}, Q, N_K)$
3 $\mathbf{M} \leftarrow$ Apply *margin-stv* to $\mathcal{E}_K$, producing a set of candidate manipulations $\mathbf{M}$. Each $(m, \mathcal{M}) \in \mathbf{M}$ is designed to realise an alternate election ending π'_K in which $\mathcal{W}_{\pi_K} \neq \mathcal{W}_{\pi'_K}$.
4 **for each** $(m, \mathcal{M}) \in \mathbf{M}$ **do**
▷ Apply candidate manipulation to $\mathcal{E}$ to form a new election profile, as per Def. 4.
5 $\mathcal{E}' = (\mathcal{C}, \hat{\mathcal{B}}, \hat{Q}, N) \leftarrow$ Apply $\mathcal{M}$ to $\mathcal{E}$
6 $\pi' \leftarrow$ **SIM-STV**($\mathcal{E}'$)
7 **if** $\mathcal{W}_{\pi'} \neq \mathcal{W}_\pi$ **then**
8 Replace $(m_b, \mathcal{M}_b)$ with $(m, \mathcal{M})$ if $m < m_b$.
else
9 $(m_g, \mathcal{M}_g) \leftarrow$ GREEDY($\mathcal{E}'$, π', $\mathcal{W}_{\pi'}$)
10 **if** $m_g + m < m_b$ **then**
11 $(m_b, \mathcal{M}_b) \leftarrow (m_g + m, \mathcal{M}_g \cup \mathcal{M})$.
12 **return** $(m_b, \mathcal{M}_b)$

Figure 2: *Guided-Greedy* heuristic for finding a manipulation $\mathcal{M}$ of a STV election $\mathcal{E} = (\mathcal{C}, \mathcal{B}, Q, N)$, whose unmanipulated outcome is a sequence of candidate elections and eliminations π, with candidates $\mathcal{W}_\pi \subset \mathcal{C}$ awarded a seat.

6 Case Studies

We consider the 2016 and 2019 Australian Federal Senate elections. Each state and territory holds a separate STV election to elect candidates to a number of seats reserved for the region. In 2016, a double dissolution election was held with 12 seats available in each state, and 2 in each territory. In 2019, 6 seats were available in each state, and 2 in each territory. Table 1 records the smallest manipulation discovered via *Greedy* or *Guided-Greedy* in each state and territory STV election held during the 2016 and 2019 Federal elections.

We have run *Guided-Greedy* in the setting where 0.5%, 1%, 5%, 15%, 25%, 50%, and 100% of the candidate manipulations found by *margin-stv* are applied, and have recorded the best overall manipulation discovered across those settings in Table 1. All experiments have been conducted on a machine with an Intel Xeon Platinum 8176 chip (2.1GHz), and 1TB of RAM. Both heuristics can be run in an overnight period, even in the largest of the considered elections.

Table 1 highlights how fragile the election results for the Australian Senate can be. In Tasmania 2016, the number of ballot changes required to alter the outcome is less than the number of votes lost by rounding. If the election rules were altered to avoid rounding down, the result could have

Contest	N	$\|C\|$	Enrollment	Formal Votes	Informal Votes	Voter No Shows	Lost by Rounding	Best Manipulation		
								AO	SO	DO
2019 Australian Federal Senate Election										
ACT	2	17	295,847	270,231	6,420	19,196	22	**32,760**	12,938	21,384*
SA	6	42	1,210,817	1,094,823	39,733	76,261	147	**40,439**	**27,362**	**86,549**
WA	6	67	1,646,262	1,446,623	50,909	148,730	232	**86,670**	**39,991**	**88,520**
VIC	6	82	4,184,076	3,739,443	156,793	287,840	255	**154,510**	93,164	**247,253**
NSW	6	105	5,294,468	4,695,326	210,146	388,996	245	153,560	**94,060**	**221,832**
QLD	6	83	3,262,898	2,901,464	97,908	263,526	244	92,761	**39,338**	**66,647**
TAS	6	44	385,816	351,988	13,284	20,544	92	22,099	**13,915**	26,108*
2016 Australian Federal Senate Election										
ACT	2	22	282,045	254,767	5,754	21,524	31	43,316*	18,836	31,814*
SA	12	64	1,183,004	1,061,165	36,545	85,294	449	4,532	1,772	**5,899**
WA	12	79	1,577,215	1,366,182	47,371	163,662	524	**19174**	**10,283**	**20,942**
VIC	12	116	3,963,992	3,500,237	153,499	310,256	782	**20429**	**13,068**	38,869*
NSW	12	151	5,084,274	4,492,197	213,073	379,004	770	15577	**13,266**	**26828**
QLD	12	122	3,074,422	2,723,166	95,831	255,425	665	20,520	**9,640**	18298*
TAS	12	58	373,470	339,159	12,221	22,090	285	234	71	**170**

Table 1: Best manipulations found for Australian Federal Senate (2016,2019) STV elections, recording the number of: enrolled voters; valid (formal) and invalid (informal) ballots cast; enrolled voters who did not vote (no shows); votes lost due to rounding; and ballot shifts (SO), additions (AO), or removals (DO) required to realise an outcome-change. Instances for which *Guided-Greedy* results in the smallest manipulation are in bold. An asterisk indicates that *Greedy* found the smallest manipulation.

Candidate	Greedy		Guided-Greedy	
	SO	AO	SO	AO
Family First [1]	4,700	**5,205**	**3,354**	**5,205**
Liberals [5]	1,413	**1,846**	**951**	**1,846**
One Nation [1]	**71**	**234**	**71**	**234**

Table 2: Upper bound on the number of ballot shifts (SO), or additions (AO), required to elect a specific candidate to a seat in the Tasmania 2016 Federal Senate Election.

changed. A small error percentage in the automatic scanning of ballots could also have changed the election. The set of informal ballots is large enough to influence the election outcome, if they had been cast properly, in VIC, NSW, and QLD in 2019, and in all regions except ACT in 2016.

Table 2 reports the size of the best manipulations found in which a specific losing candidate (e.g., the first candidate on the Family First ticket) is awarded a seat, using the 2016 Tasmanian senate election as a case study.

When comparing the different kinds of manipulation, shifting ballots is clearly more flexible, generating the smallest manipulations. These are often less than half of that required for addition- and deletion-only. Of the two heuristics, the more complex *Guided-Greedy* is able to find better manipulations in general, but does not dominate *Greedy*.

Source code for *Greedy* and *Guided-Greedy* can be found at: https://github.com/michelleblom/STV-Manipulator

7 Conclusion

The validity of election results is a question of crucial interest to any democracy. In this paper we provide the first method we are aware of which is able to determine upper bounds on manipulations required to change the result of large real world STV elections. This is important knowledge for reasoning about how valid an election result is. We see that, while for most of the elections we consider the manipulation we find is substantial, there are cases where very few ballots need to be changed to change the overall result.

References

Blom, M.; Teague, V.; Stuckey, P. J.; and Tidhar, R. 2016. Efficient computation of exact IRV margins. In *European Conference on Artificial Intelligence (ECAI)*, 480–488.

Blom, M.; Stuckey, P. J.; and Teague, V. J. 2019. Toward computing the margin of victory in single transferable vote elections. *INFORMS Journal on Computing*.

Cary, D. 2011. Estimating the margin of victory for instant-runoff voting. In *USENIX Accurate Electronic Voting Technology Workshop on Trustworthy Elections*.

Conway, A. 2019. Australian federal senate simulator. https://github.com/SiliconEconometrics/ PublicService. Accessed: August 2019.

Magrino, T.; Rivest, R.; Shen, E.; and Wagner, D. 2011. Computing the margin of victory in IRV elections. In *USENIX Accurate Electronic Voting Technology Workshop on Trustworthy Elections*.

Sarwate, A.; Checkoway, S.; and Shacham, H. 2013. Risk-limiting audits and the margin of victory in nonplurality elections. *Politics, and Policy* 3(3):29–64.

2019. Australian federal senate election rules. www. austlii. edu.au/au/legis/cth/consol_act/ cea1918233/s273.html. Accessed: Aug 2019.

Weeks, L. 2011. Tolerable Chance or Undesirable Arbitrariness? Distributing Surplus Votes Under PR-STV. *Parliamentary Affairs* 64:530–551.