Ballot-polling Risk Limiting Audits for IRV Elections

Michelle Blom, Peter J. Stuckey, and Vanessa J. Teague

[michelle.blom,p.stuckey,vjteague]@unimelb.edu.au
School of Computing and Information Systems
The University of Melbourne
Parkville, Australia

Abstract. Risk-limiting post election audits guarantee a high probability of correcting incorrect election results, independent of why the result was incorrect. Ballot-polling audits select ballots at random and interpret those ballots as evidence for and against the actual recorded result, continuing this process until either they support the recorded result, or they find evidence that it is wrong. Ballot-polling for first-past-the-post elections is well understood, and used in some US elections. We define a number of approaches to ballot-polling risk-limiting audits for Instant Runoff Voting (IRV) elections. We show that for almost all real elections we found, we can perform a risk-limiting audit by looking at only a small fraction of the total ballots.

1 Introduction

Instant Runoff Voting (IRV) is a system of preferential voting in which voters rank candidates in order of preference. IRV is used for all parliamentary lower house elections in Australia, parliamentary elections in Fiji and Papua New Guinea, presidential elections in Ireland and Bosnia/Herzogovinia, and local elections in numerous locations worldwide, including the UK and United States. Given candidates c_1, c_2, c_3, and c_4, each vote in an IRV election is a (*possibly partial*) ranking of these candidates. A vote with the ranking $[c_1, c_2, c_3]$ expresses a first preference for candidate c_1, a second preference for c_2, and a third for c_3. The tallying of votes proceeds by distributing each vote to its first ranked candidate. The candidate with the smallest number of votes is eliminated, with their votes redistributed to subsequent, less preferred candidates. Elimination proceeds in this fashion, until a single candidate w remains, who is declared the winner.

Risk Limiting Audits [6] (RLAs) provide strong statistical evidence that the reported outcome of an election is correct. The probability with which such audits fail to detect an incorrect outcome is bounded by a *risk limit*. An RLA with a risk limit of 1%, for example, has at most a 1% chance of failing to detect that a reported election outcome is wrong. In this paper we present several methods for undertaking ballot-polling RLAs of IRV elections, by adapting a ballot-polling RLA method (BRAVO) designed for first-past-the-post or k-winner plurality elections [7].

Blom *et al* [3] demonstrated an efficient algorithm for exact IRV margin computation. This immediately allows for a risk-limiting *comparison audit* [6], assuming that there is infrastructure for comparing ballots with their electronic record. This would consist of simply assessing the number of discrepancies until the hypothesis that there were enough to change the outcome could be rejected. However, that might be very inefficient because it counts every error equally, including those that help the apparent winner or rearrange candidates with no hope of winning. It might be possible to extend Stark's sharper discrepancy measure [10] to IRV, but this is challenging because it may be hard to compute the implications of a particular discrepancy.

In this paper we instead consider ballot-polling audits for IRV, by applying BRAVO to auditing certain facts about an IRV election. In a k-winner plurality contest, BRAVO maintains a running statistic T_{wl} for each pair of apparent winner w and loser l. These statistics are updated as ballots are drawn uniformly at random. A ballot that shows a valid vote for winner w increases the T_{wl} statistic (by an amount dependent on the reported votes for the two candidates), while a ballot showing a valid vote for the loser l decreases it. When each statistic exceeds a threshold, dependent on the risk limit, we know that we have seen enough evidence to reject the hypothesis that l beat w.

Each round of IRV elimination could be regarded as a multiple-winner plurality election—this idea was explored in [9]. We denote this by **IRV**, annotated with the round and eliminated candidates. Adapting BRAVO directly to this is described in Section 5.1. This is sound, but wastes a lot of auditing work proving a much stronger result than necessary—the elimination order may be wrong though the final outcome is correct. One optimization is to eliminate batches of low-tally candidates at once when this provably doesn't affect the final outcome. These batch eliminations can also be easily audited with BRAVO—this is described in Section 5.2.

An even simpler fact turns out to be very powerful: suppose we wish to reject the hypothesis that w was eliminated before l. We can apply BRAVO immediately, counting every ballot with a *first preference* for w as a vote for w, which is conservative because w must have *at least* this tally at every stage. Any vote that mentions l without a higher preference for w is attributed to l, which is also conservative because l can have *at most* this tally. If BRAVO rejects the hypothesis that l can beat w, then we can reject the hypothesis that w is eliminated before l. We call this the Winner Only hypothesis, denoted **WO**(l, w). It can also be conditioned on a set of already-eliminated candidates $\mathcal{C}$—preferences for those candidates are simply ignored when auditing the w-l pair.

Winner-only audits are described in Section 5.3. A surprising result of this paper is that WO alone often suffices for an efficient, complete audit. In about half the real elections we simulated auditing, we found that for the announced winner w, for every loser l, hypothesis **WO**(l, w) could be efficiently rejected using BRAVO. This confirms that w won, while sidestepping almost all the complexity of IRV.

The key contribution of this paper is a good heuristic for choosing which combination of facts to audit, using BRAVO, in order to provide an efficient risk-limiting audit of an IRV election result. We present an algorithm, denoted *audit-irv*, that finds a sufficient set of facts (e.g., some version of **IRV** or **WO**(c_1, c_2) given that c_3 and c_4 have been eliminated) to prove that w won. All of these facts can be audited simultaneously using BRAVO. If one of the necessary facts is false, this will be detected, with probability of at least $1 - \alpha$, by the BRAVO audit at risk limit α.

Ideally we would like to ensure that *audit-irv* selects the set of facts that produce an optimally efficient audit, but this is very difficult. When BRAVO is assessing only a single winner, its average sample number (ASN) can be easily computed, but the expected number of samples for eliminating multiple (perhaps related) hypotheses can (as far as we know) be assessed only by simulation. *audit-irv* selects the collection of facts that minimizes the maximum ASN for each fact taken separately—this is what we mean by the "optimal" auditing program below. However, this may not actually be an optimally efficient audit, or even the optimal application of BRAVO, because it is possible that some other combination of facts can be checked together more efficiently.

Our simulations show that *audit-irv* plans a feasible IRV audit, using BRAVO, for almost all the real IRV elections we could find. Although some still require large audits, this is probably inevitable because their margins are small.

Definitions and background are in Section 3. Section 4 introduces the BRAVO ballot-polling RLA for first-past-the-post elections. Section 5 describes our ballot-polling approaches, then Section 6 simulates and evaluates them on a suite of IRV instances.

2 Related Work

There is a growing literature on the use of risk-limiting audits for auditing the outcome of varying types of election [7, 9]. Risk-limiting audits have been applied to a number of plurality (first-past-the-post) elections, including four 2008 elections in California [4] and elections in over 50 Colorado counties in 2017. General auditing procedures designed to enhance electoral integrity have been outlined by [1]. The BRAVO ballot-polling risk-limiting audit of [7], designed for first-past-the-post elections, forms the basis of our IRV ballot-polling audits.

Several approaches for designing a risk-limiting comparison audit of an IRV election have been proposed [9]. Such audits retrieve paper ballots and compare them to their corresponding electronic record – an erroneous ballot is one that does not match its electronic record. The first of these methods determines whether replacing an erroneous ballot with its correct representation changes the margin of victory of the election. The second is based on auditing the elimination order, performing a plurality audit for each round of counting. The audit performed at round r checks whether the set of candidates eliminated prior to r, viewed as a single 'super candidate', loses to the set of remaining candidates (that are still standing). We consider a similar approach, in the context of a ballot-polling audit, in this paper. We show, however, that we can more efficiently audit an IRV election outcome by simply verifying that the reported winner was not defeated by any other candidate. The third method proposed by [9] samples K ballots, and determines whether the number of erroneous ballots exceeds a defined threshold, based on the margin of victory of the election.

In parliamentary elections, such as Australian state and federal elections, the overall outcome is determined by the results of a set of such elections, one for each of a set of regions or districts. In the context of multi-level elections such as these, [5] present a linear programming-based method to compute the statistical confidence with which each district-level election should be audited, given an appropriate risk-limiting auditing method, while minimising the expected number of ballots that must be checked overall. Their approach ensures that the overall outcome is audited to a given level of statistical confidence, while varying the extent to which each district-level election is audited.

For a risk-limiting audit, the margin of victory of the election provides an indication of how many ballots will need to be sampled. Automatic methods for computing electoral margins for IRV elections have been presented by [8, 3, 2].

Initially, all candidates remain standing (are not eliminated)
While there is *more than one* candidate standing
  For every candidate c standing
    Tally (count) the ballots in which c is the highest-ranked candidate of those standing
  Eliminate the candidate with the smallest tally
The winner is the one candidate not eliminated

Fig. 1: An informal definition of the IRV counting algorithm.

3 Preliminaries

In a first-past-the-post (FPTP) election, a voter marks a single candidate on their ballot when casting their vote. The candidate who receives the most votes is declared the winner. The BRAVO risk limiting audits of [7] are designed for k-winner FPTP contests. A voter may vote for up to k of the candidates on their ballot, and the k candidates with the highest number of votes are declared winners. IRV, in contrast, is a form of preferential voting in which voters express a preference ordering over a set of candidates on their ballot. The tallying of votes in an IRV election proceeds by a series of rounds in which the candidate with the lowest number of votes is eliminated (see Figure 1) with the last remaining candidate declared the winner. All ballots in an eliminated candidate's tally are distributed to the next most-preferred (remaining) candidate in their ranking.

Let $\mathcal{C}$ be the set of candidates in an IRV election $\mathcal{B}$. We refer to sequences of candidates π in list notation (e.g., $\pi = [c_1, c_2, c_3, c_4]$), and use such sequences to represent both votes and elimination orders. An election $\mathcal{B}$ is defined as a multiset[1] of ballots, each ballot $b \in \mathcal{B}$ a sequence of candidates in $\mathcal{C}$, with no duplicates, listed in order of preference (most preferred to least preferred). Throughout this paper we use the notation $\mathit{first}(\pi) = \pi(1)$ to denote the first candidate in a sequence π. In each round of vote counting, there are a current set of eliminated candidates $\mathcal{E}$ and a current set of candidates still standing $\mathcal{S} = \mathcal{C} \setminus \mathcal{E}$. The winner c_w is the last standing candidate.

Definition 1. ***Projection*** $\mathbf{p}_{\mathcal{S}}(\pi)$ *We define the projection of a sequence π onto a set $\mathcal{S}$ as the largest subsequence of π that contains only elements of $\mathcal{S}$. (The elements keep their relative order in π). For example:*
$p_{\{c_2,c_3\}}([c_1, c_2, c_4, c_3]) = [c_2, c_3]$ *and* $p_{\{c_2,c_3,c_4,c_5\}}([c_6, c_4, c_7, c_2, c_1]) = [c_4, c_2]$.

Each candidate $c \in \mathcal{C}$ has a *tally* of ballots. Ballots are added to this tally upon the elimination of a candidate $c' \in \mathcal{C} \setminus c$, and are redistributed upon the elimination of c.

Definition 2. ***Tally*** $\mathbf{t}_{\mathcal{S}}(\mathbf{c})$ *Given candidates $\mathcal{S} \subseteq \mathcal{C}$ are still standing in an election $\mathcal{B}$, the tally for a candidate $c \in \mathcal{C}$, denoted $t_{\mathcal{S}}(c)$, is defined as the number of ballots $b \in \mathcal{B}$ for which c is the most-preferred candidate of those remaining. Recall that $p_{\mathcal{S}}(b)$ denotes the sequence of candidates mentioned in b that are also in $\mathcal{S}$.*

$$t_{\mathcal{S}}(c) = \ | \, [b \mid b \in \mathcal{B}, c = \mathit{first}(p_{\mathcal{S}}(b))] \, | \qquad (1)$$

[1] A multiset allows for the inclusion of duplicate items.

Ranking	Count
$[c_2, c_3]$	4000
$[c_1]$	20000
$[c_3, c_4]$	9000
$[c_2, c_3, c_4]$	6000
$[c_4, c_1, c_2]$	15000
$[c_1, c_3]$	6000

(a)

Candidate	Rnd1	Rnd2	Rnd3
c_1	26000	26000	26000
c_2	10000	10000	—
c_3	9000	—	—
c_4	15000	24000	30000

(b)

Table 1: An example IRV election, stating (a) the number of ballots cast with each listed ranking over four candidates, and (b) the tallies after each round of counting.

The *primary vote* of candidate $c \in \mathcal{C}$, denoted $f(c)$, is the number of votes $b \in \mathcal{B}$ for which c is ranked highest. Note that $f(c) = t_{\mathcal{C}}(c)$.

$$f(c) = \ | \ [b \mid b \in \mathcal{B}, c = \mathit{first}(b)] \ | \tag{2}$$

Example 1. Consider the IRV election of Table 1. The tallies of c_1, c_2, c_3, and c_4, in the 1^{st} counting round are 26000, 10000, 9000, and 15000 votes. Candidate c_3 is eliminated, and 9000 ballots are distributed to c_4, who now has a tally of 24000. Candidate c_2, on 10000 votes, is eliminated next with 6000 of their ballots given to c_4 (the remainder have no subsequent preferences and are exhausted). Candidates c_1 and c_4 remain with tallies of 26000 and 30000. Candidate c_1 is eliminated and c_4 elected. □

4 Ballot-polling risk-limiting audits for FPTP

The aim of ballot-polling risk limiting audits is to be reassured that the results of the election are valid even if some counting errors occurred. To this end we will consider two versions of the statistics defined in the previous section. We use the regular definition for the *recorded* values made during the election, and add a tilde ˜ to mean the *actual* values which should have been calculated. Hence $f(c)$ is the recorded primary vote for candidate c and $\tilde{f}(c)$ is the actual primary vote for the candidate.

For now we consider a simple k-winner from n candidates FPTP election where the k candidates who have the greatest number of votes are elected. All winners are elected simultaneously and there is no transfer of votes. Given a set of $\mathcal{C}$ candidates ($|\mathcal{C}| = n$) there will be a set of $\mathcal{W}$ *winners* ($|\mathcal{W}| = k$) and $\mathcal{L}$ *losers* ($|\mathcal{L}| = n - k$).

We now present the BRAVO algorithm [7] for ballot-polling risk-limiting audits of such elections (Figure 2(a)). BRAVO is applicable in elections where each ballot may express a vote for one or more candidates. For our proposed IRV audits, we apply BRAVO in contexts where each ballot represents a vote for a single candidate only (i.e., in any round of an IRV count, each ballot belongs to the tally of no more than one candidate). We describe the BRAVO algorithm in the context where each ballot b is equivalent to $\mathit{first}(b)$. Then $f(c)$ is the tally of votes for each candidate $c \in \mathcal{C}$.

The ballot-polling risk-limiting audit independently tests $k(n-k)$ null hypotheses $\{\tilde{f}(w) \leq \tilde{f}(l)\}$ for each winner/loser pair. A statistic for each test $\{T_{wl}\}$ is updated when a ballot is drawn for either its winner or its loser.

Given an overall risk limit α we can estimate for each hypothesis the number of ballot polls we expect will be required to reject the hypothesis assuming the election counts are perfectly accurate. Let p_c be the proportion of recorded votes for candidate c, i.e. $p_c = f(c)/|\mathcal{B}|$. Let s_{wl} be the proportion of recorded votes for the winner w of the votes for the winner and loser, $s_{wl} = p_w/(p_w + p_l)$. Clearly $s_{wl} > 0.5$. Then the *Average Sample Number (ASN)* [7], that is the expected number of samples to reject the null hypothesis $\{\tilde{p}_w \leq \tilde{p}_l\}$ assuming the recorded counts are correct, is given by:

$$ASN \simeq \frac{ln(1/\alpha) + 0.5ln(2s_{wl})}{(p_w ln(2s_{wl}) + p_l ln(2 - 2s_{wl}))} \tag{3}$$

Example 2. Consider the first round of the IRV election of Example 1. The null hypotheses we need to reject are $\tilde{f}(c_1) \leq \tilde{f}(c_3)$, $\tilde{f}(c_2) \leq \tilde{f}(c_3)$, $\tilde{f}(c_4) \leq \tilde{f}(c_3)$. We calculate $p_1 = 26000/60000$, $p_2 = 10000/60000$, $p_3 = 9000/60000$, $p_4 = 15000/60000$ and $s_{13} = 26000/35000$, $s_{23} = 10000/19000$, and $s_{43} = 15000/24000$. The ASN for rejecting each hypothesis, assuming $\alpha = 0.05$, is 44.5, 6885, and 246 respectively. □

5 Ballot-polling risk-limiting audits for IRV

5.1 Auditing a particular elimination order

The simplest approach to applying ballot-polling risk limiting auditing to IRV is to consider the IRV election as a number of simultaneous FPTP elections, one for each IRV round. This was previously suggested by Sarwate *et al* [9], although they do not explore it algorithmically. Note that this may perform much more auditing than required, since it verifies more than just that the eventual winner is the correct winner, but that every step in the IRV election was correct (with some confidence).

Given an election $\mathcal{B}$ of n candidates $\mathcal{C}$ let the computed elimination order of the candidates be $\pi = [c_1, c_2, \ldots, c_{n-1}, c_n]$ where c_1 is the first eliminated candidate, c_2 the second, etc, and c_n the eventual winner.

Each IRV round corresponds to a FPTP election. In the i^{th} round we have a FPTP election where $l = c_i$ is eliminated. The set of candidates of this election are $C_l = \{c_j \mid i \leq j \leq n\}$ with recorded tally $t_{C_l}(c)$ for each candidate $c \in C_l$, and loser $l = c_i$ and $n - i$ winners $C_l \setminus \{l\}$.

We can audit all these FPTP elections simultaneously, by simply considering all the null hypotheses that would violate the computed result. These are $\{\tilde{t}_{C_l}(c) \leq \tilde{t}_{C_l}(c_l) \mid 1 \leq i \leq n-1, l = c_i, c \in C_i \setminus \{l\}\}$. We represent these hypotheses by a pair (w, l) of winner $w = c$, and loser $l = c_i$. The statistic maintained for this test is T_{wl}. Note each loser only loses in one round so there is no ambiguity.

The algorithm is shown in Figure 2(b). The set of hypotheses H are again pairs (w, l) of winner w and loser l, but they are interpreted as a hypothesis for the FPTP election corresponding to the round where l was eliminated. This means the calculation of the expected ratio of votes s_{wl} must be made using the tallies from this round. It also means we must consider every ballot to see how it is interesting for that particular hypothesis. Note that for example a ballot that is exhausted after k rounds will not play any role in determining statistics for later round hypotheses.

```
bravo(B̃,𝒲,ℒ,α,M)
  for(w ∈ 𝒲, l ∈ ℒ)
    T_wl := 1
    s_wl := f(w)/(f(w) + f(l))
  H := 𝒲 × ℒ
  m := 0
  while(m < M ∧ H ≠ ∅)
    randomly draw ballot b from B̃
    m := m + 1
    if(first(b) ∈ 𝒲)
      for((w,l) ∈ H, w = first(b))
        T_wl := T_wl × 2s_wl
        if(T_wl ≥ 1/α)
          % reject the null hypothesis
          H = H − {(w,l)}
    elseif(first(b) ∈ ℒ)
      for((w,l) ∈ H, l = first(b))
        T_wl := T_wl × 2(1 − s_wl)
  if(H = ∅)
    % reported results stand
    return true
  else % full recount required
    return false
```

(a)

```
irvbravo(B̃,π,α,M)
  H := ∅
  for(i ∈ 1..|π| − 1)
    l := π(i)
    C_l := {π(i), π(i+1), ..., π(|π|)}
    for(j ∈ i+1..|π|)
      w := π(j)
      T_wl := 1
      s_wl := t_{C_l}(w)/(t_{C_l}(w) + t_{C_l}(l))
      H := H ∪ {(w,l)}
  m := 0
  while(m < M ∧ H ≠ ∅)
    randomly draw ballot b from B̃
    m := m + 1
    for((w,l) ∈ H)
      if(w = first(p_{C_l}(b)))
        T_wl := T_wl × 2s_wl
        if(T_wl ≥ 1/α)
          % reject the null hypothesis
          H = H − {(w,l)}
      elseif(l = first(p_{C_l}(b)))
        T_wl := T_wl × 2(1 − s_wl)
  if(H = ∅)
    % reported results stand
    return true
  else % full recount required
    return false
```

(b)

Fig. 2: (a) BRAVO algorithm for a ballot-polling RLA audit of a FPTP election with actual ballots $\tilde{\mathcal{B}}$, declared winners $\mathcal{W}$, declared losers $\mathcal{L}$, risk limit α and limit on ballots checked M, and (b) algorithm for a ballot-polling RLA of an IRV election with actual ballots $\tilde{\mathcal{B}}$, order of elimination π, risk limit α and limit on ballots checked M. In both algorithms, ballots are drawn uniformly at random from $\tilde{\mathcal{B}}$.

Example 3. Consider the IRV election shown in Example 1. The null hypotheses we need to reject are $\tilde{f}(c_1) \leq \tilde{f}(c_3)$, $\tilde{f}(c_2) \leq \tilde{f}(c_3)$, and $\tilde{f}(c_4) \leq \tilde{f}(c_3)$ from the first round election, $\tilde{t}_{\{c_1,c_2,c_4\}}(c_1) \leq \tilde{t}_{\{c_1,c_2,c_4\}}(c_2)$ and $\tilde{t}_{\{c_1,c_2,c_4\}}(c_3) \leq \tilde{t}_{\{c_1,c_2,c_4\}}(c_2)$ from the second round election and $\tilde{t}_{\{c_1,c_4\}}(c_4) \leq \tilde{t}_{\{c_1,c_4\}}(c_1)$ from the final round. Assuming $\alpha = 0.05$ the ASNs for the first round are the same as calculated in Example 2. The ASNs for the remaining elections are 51.8, 64.0 and 1186 respectively. □

Example 4. The weakness of this naive approach is that inconsequential earlier elimination rounds can be difficult to audit even if they are irrelevant to the winner. Consider an election with five candidates c_1, c_2, c_3, c_4, c_5 and ballots (with multiplicity) $[c_1]$: 10000, $[c_2]$: 6000, $[c_3, c_2]$: 3000, $[c_3, c_1]$: 2000, $[c_4]$: 500, $[c_5]$: 499. The elimination order is $[c_5, c_4, c_3, c_2, c_1]$. Assuming $\alpha = 0.05$ then rejecting the null hypothesis that c_5 beat c_4 in the first round gives an ASN of $13,165,239$ indicating a full hand audit is required. But it is irrelevant to the election result. □

5.2 Simultaneous elimination

It is common in IRV elections to eliminate multiple candidates in a single round if it can be shown that the order of elimination cannot affect later rounds. Given an elimination order π we can simultaneously eliminate candidates $E = \{\pi(i)..\pi(i+k)\}$ if the sum of tallies of these candidates is less than the tally of the next lowest candidate. Let $C = \{\pi(i), \pi(i+1), \ldots \pi(k), \pi(k+1), \ldots \pi(n)\}$ be the set of candidates standing after the first $i-1$ have been eliminated. We can simultaneously eliminate E if:

$$t_C(c) > \sum_{c' \in E} t_C(c') \quad \forall c \in C \setminus E \tag{4}$$

This is because no matter which order the candidates in E are eliminated no candidate could ever garner a tally greater than one of the candidates in $C \setminus E$. Hence they will all be eliminated in any case. Note that since the remainder of the election only depends on the set of eliminated candidates and not their order, the simultaneous elimination can have no effect on later rounds of the election.

We can model the simultaneous elimination for auditing by considering all the simultaneously eliminated candidates E as as single loser l and rejecting hypotheses $\tilde{t}_C(c) \leq \tilde{t}_C(l)$ for each $c \in C \setminus E$. The statistic T_{wl} in this case is increased when we draw a ballot where w is the highest-ranked of remaining candidates C, and decreased when we draw a ballot where $c' \in E$ is the highest-ranked of remaining candidates C.

The elimination of all these null hypotheses is sufficient to prove that the multiple elimination is correct. This can then be combined with the audit of the rest of the elimination sequence, as described in Section 5.1, to test whether the election's announced winner is correct. Like the audit of a particular elimination sequence in Section 5.1, we are proving a stronger result than necessary, *i.e.* that a particular sequence of (possibly multiple) eliminations is valid, though there may be another way of getting the same candidate to win even if the multiple elimination isn't correct.

This often results in a much lower ASN, though not necessarily: sometimes the combined total of first preferences in E is very close to the next tally, so a lot of auditing is required. It may be better to audit each elimination individually in this case. It is possible to compute the ASN for each approach and choose the method that requires the least auditing, assuming the outcome is correct.

Example 5. Consider the election in Example 4. We can multiply eliminate the candidates $E = \{c_5, c_4\}$ since the sum of their tallies $499 + 500 < 5000$ which is the lowest tally of the other candidates. If we do this the difficult first round elimination auditing disappears. This shows the benefit of multiple elimination. The ASNs required for the joint elimination of E are 17.0, 36.2 and 49.1 as opposed to requiring a full hand audit.

Note that after this simultaneous elimination, the tallies for the three candidate election $\{c_1, c_2, c_3\}$ are $c_1 : 10000$, $c_2 : 6000$ and $c_3 : 5000$ and the ASNs to reject the hypotheses $\tilde{t}_{\mathcal{C}}(c_1) \leq \tilde{t}_{\mathcal{C}}(c_3)$ and $\tilde{t}_{\mathcal{C}}(c_2) \leq \tilde{t}_{\mathcal{C}}(c_3)$ are 77.6 and 1402 respectively.

Note we could also simultaneously eliminate the candidates $E = \{c_5, c_4, c_3\}$ since the sum of their tallies $499 + 500 + 5000 < 6000$ which is the lowest tally of the other candidate (that of c_2). But this will lead to a very difficult hypothesis to reject, $\tilde{t}_{\mathcal{C}}(c_2) \leq \tilde{t}_{\mathcal{C}}(\{c_5, c_4, c_3\})$ since the tallies are almost identical! The ASN is 158,156,493! This illustrates that multiple elimination may not always be beneficial. □

5.3 Winner only auditing

Up until now we consider auditing the entire IRV process to ensure that we are confident on all its outcomes. This is too strong since even if earlier eliminations happened in a different order it may not have any effect on the eventual winner.

Example 6. Consider an election with ballots $[c_1, c_2, c_3]$: 10000, $[c_2, c_1, c_3]$: 6000 and $[c_3, c_1, c_2]$: 5999. No simultaneous elimination is possible, and auditing that c_3 is eliminated before c_2 will certainly require a full hand audit. But even if c_2 were eliminated first it would not change the winner of the election. □

An alternate approach to ballot-polling RLAs for IRV elections is to simply reject the $n-1$ null hypotheses $\{\tilde{f}(w) \leq \tilde{t}_{\{w,l\}}(l)\}$ where w is the declared winner of the IRV election, and $l \in \mathcal{C} \setminus \{w\}$. This hypothesis states that l gets more votes than w where l is given the maximal possible votes it could ever achieve before w is eliminated, and w gets only its first round votes (the minimal possible votes it could ever hold). When we reject this hypothesis we are confident that there could not be any elimination order where w is eliminated before l. If all these hypotheses are rejected then we are assured that w is the winner of the election, independent of a particular elimination order.

Example 7. Consider the election of Example 6. We must reject the hypotheses that $\{\tilde{f}(c_1) \leq \tilde{t}_{\{c_1,c_2\}}(c_2)\}$ (c_1 is eliminated before c_2) and $\{\tilde{f}(c_1) \leq \tilde{t}_{\{c_1,c_2\}}(c_3)\}$ (c_1 is eliminated before c_3). The primary votes for c_1 are 10000, while the maximum votes that c_2 can achieve before c_1 is eliminated are 6000. Simultaneously the maximum votes that c_3 can achieve before c_1 is eliminated are 5999. Auditing to reject these hypotheses is not difficult. The ASNs are 98.4 and 98.3 ballots.

Note however that if the $[c_2, c_1, c_3]$ ballots were changed to be $[c_2, c_3, c_1]$ then the maximum votes that c_3 can achieve are 12000, and the hypothesis that (c_1 is eliminated before c_3) could not be rejected. Indeed in this case just changing a single vote could result in c_3 winning the election, so this election will need a full recount. □

There are, of course, some circumstances in which this does not work efficiently even though the margin of victory is large, for example if there are two runners-up who mostly (but not exclusively) preference each other.

Example 8. Consider an election with ballots $[c_1, c_2, c_3]$: 10000, $[c_2, c_3, c_1]$: 5000 $[c_2, c_1, c_3]$: 1500, $[c_3, c_2, c_1]$: 5000 and $[c_3, c_1, c_2]$: 500, and winner c_2. We cannot validate that c_2 won the election by a winner-only audit as we cannot reject the hypotheses that $\{\tilde{f}(c_2) \leq \tilde{t}_{\{c_2,c_1\}}(l)\}$. The winner's first preference tally is 6,500, while the total number of votes c_1 could have prior to c_2 being eliminated is 10,500. □

5.4 A general algorithm for finding efficient RLAs for IRV

This idea can be generalised to a method of choosing the set of facts that can be checked most efficiently (assuming no errors are found). We present an algorithm that achieves this by finding the easiest way to show that all election outcomes in which a candidate other than c_w won, did not arise, with a given level of statistical confidence.

Our algorithm, *audit-irv*, outlined in Figure 3, explores the tree of alternate elimination sequences, ending in a candidate $c' \neq c_w$. Each node is a partial (or complete) elimination sequence. For each node π, we consider the set of hypotheses that (i) can be proven with an application of BRAVO and (ii) any one of which disproves the outcome that π represents. We label each node π with the hypothesis h from this set that requires the least number of anticipated ballot polls (ASN) to prove, denoted $asn(h)$. We use the notation $h(\pi)$ and $asn(\pi)$ to represent the hypothesis assigned to π and the ASN for this hypothesis, respectively. Our algorithm finds a set of hypotheses to prove, denoted *audits*, that: validates the correctness of a given election outcome, with risk limit α; and for which the largest ASN of these hypothesis is minimised.

Note that our *risk-limit* follows directly from BRAVO: if the election outcome is wrong, then one of the facts in h must be false—a BRAVO audit with risk limit α will detect this with probability of at least $1 - \alpha$. However, our estimate of *efficiency* is only heuristic: ASNs for testing a single fact can be derived analytically, but the expected number of samples required to reject multiple hypothesis at once is very hard to compute, even if there are no discrepancies. We make a best guess based on the maximum ASN for any single fact—this is what we meant by "optimal" in this section, though it may not guarantee an optimally efficient audit overall. In Section 6 we describe simulated sample numbers for the results of our algorithm applied to real elections (assuming no discrepancies).

Consider a partial elimination sequence $\pi = [c, \ldots, w]$ of at least two candidates, leading to an alternate winner w. This sequence represents the suffix of a complete order – an outcome in which the candidates in $\mathcal{C} \setminus \pi$ have been previously eliminated, in some order. We define a function FindBestAudit(π, $\mathcal{C}$, $\mathcal{B}$, α) that finds the easiest to prove hypothesis (or fact) h, with the smallest ASN, which disproves the outcome π given risk limit α. For the outcome $\pi = [c|\ldots]$, FindBestAudit considers the following hypotheses:

WO(c,c')**:** Hypothesis that c beats $c' \in \pi$, for some $c' \in \pi$, $c' \neq c$, in a winner only audit of the form described in Section 5.3, with winner c and loser c', thus invalidating the sequence since c cannot be eliminated before c';

WO(c'',c)**:** Hypothesis that $c'' \in \mathcal{C} \setminus \pi$ beats c in a winner only audit with winner c'' and loser c, thus invalidating the sequence since c'' cannot be eliminated before c;

IRV(c,c',$\{c'' \mid c'' \in \pi\}$)**:** Hypothesis that c beats some $c' \neq c \in \pi$ in a BRAVO audit with winner c and loser c', under the assumption that the only candidates remaining are those in π (i.e. the set $\{c'' \mid c'' \in \pi\}$) with other candidates eliminated with their votes distributed to later preferences, thus invalidating the sequence since then c is not eliminated at this stage in an IRV election.

We assume that if no hypothesis exists with ASN less than $|\mathcal{B}|$ the function returns a dummy **INF** hypothesis with $ASN(\textbf{INF}) = +\infty$.

For an election with candidates $\mathcal{C}$ and winner c_w, *audit-irv* starts by adding $|\mathcal{C}| - 1$ partial elimination orders to an initially empty priority queue F, one for each alternate winner $c \neq c_w$ (Steps 4 to 9). The set *audits* is initially empty. For orders π containing a single candidate c, FindBestAudit considers the hypotheses **WO**(c'',c), candidate $c'' \neq c$ beats c in a winner only audit of the form described in Section 5.3, with winner c'' and loser c, for each $c'' \in \mathcal{C} \setminus \{c\}$. The hypothesis h with the smallest $ASN(h)$

```
audit-irv(𝒞, ℬ, c_w, α)
1  audits ← ∅
2  F ← ∅ ▷ F is a set sequences to expand (the frontier)
3  LB ← 0
   ▷ Populate F with single-candidate sequences
4  for each(c ∈ 𝒞 \ {c_w}):
5      π ← [c]
6      h ← FindBestAudit(π, 𝒞, ℬ, α)
7      hy[π] ← h ▷ Record best hypothesis for π
8      ba[π] ← π ▷ Record best ancestor sequence for π
9      F ← F ∪ {π}
   ▷ Repeatedly expand the sequence with largest ASN in F
10 while(|F| > 0):
11     π ← argmax{ASN(hy[π]) | π ∈ F}
12     F ← F \ {π}
13     if(ASN(hy[ba[π]]) ≤ LB):
14         audits ← audits ∪ {hy[ba[π]]}
15         F ← F \ {π′ ∈ F | ba[π] is a suffix of π′}
16         continue
17     for each(c ∈ 𝒞 \ π):
18         π′ ← [c] ++π
19         h ← FindBestAudit(π′, 𝒞, ℬ, α)
20         hy[π′] ← h
21         ba[π′] ← if ASN(h) < ASN(hy[ba[π]]) then π′ else ba[π]
22         if(|π′| = |𝒞|):
23             if(ASN(hy[ba[π′]]) = ∞):
24                 terminate algorithm, full recount necessary
25             else:
26                 audits ← audits ∪ {hy[ba[π′]]}
27                 LB ← max(LB, ASN(hy[ba[π′]]))
28                 F ← F \ {π′ ∈ F | ba[π] is a suffix of π′}
29                 continue
30         else:
31             F ← F ∪ {π′}
32 return audits with maximum ASN equal to LB
```

Fig. 3: The *audit-irv* algorithm for searching for a collection of hypothesis to audit, with parallel applications of BRAVO, that validate the outcome of an IRV election with candidates $\mathcal{C}$, ballots $\mathcal{B}$, and winner c_w, with a given risk limit α.

is recorded in $hy[\pi]$. The best ancestor for π is recorded in $ba[\pi]$, for these singletons sequences it is always the sequence itself.

We repeatedly find and remove a partial sequence π in F for expansion (Steps 11 and 12). This is the sequence with the (equal) highest ASN. If the best ancestor for this sequence has an ASN lower than the current lower bound LB (Steps 13 to 16) we simply add the corresponding hypothesis to $audits$ and remove any sequences in F which are subsumed by this ancestor (have it as a suffix), and restart the main loop.

Otherwise (Steps 17 to 31) we create a new elimination sequence π' with c appended to the start of π ($[c]$ $++\pi$) for each $c \in \mathcal{C} \setminus \pi$. For a new sequence π', FindBestAudit

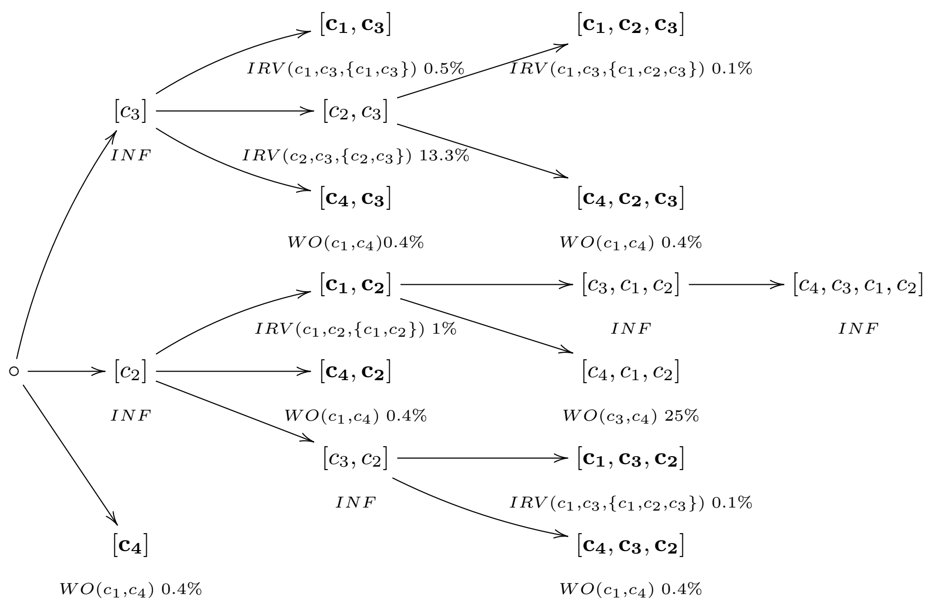

Fig. 4: Tree formed by *audit-irv* for the election of Example 9. The best hypothesis for each sequence is shown below the sequence, together with the ASN. The selected frontier is shown in bold which requires the audits: $IRV(c_1, c_3, \{c_1, c_3\})$, $IRV(c_1, c_3, \{c_1, c_2, c_3\})$, $WO(c_1, c_4)$, $IRV(c_1, c_2, \{c_1, c_2\})$.

finds the hypothesis h requiring the least auditing effort to prove. We record (Step 20) this as the hypothesis for $hy[\pi'] = h$. We calculate (Step 21) the best ancestor of π' by comparing the ASN for its hypothesis with that of its ancestor.

If the sequence π' is complete, then we known one of its ancestors (including itself) must be audited. If the best of these is infinite, we terminate, a full recount is necessary. Otherwise we add the hypothesis of its best ancestor to $audits$ and remove all sequences in F which are subsumed by this ancestor. If the sequence is not complete we simply add it into the set of sequences to be expanded F.

Example 9. Consider an election with ballots $[c_1, c_2, c_3]$: 5000, $[c_1, c_3, c_2]$: 5000 $[c_2, c_3, c_1]$: 5000, $[c_2, c_1, c_3]$: 1500, $[c_3, c_2, c_1]$: 5000, $[c_3, c_1, c_2]$: 500, and $[c_4, c_1]$: 5000, and candidates c_1 to c_4. The initial tallies are: c_1: 10000; c_2: 6500; c_3: 5500; c_4: 5000. Candidates c_4, c_3, and c_2 are eliminated, in that order, with winner c_1. In a winner only audit ($\alpha = 0.05$), we cannot show that c_1 beats c_3, or that c_1 beats c_2, as c_1's first preference tally (of 10000 votes) is less than the total number of ballots that we could attribute to c_2 and c_3 (11500 and 10500, respectively). Simultaneous elimination is not applicable in this instance, as no sequences of candidates can be eliminated in a group. In an audit of the whole elimination order (as per Section 5.1), the loss of c_4 to c_1, c_2, and c_3 is the most challenging to audit. The ASN for this audit is 25% of all ballots.

Our *audit-irv* algorithm, however, finds a set of hypotheses that can be proven with a maximum ASN of 1% (with $\alpha = 0.05$), and that consequently rule out all elimination sequences that end in a candidate other than c_1. This audit proves the hypotheses: c_1 beats c_2 if c_3 and c_4 have been eliminated (ASN of 1%); c_1 beats c_3 if c_2 and c_4 have been eliminated (ASN 0.5%); c_1 beats c_4 in a winner only audit (ASN 0.4%); and that c_1 beats c_3 if c_4 has been eliminated (ASN 0.1%). Figure 4 shows the final state of the tree explored by *audit-irv*. We record, under each sequence, the easiest hypothesis that, if proven, *disproves* an outcome ending in that sequence (alongside its ASN). The hypotheses underneath each leaf node (excluding duplicates) form our audit. Once *audit-irv* creates the node $[c_4, c_3, c_1, c_2]$ and finds that it cannot disprove this hypothesis, all descendants of $[c_1, c_2]$ are pruned from the tree. At this stage, LB is equal to 1%, and all leaves can be disproved with an ASN $\leq LB$ and the algorithm terminates. □

6 Computational Results

We have simulated the audits described in Section 5.1 (auditing the elimination order, EO), Section 5.2 (auditing with simultaneous elimination, SE), and Section 5.3 (winner only auditing, WO), on 21 US IRV elections held between 2007 and 2014, and on the IRV elections held across 93 electorates in the 2015 state election in New South Wales (NSW), Australia. We report the average number of ballot polls (expressed as a percentage of ballots cast) required to complete each of these audits, for varying risk limits, in Table 2, alongside the ASN of each audit. Each audit is run 10 times, using 10 different random seeds to control the sequence of ballots polled, and the number of ballots polled averaged over those runs. For brevity, we include the results for only a portion of the NSW seats, with the full results provided in the full version of this paper. The margin of victory (MOV) for each election is computed using the algorithm of [3].

All experiments have been conducted on a machine with an Intel Xeon Platinum 8176 chip (2.1GHz), and 1TB of RAM.

Table 2 shows that performing a winner only audit can be much easier than auditing the full elimination order (with or without the use of simultaneous elimination). This is the case for the 2013 Minneapolis Mayor, 2014 Oakland Mayor, and the 2010 Oakland D4 City Council elections. In some cases, winner only audits are more challenging (or not possible) as we seek to show that a candidate c (on just their first preference votes) could have beaten another c' (who is given all votes in which they appear before c or in which they appear, but c does not). Even if c does beat c' in the true outcome of the election, this audit may not be able to prove this (see Pierce 2008 County Executive, Oakland 2012 D5 City Council, and Aspen 2009 Mayor for examples). Auditing with simultaneous elimination (grouping several eliminated candidates into a single 'super' candidate) can be more efficient than auditing each individual elimination (see Berkeley 2010 D8 City Council, Berkeley 2012 Mayor, Oakland 2010 Mayor, San Francisco 2007 Mayor, and Sydney NSW). In some instances, however, the tally of the super candidate is quite close to that of the next eliminated candidate, resulting in a challenging audit (see Campbelltown NSW, and Berkeley 2010 D4 City Council).

Table 3 reports the maximum ASN of the audit found by *audit-irv* for each of the 26 elections examined in Table 2, alongside the ASN and average actual ballot polls

Election	$\|\mathcal{C}\|$	$\|\mathcal{B}\|$	MOV	EO				SE				WO			
				α 0.01		α 0.05		α 0.01		α 0.05		α 0.01		α 0.05	
				Polls %	ASN %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %	Polls %	ASN %
Berkeley 2010 D7 CC	4	4,682	364 (7%)	6.7	7.2	3.9	4.7	7.5	7.2	4	4.7	8.7	22.4	4.9	14.7
Berkeley 2010 D8 CC	4	5,333	878 (16%)	∞	∞	∞	∞	2.9	4.2	2	2.8	1.3	1.8	0.8	1.2
Oakland 2010 D6 CC	4	14,040	2,603 (19%)	4.0	4.4	3	2.9	0.7	0.9	0.5	0.6	0.4	0.5	0.3	0.3
Pierce 2008 CC	4	43,661	2,007 (5%)	3.1	2.2	1.8	1.4	3.1	2.2	1.8	1.4	3.2	4.1	1.8	2.7
Pierce 2008 CAD	4	159,987	8,396 (5%)	0.3	0.5	0.2	0.3	0.3	0.5	0.2	0.3	0.5	1.2	0.3	0.8
Aspen 2009 Mayor	5	2,544	89 (4%)	62.4	71.8	52.7	46.9	62.4	71.8	54.8	46.9	∞	∞	∞	∞
Berkeley 2010 D1 CC	5	6,426	1,174 (18%)	2.4	1.7	1.6	1.1	2.4	1.7	1.6	1.1	1.1	1.1	0.8	0.7
Berkeley 2010 D4 CC	5	5,708	517 (9%)	7.5	7	6	4.7	28.7	40.7	17.8	26.6	4.9	7.3	3.8	4.8
Oakland 2012 D5 CC	5	13,482	486 (4%)	11.2	10.3	7.3	6.7	15.1	10.3	11.8	6.7	∞	∞	∞	∞
Pierce 2008 CE	5	312,771	2,027 (1%)	11.6	15.1	7.6	9.8	11.6	15.1	7.6	9.8	∞	∞	∞	∞
San Leandro 2012 D4 CC	5	28,703	2,332 (8%)	9.3	9.7	6.3	6.3	9.3	9.7	6.3	6.3	1.1	4.4	0.8	2.9
Oakland 2012 D3 CC	7	26,761	386 (1%)	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
Pierce 2008 CAS	7	312,771	1,111 (0.4%)	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
San Leandro 2010 Mayor	7	23,494	116 (0.5%)	∞	∞	92.9	∞	∞	∞	92.9	∞	∞	∞	∞	∞
Berkeley 2012 Mayor	8	57,492	8,522 (15%)	94.6	∞	77	∞	2.3	2.6	1.6	1.7	0.2	0.2	0.1	0.2
Oakland 2010 D4 CC	8	23,884	2,329 (10%)	∞	∞	76.4	∞	∞	∞	∞	∞	0.9	3.1	0.6	2
Aspen 2009 CC	11	2,544	35 (1%)	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
Oakland 2010 Mayor	11	122,268	1,013 (1%)	∞	∞	∞	∞	21.5	23.8	15	15.5	∞	∞	∞	∞
Oakland 2014 Mayor	11	101,431	10,201 (10%)	∞	∞	∞	∞	∞	∞	∞	∞	0.8	19.8	0.5	12.9
San Francisco 2007 Mayor	18	149,465	50,837 (34%)	∞	∞	∞	∞	0.03	0.03	0.02	0.02	0.01	0.01	0.01	0.01
Minneapolis 2013 Mayor	36	79,415	6,949 (9%)	∞	∞	∞	∞	∞	∞	∞	∞	0.5	3.1	0.3	2.1
Balmain NSW 2015	7	46,952	1,731 (3.7%)	∞	∞	∞	∞	83.8	∞	65.4	82	5.2	31.6	3.7	20.6
Campbelltown NSW 2015	5	45,124	3,096 (6.9%)	13.6	12.2	8.4	8	∞	∞	∞	∞	1.3	1.7	0.9	1.1
Gosford NSW 2015	6	48,259	102 (0.2%)	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
Lake Macquarie NSW 2015	7	47,698	4,253 (8.9%)	27.7	22.8	14.5	15	6.9	7.8	3.2	5.1	0.7	1.6	0.5	1
Sydney NSW 2015	8	42,747	2,864 (6.7%)	∞	∞	∞	∞	3.3	4.6	2.2	3	1.6	6.9	1	4.5

Table 2: Average ballot polls performed (as a percentage of ballots cast) over 10 simulated audits of 26 IRV elections using a series of different auditing methods (with an α of 0.01 and 0.05): auditing the elimination order (EO); auditing with simultaneous elimination (SE); and winner only auditing (WO). Also reported is each elections margin of victory (MOV). The notation ∞ indicates a percentage of ballots (or ASN) greater than 100%. CC, CE, CAD, and CAS denote City Council, County Executive, County Auditor, and County Assessor.

required across 10 simulations of the best alternative audit (elimination order EO, simultaneous elimination SE, and winner only WO). Also reported is the runtime (in seconds) of *audit-irv* and the number nodes expanded (note that this does not include the creation of nodes forming the initial queue). Our *audit-irv* algorithm finds an audit (a collection of facts to prove by simultaneous applications of BRAVO) with an ASN that is equal to or lower than the ASN of the best alternative. The Oakland 2012 D3 City Council and Pierce 2008 County Assessor elections are particularly interesting. We are able to find a method of auditing the outcome of these elections that is significantly easier than the EO, SE, and WO methods, which suggest a full recount. The ASN is just an estimate, however, and the actual auditing effort required may deviate from this. For the Balmain NSW election, for example, the ASN of the best alternative audit (WO) is 20.6%. The average actual number of ballot polls required is 3.7% of the total, across 10 simulations of the audit. The ASN and actual audit effort required for the *audit-irv* audit in this instance is 1.9% and 3.2%, respectively. For the Oakland 2014 Mayor election, the ASN of the best alternative audit (WO) is 12.9% while the average actual auditing effort required is 0.5%. In contrast, the ASN of the audit found by *audit-irv* is 0.1% while the average actual effort required is 5.4%.

7 Conclusion

This paper provides a comprehensive, practical method of conducting risk-limiting ballot-polling audits for IRV. We use Stark's BRAVO as a black box, and show how to combine facts together to audit an IRV outcome. Most can be audited very efficiently. This algorithm dominates other approaches to auditing IRV elections. Over a collection of parliamentary seats or council races, most outcomes could be confirmed quickly with very little effort, while others would require some more careful auditing, and those with very small margins could be identified immediately and sent for a full manual recount.

References

1. Tigran Antonyan, Seda Davtyan, Sotirios Kentros, Aggelos Kiayias, Laurent Michel, Nicolas Nicolaou, Alexander Russell, and Alexander A Shvartsman. State-wide elections, optical scan voting systems, and the pursuit of integrity. *IEEE Transactions on Information Forensics and Security*, 4(4):597–610, 2009.
2. Bernhard Beckert, Michael Kirsten, Vladimir Klebanov, and Carsten Schürmann. Automatic margin computation for risk-limiting audits. In *International Joint Conference on Electronic Voting*, pages 18–35. Springer, 2016.
3. M. Blom, P. J. Stuckey, V. Teague, and R. Tidhar. Efficient Computation of Exact IRV Margins. In *European Conference on AI (ECAI)*, pages 480–487, 2016.
4. J.L. Hall, L.W. Miratrix, P.B. Stark, M. Briones, E. Ginnold, F. Oakley, M. Peaden, G. Pellerin, T. Stanionis, and T. Webber. Implementing risk-limiting post-election audits in California. In *Proc. 2009 Electronic Voting Technology Workshop/Workshop on Trustworthy Elections (EVT/WOTE '09)*, Montreal, Canada, August 2009. USENIX.
5. Joshua A Kroll, J Alex Halderman, and Edward W Felten. Efficiently auditing multi-level elections. *Ann Arbor*, 1001:48109.

Election	Best Alt.			*audit-irv* ($\alpha = 0.05$)			
Election	**Audit**	**Polls %**	**ASN %**	**Polls %**	**ASN %**	**Time (s)**	**Exp.**
Berkeley 2010 D7 CC	EO	3.9	4.7	5.4	4.7	0.003	3
Berkeley 2010 D8 CC	WO	0.8	1.2	0.9	0.9	0.01	6
Oakland 2010 D6 CC	WO	0.3	0.3	0.3	0.3	0.01	3
Pierce 2008 CC	EO,SE	1.8	1.4	1.5	1.4	0.03	3
Pierce 2008 CAD	EO,SE	0.2	0.3	0.3	0.3	0.1	3
Aspen 2009 Mayor	EO	52.7	46.9	28.1	46.9	0.01	9
Berkeley 2010 D1 CC	WO	0.8	0.7	0.6	0.6	0.01	5
Berkeley 2010 D4 CC	WO	3.8	4.8	1.6	2.7	0.01	5
Oakland 2012 D5 CC	EO	7.3	6.7	5.2	6.7	0.02	5
Pierce 2008 CE	EO,SE	7.6	9.8	13.9	9.8	0.9	10
San Leandro 2012 D4 CC	WO	0.8	2.9	0.8	0.6	0.06	8
Oakland 2012 D3 CC	**–**	∞	∞	**14.2**	**13.1**	**0.2**	**20**
Pierce 2008 CAS	**–**	∞	∞	**17**	**22.7**	**3.4**	**28**
San Leandro 2010 Mayor	EO,SE	92.9	∞	87.6	∞	0.08	8
Berkeley 2012 Mayor	WO	0.1	0.2	0.1	0.1	0.3	14
Oakland 2010 D4 CC	WO	0.6	2	0.6	0.5	0.3	15
Aspen 2009 CC	–	∞	∞	∞	∞	0.4	172
Oakland 2010 Mayor	SE	15	15.5	15.3	15.5	2.7	44
Oakland 2014 Mayor	WO	0.5	12.9	5.4	0.1	106	606
San Francisco 2007 Mayor	WO	0.01	0.01	0.01	0.01	23	130
Minneapolis 2013 Mayor	WO	0.3	2.1	0.2	0.2	10.8	43
Balmain NSW 2015	WO	3.7	20.6	3.2	1.9	0.2	8
Campbelltown NSW 2015	WO	0.9	1.1	0.8	0.7	0.1	5
Gosford NSW 2015	–	∞	∞	∞	∞	0.1	6
Lake Macquarie NSW 2015	WO	0.5	1	0.5	0.3	0.2	8
Sydney NSW 2015	SE	1	4.5	1.3	0.7	0.2	11

Table 3: ASN, and average ballot polls required across 10 simulations, of audit found by *audit-irv* for 26 IRV elections, alongside best known alternative (EO, SE, or WO) given a risk limit α of 0.05. Notation ∞ indicates a percentage of ballots (or ASN) greater than 100%. 'Exp' denotes the number of node expansions performed by *audit-irv*.

6. M. Lindeman and P.B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security and Privacy*, 10:42–49, 2012.
7. M. Lindeman, P.B. Stark, and V. Yates. BRAVO: Ballot-polling risk-limiting audits to verify outcomes. In *Proceedings of the 2011 Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (EVT/WOTE '11)*. USENIX, 2012.
8. T.R. Magrino, R.L. Rivest, E. Shen, and D.A. Wagner. Computing the margin of victory in IRV elections. In *USENIX Accurate Electronic Voting Technology Workshop: Workshop on Trustworthy Elections*, USENIX Association Berkeley, CA, USA, 2011.
9. A.D. Sarwate, S. Checkoway, and H. Shacham. Risk-limiting audits and the margin of victory in nonplurality elections. *Politics, and Policy*, 3(3):29–64, 2013.
10. Philip B Stark. A sharper discrepancy measure for post-election audits. *The Annals of Applied Statistics*, pages 982–985, 2008.