Analysis of the 2021 Instant Run-Off Elections in Utah

Jiří Navrátil
Independent Researcher
litarvan@gmail.com

Warren D. Smith
The Center for Range Voting
rangevoting.org

August 29, 2022

1 Executive Summary

We analyzed the 2021 Ranked Choice Voting elections in Utah County and Moab (the capitol of Grand County), focusing on the Instant Runoff Voting (IRV) algorithm. We found three issues:

1. The fractions of ballots discarded and those that needed rectification exceeded 10% in 7 of the 17 races (across 4 municipalities) indicating a considerable degree of voter confusion (Table 1).

2. Four different election pathologies were detected in the Council (Seat 1) race in Moab: failure to elect a consensus winner, monotonicity failure and two participation failures (Table 2). A "spoiler" candidate, which IRV proponents have claimed this method prevents, was also detected.

3. Four towns elected *two* seats by discarding the winner of the first seat, then re-running IRV with the resulting modified ballots to determine the winner of the second seat. This all four times caused the second-place finisher in the first election, *not* to win the 2nd seat, but rather to finish second *again*, somewhat frustratingly for them.

Overall, IRV elected the same winner as standard plurality in 15 of 16 races with ≥ 3 candidates, with the single changed outcome in the Vineyard City Council (Seat 2) race. We also analyzed the data from a recount viewpoint – implementing the Utah Election Code rules for automatic recounts versus Blom et al. [10]'s "Exact Margin of Victory (EMOV)" method. According to the former, a recount was justified in two races, namely the Springville (4yr) and Moab City Councils; but only Moab was actually recounted. However, EMOV showed recounting Springville almost certainly would have been inconsequential (Table 3).

As far as we know, this is the first-of-kind analysis of IRV elections in the state of Utah and it highlights the paradoxical properties of the IRV algorithm

— often incorrectly dismissed as too rare to worry about — showing that these unfortunately indeed occur in real-world elections hence really are worrisome. In conjunction with the above-mentioned ballot issues, these problems cast doubt on the wisdom of IRV for Utah. We believe there are better alternatives than IRV, e.g. "range voting," and these should become part of a debate towards fundamentally rethinking the program.

2 Introduction

On March 19, 2018, Utah Bill HB0035[1] was signed by then-Gov. Gary Herbert launching a "Municipal Alternate Voting Methods Pilot Project." The project allows Utah municipalities to conduct preferential "Ranked-Choice Voting (RCV)" elections with Instant Runoff Voting (IRV) as the method of tallying.

Unlike in plain plurality elections, RCV allows voters to rank multiple candidates by relative preference. A voter may choose to rank all or a subset of the candidates in a race. Tied preferences are forbidden. The IRV algorithm works in multiple rounds as follows: in each round, only the top candidates of each ballot are tallied. If there is no majority winner[2], the candidate topping the fewest ballots gets eliminated and his/her votes automatically transfer to the next-ranked candidate on each ballot. The eliminations continue until one of the candidates achieves majority.

Pitching RCV to municipalities has revolved around (1) a financial argument (RCV makes primaries unnecessary), and (2) supposed electoral benefits (e.g., avoiding the "spoiler effect", producing a "majority" winner, encouraging "amicable" campaigns)[15]. As a result, 9 municipalities in Salt Lake County, 6 in Utah County, and the city of Moab participated in the pilot program in November 2021's general elections. In total, 38 races used RCV. A subset of the municipalities conducted voter surveys in order to gauge overall satisfaction with the RCV process. While assessing RCV usability from voters' perspective is important, *voter* surveys only capture half of the overall RCV process. The other half, namely *tallying*/counting, is equally crucial.

To the best of our knowledge, this report is the first public in-depth analysis of the IRV counting process on the above-mentioned elections. After briefly introducing the relevant IRV properties, we'll describe results obtained on data from 17 races collected in Utah County and the city of Moab.

The data exhibit several paradoxes/pathologies. Our findings underscore the importance of this kind of analysis to assess the ongoing pilot program.

3 Analysis

We investigate "paradoxes" in IRV vote-counting. These phenomena, sometimes also called "pathologies," may include the so-called monotonicity, reversal, and

[1] https://le.utah.gov/ 2018/bills/static/HB0035.html

[2] Candidate whose tally exceeds 50% of the total votes.

participation failures. The following subsections explain these terms and introduce the data we used.

3.1 Ballot Ranking Data

Ballot Structure: A schematic sample of an RCV ballot used in Utah is shown in Figure 1. The voter specifies a preference order for a candidate (e.g., "Red") by marking the corresponding column in the candidate's row. Ideally, every voter specifies a complete ranking, i.e., marks *all* rows with a unique preference. But partial rankings are allowed - for instance, a voter might mark her 1st and 2nd choice, then leave the rest of the ballot empty. Unfortunately, the ballot schema also allows erroneous entries: (1) assigning multiple ranks to one candidate, (2) assigning multiple candidates same rank (**overvoting**), (3) leaving a particular rank position unfilled (**undervote**). The Utah Election Code says how to handle such ballots: A ballot is truncated at a first occurrence of under/overvote, with any subsequent ranks on that ballot ignored! In particular, any ballot with an under- or overvote at top rank is entirely discarded.

	1 First choice	2 Second choice	3 Third choice	4 Fourth choice
Red	⬭ 1	⬭ 2	⬭ 3	⬭ 4
Orange	⬭ 1	⬭ 2	⬭ 3	⬭ 4
Green	⬭ 1	⬭ 2	⬭ 3	⬭ 4
Yellow	⬭ 1	⬭ 2	⬭ 3	⬭ 4

Figure 1: A schematic sample of an RCV ballot for an election involving 4 candidates: "Red," "Orange," "Green," and "Yellow" (retrieved from `https://slco.org/clerk/elections/ranked-choice-voting/`.

Data set: For our analysis we used data from the Utah County Clerk Office [11]. This set contains 17 races comprising 6 municipalities in Utah County plus 1 other municipality – Moab[3] in Grand County. There "really" were only 16, not 17 IRV races in the sense that one (Genola Council seat 2) had only 2 candidates; but IRV and plain plurality only can differ if there are at least 3 candidates. There were two race-types: mayoral and city council. Some council races involved two "seats" to be elected. In these cases, the IRV winner received the first seat; then his/her name was removed from the ballots; then IRV was re-run to determine the second winner. Note, this does *not* necessarily award the 2nd seat to IRV's "second-place finisher" (i.e. the candidate who lost the final IRV round to seat 1's winner) – and whenever it does not, that itself is

[3]The data we received for Moab are from the official Release #3 [4]. There has been a subsequent recount in this election, data of which we are in process of obtaining.

yet another kind of self-contradictory "phenomenon" which we call **Second-Winner Misidentification (SWM)**.

The data we received for each race was raw rankings stripped of voter-identifications, precinct info, etc.

Table 1 summarizes the race statistics. Multi-seat races, namely the Genola, Lehi, Springville, Vineyard, and Woodlang Hills councils, appear on separate rows with the seat specified in brackets. The number of voters and candidates are given in the first two columns, respectively. The largest race both in terms of voters and candidates is Lehi Council Seat 1, with 10902 voters and 9 candidates. By far the smallest race (296 voters and 2 candidates) was Genola City Council seat 2, *but* as we already pointed out that was not really an "IRV" race at all.

Column "Average Length" shows the average length of a voter's ranking. For example, the mayoral race in Moab had 6 candidates but voters only ranked an average of about 3 candidates before leaving the rest of their ballot unfilled. Note that we count ballots of length $C-1$ as the full length C (with C the number of candidates) since the last ranking then is implied. E.g. in a 6-candidate race, if a voter ranks 5, she effectively ranked all 6.

Column "#Discarded Ballots" lists the number of ballots (and their percentage) that were completely excluded from the tally because of being invalid due to an under/overvote at top rank. Note that in many second-seat races a considerable increase in ballot loss can be seen. This is either due to voters only ranking one candidate or ranking multiple candidates but with a gap at the second rank position. (Example: their top-ranked candidate wins Seat 1, hence is removed from the ballot, and now the adjacent gap becomes an undervote at the top of the ballot in the seat-2 election – oops.) The percentages of discarded ballots ranged between 0.2% (Elk Ridge Mayor) up to the incredibly high 32.8% (Genola Council Seat 2).

Column "#Impure Ballots" includes any ballots that were not excluded a-priori but had to be rectified before counting. This includes the cases of gaps in ranking (an undervote followed by a valid candidate) and overvotes. In both cases Utah truncated the ballots at the first offending entry. While not invalidating the entire ballot, the voter's intent thus was altered and hence we refer to such ballots as "impure." The percentage of impure ballots ranged from 0.4% (Elk Ridge Mayor) to 41.9% (Genola Council Seat 2). Taking both the discarded and impure ballots together, in some instances the fraction of problematic ballots is extremely high: 38% in Moab (Seat 2), 58% in Genola (Seat 1), and 75% in Genola (Seat 2).

3.2 Instant Runoff Voting Paradoxes and Phenomena

Ranked Choice Voting (RCV) has been touted as a way to give a voter more expressive power and to rectify other problems associated with plurality voting. However, the counting method Utah uniquely tied to RCV, namely Instant Runoff Voting (IRV), exhibits several surprising possibly-damning phenomena, some of which can be called "defects" or "pathologies." We summarize some we

Race (Seat)	#Ballots	#Candidates	Average Length	#Discarded Ballots (%)		#Impure Ballots (%)	
Elk Ridge Mayor	1309	3	2.7	2	(0.2%)	5	(0.4%)
Genola Council (1)	337	3	2.6	56	(16.6%)	140	(41.5%)
Genola Council (2)	296	2*	2.0	97	(32.8%)	124	(41.9%)
Lehi Council (1)	10902	9	7.2	569	(5.2%)	575	(5.3%)
Lehi Council (2)	10841	8	6.4	630	(5.8%)	565	(5.2%)
Springville Mayor	6107	3	2.5	200	(3.3%)	48	(0.8%)
Springville Council 4yr (1)	6031	7	5.9	276	(4.6%)	273	(4.5%)
Springville Council 4yr (2)	5946	6	5.0	361	(6.1%)	269	(4.5%)
Springville Council 2yr	6040	3	2.7	267	(4.4%)	54	(0.9%)
Vineyard Mayor	1534	3	2.6	6	(0.4%)	7	(0.5%)
Vineyard Council (1)	1517	5	3.8	23	(1.5%)	13	(0.9%)
Vineyard Council (2)	1460	4	2.9	80	(5.5%)	11	(0.8%)
Woodland Hills Mayor	655	3	2.6	3	(0.5%)	3	(0.5%)
Woodland Hills Council (1)	645	4	3.4	13	(2.0%)	13	(2.0%)
Woodland Hills Council (2)	628	3	2.6	30	(4.8%)	13	(2.1%)
Moab Council (1)	1803	6	3.3	44	(2.4%)	528	(29.3%)
Moab Council (2)	1698	5	2.8	149	(8.8%)	489	(28.8%)

Table 1: Ballot statistics per race. A ballot is discarded if there is under/overvote at the first-rank position. An impure ballot is one that needed modification (truncation) to make it valid. *Note that the Genola City Council (2) has only two candidates and hence is not an IRV race.

analysed. For a more comprehensive list see [12, 20, 18] (and references listed there).

"More is Less Paradox" occurs when, if hypothetically one or more voters *up*rank the existing IRV winner, leaving all else unchanged, that makes that winner *lose*! This incredible phenomenon, often called *Monotonicity Failure*, can be illustrated on a simple example explained in [18, 12], along with examples of real elections where this phenomenon was observed (e.g. [14]). We will call this "Promotion Non-Monotonicity."

"Less is More Paradox" is a different kind of monotonicity failure in which if one or more voters by *down*ranking an IRV loser can make that loser win. We will call that "Demotion Non-Monotonicity."

"Participation Failure Paradox" comes in two types and relates to adding new voters to, or removing old voters from, an election. The first kind, also known as "No-Show Paradox," occurs when adding ballots ranking a loser last

causes that loser to win (we will call that "bottom-add" failure). The second type involves *removing* ballots from the election that had ranked the current IRV winner bottom, thus paradoxically stopping him from winning (we'll call this "bottom-remove" failure). Both types are termed "participation failures" as such additions and removals of voters can be viewed as voters not showing up (or showing up more) at the polls. Examples can be found in [20].

"Loser Dropout Paradox" involves a scenario where a current loser in an IRV election, by dropping out of the race, alters the election outcome (i.e. the current IRV winner would no longer win). An illustrative example can be found in [17]. A real example of this phenomenon occurred in the Burlington 2009 IRV election [14] in which candidate Wright was a "**spoiler**." Had he dropped out of the race, the consensus candidate Montroll would have won instead of Kiss.

"Reversal Failure" is a paradox whereby *reversing* all voters' rankings (i.e. ranking last candidate first, second-last candidate second, etc.) results in electing the *same* IRV winner. This obviously is a self-contradicting result, where IRV declares a candidate simultaneously "worst" and "best" [3].

"Thwarted Majorities Paradox" , also known as "Condorcet Failure" or "Consensus Failure," occurs when IRV fails to elect a candidate who would defeat every rival in direct-comparison majority votes. Such a "beats-all" winner may not always exist – but when there is one, one would hope the counting method would elect him. A number of RCV counting methods, termed "Condorcet methods" [13], guarantee electing a Condorcet winner whenever one exists. But IRV is not among them.

3.3 Procedure

Detecting Condorcet and Reversal failures is computationally straight-forward. For the former we test whether a candidate exists such that (s)he defeats all other candidates in every pair-wise preferential comparison, then check whether IRV elected that candidate. For the latter, we just reverse all ballots and see whether IRV's new winner is identical to the original winner. The Loser-Dropout and SWM paradoxes can be detected with similar ease.

Most of the other phenomena introduced in Section 3.2, however, present a challenge. Both the monotonicity and the participation failures involve a hypothetical manipulation of any number of existing votes (or their addition/removal). It is not immediately obvious how to proceed: A naive "brute force" search through all possible such manipulations of each individual ballot would involve running on the order of $(C+1)^N$ IRV elections, where C denotes the number of candidates and N the number of ballots. Even for a relatively small election (say 3 candidates, 50 voters: $4^{50} \approx 1.3 \cdot 10^{30}$ IRV elections), that exceeds what

all computers in the world can do. In general the IRV manipulation problem is considered NP-hard [21][4].

We developed an efficient algorithm that utilizes (1) Integer Linear Programming (ILP) and (2) Branch-and-Bound search paradigm, to tackle the above-mentioned complexity. Our program can detect monotonicity and participation failures (or prove none exist) within reasonable run time: most elections were tested within seconds, while the largest election (Lehi Council) took about 2 hours. Technical details about the ILP-based algorithm and code are in the Appendix.

3.4 How "partial ballots" complicate those definitions (and what we do about that)

The phenomena above have been studied theoretically for many decades. The theory assumes that all ballots contain complete rankings, i.e, all voters rank all candidates. But that's usually not true in real-world RCV elections, including our data. As Table 1 shows, the average length of a ballot varies widely and was always below the total number of candidates, i.e. some portion of voters chose to fill out their ballot only partially. Detection of certain phenomena becomes challenging for partial ballots as it is not clear how to handle the partial information. For instance, the Reversal Failure mentioned above has been defined for complete ballots where each ranking is transformed such that the last-ranked candidate becomes the first-ranked. However, if a ballot has only, say, two candidates ranked, then a corresponding reversal is not necessarily of type "worst becomes best" but of a type "second best becomes the best." Utah's IRV method regards missing candidates as "co-equal last." Our remedy for partial ballots is *ballot completion*. I.e, since the IRV algorithm disallows equal ranks, we circumvent this problem by generating all possible permutations of the missing candidates. Thus, if we input a ballot with r ranked positions and m missing candidates, we generate $m!$ new ballots each beginning with the original r, but ending with one of $m!$ specific permutations. Each such expanded ballot carries a fractional weight of $1/m!$. To make the ballots carry weights that are whole numbers, we re-scale each vote by $(C-\ell)!$ where C denotes the number of candidates in a race and ℓ the minimum ballot length observed in the election. Obviously, re-scaling an election with a large C and large N (number of voters) will result in very large number of ballots, which can make it hard to process. We use such a "combinatorial ballot completion" only in the detection of reversal failure (where needed).

3.5 Results

The results, on a per-race basis, are summarized in Table 2. Each row corresponds to a specific race and each column corresponds to a phenomenon of

[4] Arguments exist [19] that, under certain assumptions, this complexity reduces to polynomial in C and N.

interest indicating whether it occurred in that race (signified by a check-mark). The column "Plurality Disagreement" indicates whether the elected IRV winner is identical to the one elected by plain plurality voting. This disagreement should not be called a "paradox" because IRV's aim was to improve over plurality – indeed IRV-advocates would call this an IRV "*success*." In the second column ("Failed Majority") we record whether the IRV winner fell short of real majority among voters, including those whose (a) ballots were exhausted by the final round or discarded in round 1 (due to under-/over-votes at first position), and (b) whose ballots were exhausted by the final round (but not discarded in round 1 [5]. We place a first check-mark (✓) for every occurrence of (a) and an additional check-mark (✓✓) for occurrence of (b). The next column, "2nd Winner Misidentification," as mentioned above, applies to the situation where the runner-up of a seat-1 race does not win the subsequent seat-2 race. Such a disagreement is a consequence of the "staggered" reading of the ballot positions by the IRV (see discussion below). Note that we check for and report SWM in all races, also including single-seat and seat-2 elections (by running hypothetical next-seat elections). The subsequent columns (except last) comprise paradoxes introduced in Section 3.2. (Unfortunately we will see that, at least in our data, these pathologies were 10× more common than those successes!) The last column ("% Fields Read") gives the percentage of the ballots' individual fields actually "read" (accessed) by the IRV algorithm to reach the final round. To account for the fact that the C-th field in a full ballot is never accessed, we compute the total number of fields as the sum of ballot lengths in which every ballot of length C is only counted as length $C-1$. As a result, the maximum of this percentage can reach an amount of 100%. Not captured in the Table is our observation that in all 16 races (not counting Genola Seat 2 as true IRV, as mentioned above) a Condorcet winner existed, including a complete Condorcet Ordering[6].

The first noteworthy observation is that, of the 16 races, IRV elected the same winner as plain plurality voting in 15 cases, disagreeing only for the single Vineyard Council Seat 2 race. The second observation relates to failed majorities: winners in 3 of the 16 races (18.8%) did not reach a true majority among all voters, 1 of whom failed to achieve majority even after discounting blank ballots at round 1 (in Moab Council Seat 1). This observation is a direct counter-example to claims of IRV "ensuring a true majority winner" [8], and is also interesting to compare with the following statistic: In Utah, out of 137 non-IRV primary elections between 2016-2020, 23 (16.8%) were won by mere plurality [9], the rest by majority. Also, of Utah's last 50 non-IRV governor, US congress, and senate races before 2022 (now general elections only), a majority winner occurred in 96% of them[7]. (The only two exceptions were Burgess

[5]Usually, it is claimed that IRV always elects a majority winner, unfortunately tacitly assuming such a majority is defined only on ballots valid in the last round

[6]Suppose a Condorcet winner exists in an election with C candidates. Then, remove the winner thus creating a new election with $C-1$ candidates. If the entire sequence of such elections created by sequentially removing previous winners, has at each stage a Condorcet winner, it is said to have a *complete Condorcet Ordering*.

Owens 2020's and Jim Matheson 2012's congress victories in district 4, with 47.7 and 48.8% of the vote respectively.) Evidently, IRV delivered majority winners for Utah less often than plain plurality voting, despite "ensuring" them. Our third observation is that the SWM phenomenon occurs relatively frequently: IRV misidentified second winner in 7 out of the 16 races. In all but one of these cases, the second-place winner for seat 1 also ended up second place for seat 2. In these situations a significant portion of voters effectively only cast a single vote in a two-seat race[7].

The above observations aside, we have detected a total of **5 paradoxes in the 16 races**:

- Condorcet Failure in the Moab Council Seat 1 race: The Condorcet winner was Luke Wojciechowski (LW), but Jason Taylor won (see [5]), with LW ending up third. The Condorcet winner represents the consensus candidate and, in general, a failure of any RCV election method to choose that candidate is worrying [1, 14]. See Appendix A.4 for specifics.

- Promotion Non-Monotonicity in Moab Council Seat 1 race. Similar to the Burlington 2009 mayoral race [1], non-monotonicity occurred in Moab: Had 3 of the 1803 total voters changed their ballots to *uprank* the IRV winner JT, then LW would have won instead (see Appendix A.1 for specifics). (As mentioned previously, LW is also the Condorcet winner.)

- Participation Failures of both types occurred in the Moab Council Seat 1 race (recall that the IRV winner is JT=Jason Taylor):

    1. There were 3 out of 1803 voters, who, among others, ranked JT at the *bottom* of their ballot (6th rank in all three cases). Removing these three votes would make JT lose in favor of LW. In other words, had these 3 voters not showed up at the polls, their most-disliked candidate would have lost, but their actual participation made him win (see Appendix A.3 for specifics).
    2. If 765 additional voters had shown up ranking LW bottom (6th rank, in a single specific ranking), LW would have won the election (see Appendix A.2 for specifics).

- Loser-Dropout in the Moab Council Seat 1 race: Had Josie Kovash — a losing candidate — dropped out of the race, the current winner, JT, would lose to LW. This finding is a counter-example to the claim IRV "eliminates the spoiler effect"[8].

Finally, the last column of Table 2 gives the percentage of "Fields Read." Consider each ballot position as a field. We count fields as read when the IRV algorithm accesses them in order to update the candidate tally. Not all fields will be looked at by the final round. The percentage of those that were

[7]This is a symptom of a larger problem: IRV ignores asymptotically 100% of ballot information in C-candidate elections when $C \to \infty$ [2]

Race (Seat)	Plurality Disagreement	Failed Majority	2nd Winner Misidentification	Condorcet Failure	Reversal (partial)	Reversal (compl.)	Promoting NM	Demoting NM	Bottom Add	Bottom Remove	Loser Dropout	% Fields Read
Elk Ridge Mayor	·	·	·	·	·	·	·	·	·	·	·	63.1
Genola Council (1)	·	✓	·	·	·	·	·	·	·	·	·	65.4
Lehi Council (1)	·	·	✓	·	·	·	·	·	·	·	·	29.4
Lehi Council (2)	·	·	✓	·	·	·	·	·	·	·	·	31.0
Springville Mayor	·	·	·	·	·	·	·	·	·	·	·	62.5
Springville Council 4yr (1)	·	·	✓	·	·	·	·	·	·	·	·	32.9
Springville Council 4yr (2)	·	·	·	·	·	·	·	·	·	·	·	37.4
Springville Council 2yr	·	·	·	·	·	·	·	·	·	·	·	60.4
Vineyard Mayor	·	·	·	·	·	·	·	·	·	·	·	58.3
Vineyard Council (1)	·	✓	·	·	·	·	·	·	·	·	·	39.1
Vineyard Council (2)	✓	·	✓	·	·	·	·	·	·	·	·	49.8
Woodland Hills Mayor	·	·	·	·	·	·	·	·	·	·	·	66.8
Woodland Hills Council (1)	·	·	✓	·	·	·	·	·	·	·	·	50.6
Woodland Hills Council (2)	·	·	·	·	·	·	·	·	·	·	·	67.8
Moab Council (1)	·	✓✓	✓	✓	·	·	✓	·	✓	✓	✓	43.6
Moab Council (2)	·	·	✓	·	·	·	·	·	·	·	·	52.7

Table 2: Overview of the observed phenomena on the Utah County & Moab IRV data set. Note that Genola Council Seat 2 is excluded as it is not an IRV race (less than three candidates). "NM" stands for Non-Monotonicity.

accessed is reported in the column "% Fields Read." This percentage ranges between 29.4% (Lehi Council Seat 1) and 67.8% (Woodland Hills Council Seat 2) and anti-correlates with C the number of candidates (Pearson $\rho = -0.938$), consistent with [2].

3.6 Margin of Victory and Election Recounts

In plurality elections, Margin of Victory (MOV) is a simply-defined quantity: half of the vote difference between the winner and the runner-up represents the amount of votes that would have to change in order to change the election outcome. The MOV is an important quantity to record as it serves audits: the smaller it is, the greater the need for an audit. Often, election laws stipulate automatic recounts based on certain thresholds derived from this quantity.

But in IRV elections, the MOV definition is less obvious. A commonly used definition is "half of the difference between the last-round tally of the IRV winner and the runner-up" [10]. However, given the *multi*-round nature of the IRV algorithm, smaller tally discrepancies in earlier rounds could potentially impact the elimination sequence in a way that changes the winner. Blom et al. [10]

proposed an efficient algorithm to find the "Exact MOV": the smallest number of ballots that, by changing, could change the election outcome. Their algorithm produces a margin that is at most the last-round margin but in some instances may be significantly smaller (and is provably minimum). The computation in [10] requires solving a number of non-trivial optimization problems and, to the best of our knowledge, has not yet been implemented in governmental practice. The Utah State Election Code[8] seems to recognize the weakness of a simple last-round MOV, and defines the recount criteria explicitly via testing certain conditions in each IRV round. In essence: (1) if the smallest difference between the declared winner and any other candidate in any round is less than a round-dependent threshold, then a recount is triggered, or, (2) if the smallest count difference between a loser (candidate who is being eliminated in a round) and any other candidate falls below a threshold, then a recount is triggered.

Let t_r denote the critical vote fraction for a given IRV round, r. According to the Utah State Election Code, its value is obtained as follows:

$$t_r = N_r \cdot \tau_r$$

whereby N_r denotes the number of valid votes in round r and τ_r is the "recount threshold"[9] with

$$\tau_r = (C_r - 2) \cdot 0.02\% + b_r$$

where C_r is the number of active candidates in round r, the product $(C_r - 2) \cdot 0.02\%$ is termed the "candidate amplifier," and b_r is a quantity tied to the total number of votes in the election (tabulated in the above-referenced Election Code). A recount shall be ordered if

$$\#votes_r(c_w) - \#votes_r(c) \leq t_r \tag{1}$$

for any round r and any candidate c other than the winner c_w. Alternatively, the recount shall also be triggered if

$$\#votes_r(c) - \#votes_r(c_e) \leq t_r \tag{2}$$

for any round r and any candidate c other than the candidate eliminated in round r, c_e.

As part of our recount analysis, we obtained [10]'s Exact MOV code (helpfully shared by its authors) and ran it on our data. Furthermore, we implemented the Utah Election Code's prescriptions for automatic recount. Table 3 shows the recount-related quantities we found. Starting with the first two data columns, we confirm that the Last-Round Margin (half of the difference between the winner's and the loser's vote count in the last IRV round) is in all cases greater than or equal to the Exact MOV developed by Blom et al. In some instances this discrepancy was substantial, e.g. Moab Council Seat 1's EMOV was only 1 vote rather than the 29 vote last round margin!

[8] See https://le.utah.gov/xcode/Title20A/Chapter4/20A-4-S603.html
[9] See https://le.utah.gov/xcode/Title20A/Chapter4/20A-4-S601.html

Race (Seat)	Last-Round Margin	Exact MOV (Blom et al.)	Smallest Winner Difference (Round)	Recount Threshold	Smallest Loser Difference (Round)	Recount Threshold	Recount Required	Recount Performed
Elk Ridge Mayor	92	92	72 (1)	2	183 (2)	2	no	no
Genola Council (1)	35	35	69 (2)	1	5 (1)	1	no	no
Genola Council (2)	15	15	30 (1)	1	30 (1)	1	no	no
Lehi Council (1)	986	276	353 (6)	16	28 (4)	21	no	no
Lehi Council (2)	717	309	977 (5)	16	26 (4)	18	no	no
Springville Mayor	1936	9136	3871 (2)	8	57 (1)	9	no	no
Springville Council 4yr (1)	707	667	817 (4)	10	47 (1)	14	no	no
Springville Council 4yr (2)	336	321	267 (3)	10	1 (2)	11	**yes**	no
Springville Council 2yr	274	274	372 (1)	9	548 (2)	8	no	no
Vineyard Mayor	605	605	1197 (1)	3	59 (1)	3	no	no
Vineyard Council (1)	326	319	432 (1)	3	39 (3)	3	no	no
Vineyard Council (2)	93	93	6 (1)	3	186 (3)	2	no	no
Woodland Hills Mayor	32	32	64 (2)	1	6 (1)	1	no	no
Woodland Hills Council (1)	99	73	96 (1)	1	19 (2)	1	no	no
Woodland Hills Council (2)	54	36	38 (1)	1	67 (1)	1	no	no
Moab Council (1)	29	1	57 (5)	3	2 (4)	3	**yes**	yes
Moab Council (2)	226	42	10 (3)	3	39 (1)	4	no	no

Table 3: Key statistics related to recounts: Last-Round Margin and Exact Margin of Victory (MOV) are shown in the first two data columns. Smallest winner/loser differences as calculated in EQs (1) and (2) along with the corresponding recount thresholds are listed in the third and fourth column. Last two columns indicate whether a recount was required by the Utah Election Code and whether it was actually performed.

The rest of the columns relate to the Utah State Code: The column "Smallest Winner (or Loser) Difference" lists the smallest count difference between the declared winner (or the round's loser) and any other candidate with the same round (the round where this minimum occurred is listed in brackets). These quantities correspond, respectively, to the left-hand side of EQ (1) and EQ (2). The recount vote thresholds (t_r on the right-hand side of EQs (1) and (2)) are listed along with each difference type. If any of the vote differences fall below the corresponding threshold, a recount should be triggered automatically. This is reflected in the penultimate column. The final column indicates whether a recount was actually carried out (as published in [6, 4]).

In two cases, namely Springville City Council (4yr) Seat 2 and the Moab City

Council Seat 1, the recount criteria triggered. In Springville[10], the candidate Marcia Conover-Harris was eliminated with 777 votes but another candidate, Dale Watson, lay only 1 vote away with 778 votes. This difference falls below the mandated recount threshold of 11 votes. But an actual recount was not performed. The Exact MOV for this race happens to be 321 votes thus proving that the Conover-Harris vs. Watson issue would have had no impact on the race outcome (it might merely have swapped their elimination order in rounds 2 and 3). This observation also shows that the Utah Election Code appears to be more stringent than the calculated Exact MOV, tending to err on the side of a recount. In the second case (Moab), the Exact MOV and the Utah Election Code criteria do agree on a recount, which was duly performed [4] and did not change the outcome.

4 Conclusions

We presented a first-of-a-kind analysis of 2021 Municipal Instant Runoff Voting (IRV) Elections conducted in the State of Utah. While various voter surveys had already been performed across the state, the *tallying* side of these elections, e.g. the sometimes "paradoxical" behavior of the IRV algorithm, had not been studied. This report introduces IRV phenomena of interest, describes our approach to detecting them from data and discusses results of 17 races in Utah County. The results include worrisome findings: (1) significant voter ballot confusion, and (2) several IRV "paradoxes."

Utah's Municipal Alternate Voting Methods Pilot Project had the goal of testing and assessing the concept of Ranked Choice Voting and IRV algorithm. An integral part of such an assessment is evaluating the IRV algorithm on actual ranking data. Unfortunately, our experience working with county clerks' offices indicates a considerable lack of political will and commitment: in contrast to Utah County (and contrary to nationwide practice), Utah's largest county, Salt Lake, with 9 municipalities participating in the pilot program, consistently refused to release basic ranking information — the data we need to perform our type of analysis. We believe that Utah's decision makers: council members, mayors, state legislators, as well as voters want to make a well-educated decision on the basic question: should we keep IRV as our vote-counting method? Are there alternatives? If so, what are they and how do they work? In Utah, these questions have not been widely discussed, and they appear to have not even been asked yet. We hope this report and our findings will serve as the first step toward a remedy.

5 Acknowledgements

We are grateful to Josh Daniels (Utah County Clerk) and Carolyn Phippen (Executive Director, Freedom Front) for their expertise, insight and kind help

[10] see details at: `https://rcvis.com/v/21g_sp_cc4y_2_u4`

with obtaining the IRV ranking data. We also thank Dr. Michelle Blom (Senior Research Fellow, University of Melbourne) for sharing the Exact Margin of Victory computation code and her kind support with related questions.

References

[1] 2009 Burlington Mayoral Election. https://en.wikipedia.org/wiki/2009_Burlington_mayoral_election, last accessed on 07/06/2022.

[2] Ignoring Your Votes. https://www.rangevoting.org/Ignoring.html, last accessed on 08/01/2022.

[3] IRV Contradicts Itself. http://rangevoting.org/IrvRevFail.html, last accessed on 07/02/2022.

[4] Moab City 2021 Election. https://moabcity.org/236/Election-Information, last accessed on 07/02/2022.

[5] Moab City Council Election Visualization. https://rcvis.com/v/moab-council-seat-1-release-3, last accessed on 07/02/2022.

[6] Utah County Ranked Choice Election Results. https://www.utahcounty.gov/dept/clerkaud/elections/2021RankedResults.asp, last accessed on 07/02/2022.

[7] Utah Historical Election Results. https://voteinfo.utah.gov/historical-election-results/, last accessed on 08/15/2022.

[8] Utah Ranked Choice Voting. https://utahrcv.com/, last accessed on 08/13/2022.

[9] Plurality in Utah, May 2021. Report by the Office of Legislative Research and General Counsel, Government Operations Interim Committee, https://le.utah.gov/interim/2021/pdf/00002096.pdf, last accessed 08/13/2022.

[10] Michelle Blom, Vanessa Teague, Peter J. Stuckey, and Ron Tidhar. Efficient computation of exact irv margins. In *Proceedings of the Twenty-Second European Conference on Artificial Intelligence*, ECAI'16, page 480–488, NLD, 2016. IOS Press.

[11] Joshua Daniels. Personal Communication. March, 2022.

[12] Peter C. Fishburn and Steven J. Brams. Paradoxes of preferential voting. *Mathematics Magazine*, 56(4):207–214, 1983.

[13] William V. Gehrlein and Peter C. Fishburn. Condorcet's paradox and anonymous preference profiles. *Public Choice*, 26:1–18, 1976.

[14] Anthony Gierzinski, Wes Hamilton, and Warren D. Smith. Burlington Vermont 2009 IRV Mayor Election, March 2009. http://www.rangevoting.org/Burlington.html, last accessed on 07/02/2022.

[15] S. Lockhart, K. Holdaway, T. Morgan, and D. May. Utah Ranked Choice Voting - Better, Faster, Cheaper, It Works. Presented to the Draper City Council on October 6, 2020. Recording available at https://draper.granicus.com/player/clip/342, last accessed on 07/22/2022.

[16] Thomas R. Magrino, Ronald L. Rivest, Emily Shen, and David Wagner. Computing the margin of victory in irv elections. In *Proceedings of the 2011 Conference on Electronic Voting Technology/Workshop on Trustworthy Elections*, EVT/WOTE'11, page 4, USA, 2011. USENIX Association.

[17] Warren D. Smith. Example of Instant Runoff Voting Insanity. http://rangevoting.org/IRVbadRemove.html, last accessed on 07/02/2022.

[18] Warren D. Smith. (non)Monotonicity and Instant Runoff Voting. http://rangevoting.org/Monotone.html, last accessed on 07/02/2022.

[19] Warren D. Smith. Personal Communication. May 2022.

[20] Warren D. Smith. Survey of Instant Runoff Voting (IRV) Pathologies. http://rangevoting.org/IrvPathologySurvey.html, last accessed on 07/02/2022.

[21] Lirong Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, page 982–999, New York, NY, USA, 2012. Association for Computing Machinery.

A Moab City Council Race

Acronyms: AC (Anthony Charles), JT (Jason Taylor), Josie Kovash (JK), Luke Wojciechowski (LW), Mike McMurdy (MM), Randall Fox (RF).

A.1 Promotion Non-Monotonicity Manipulation

These three vote changes that up-rank the original winner JT, cause him to lose:

```
Original ranking                    Manipulated ranking
MM > AC > JK > LW > JT        ==>   JT > MM > AC > JK > LW
MM > JK > JT > AC             ==>   JT > MM > JK > AC
MM > JK > LW > AC > RF > JT   ==>   JT > MM > JK > LW > AC > RF
```

The new winner is LW.

A.2 Participation Failure - Bottom Add

By adding these 765 identical new votes ranking LW bottom, the loser, LW, will win the election:

```
765:   AC > JT > JK > MM > RF > LW
```

A.3 Participation Failure - Bottom Remove

By removing these 3 existing votes that bottom-rank the winner, JT, he will lose the election:

```
Remove 1:   JK > LW > AC > MM > RF > JT
Remove 2:   JK > LW > AC > RF > MM > JT
```

The new winner is LW.

A.4 Condorcet Failure

Below table shows pairwise preference statistics in the Moab Seat 1 race. Each row corresponds to a candidate and each column to their opponent. The count in each cell is the number of preferences the row-candidate holds against the respective opponent. Column "wins" gives the number of head-to-head wins for each row-candidate.

	wins	AC	JT	JK	LW	MM	RF
AC	(1):	--	265	216	181	320	675
JT	(4):	1183	--	896	835	1098	1298
JK	(3):	1062	839	--	562	965	1122
LW	(5):	1191	914	758	--	1060	1250
MM	(2):	902	353	661	608	--	1056
RF	(0):	41	29	35	31	29	--

The following are the preferential tallies for the (Condorcet) candidate Luke Wojciechowski, defeating all five opponents pairwise. As an example, the first line tells us that LW was preferred over AC 1191 times, while AC was preferred over LW 181 times.

```
LW  (1191)  >  AC  (181)
LW  (914)   >  JT  (835)
LW  (758)   >  JK  (562)
LW  (1060)  >  MM  (608)
LW  (1250)  >  RF  (31)
```

Recall that the IRV winner was JT.

B Anomaly Detection: Algorithms

This section gives detail on the detection approach and individual algorithms to perform the search efficiently.

B.1 Promotion Non-Monotonicity

If somebody increases their vote for candidate C (leaving the rest of their vote unchanged) that should not worsen C's chances of winning the election[18].

- $\mathcal{C}$ - set of candidates
- s - a ballot signature (tuple), e.g., (a, b, c) with $a, b, c \in \mathcal{C}$
- $\mathcal{B}$ - set of signature $\rightarrow$ signature transitions that promote winner $w \in \mathcal{C}$, i.e., $\mathcal{B}\{(s_i, s_j) : \text{s.t. signature transition promotes candidate w}\}$
- n_s - original count for signature s
- y_s - count for signature s after modification
- $b_{(\sigma,s)}$ - number of ballots changed from signature σ to signature s
- n - total votes

Set up an integer linear program (ILP) with the following constraints and objective.

Constraints:

$$y_s = n_s - \sum_{\forall\sigma:(s,\sigma)\in\mathcal{B}} b_{(s,\sigma)} + \sum_{\forall\sigma:(\sigma,s)\in\mathcal{B}} b_{(\sigma,s)} \tag{3}$$

$$0 \leq y_s \leq n \tag{4}$$

$$\sum_{\forall\sigma:(s,\sigma)\in\mathcal{B}} b_{(s,\sigma)} \leq n_s \tag{5}$$

$$0 \leq b_{(s_i,s_j)} \tag{6}$$

Add constraints enforcing specific elimination order as in [16] (also see Algorithm 1 below).

Objective:

$$\min \sum_{\forall(s,\sigma)\in\mathcal{B}} b_{(s,\sigma)} \tag{7}$$

Candidate-promoting transitions Example shown in Figure 2.

B.2 Demotion Non-Monotonicity

If somebody decreases their vote for candidate B (leaving the rest of their vote unchanged) that should not improve B's chances of winning the election[18].

Keep symbols defined as in Section B.1, replacing $\mathcal{B}$ and b with the following:

- $\mathcal{D}_l$ - set of signature $\rightarrow$ signature transitions that demote loser $l \in \mathcal{C} \setminus w$, i.e., $\mathcal{D}_l\{(s_i, s_j) : \text{s.t. signature transition demotes candidate l}\}$ for $l \in \mathcal{C} \setminus w$
- $d_{(\sigma,s)}$ - number of ballots changed from signature σ to signature s

Constraints: Applicable to a specific candidate, $l \in \mathcal{C} \setminus w$ test.

$$y_s = n_s - \sum_{\forall\sigma:(s,\sigma)\in\mathcal{D}_l} d_{(s,\sigma)} + \sum_{\forall\sigma:(\sigma,s)\in\mathcal{D}_l} d_{(\sigma,s)} \tag{8}$$

$$0 \leq y_s \leq n \tag{9}$$

$$\sum_{\forall\sigma:(s,\sigma)\in\mathcal{D}_l} d_{(s,\sigma)} \leq n_s \tag{10}$$

$$0 \leq d_{(s_i,s_j)} \tag{11}$$

Add constraints enforcing specific elimination order as in [16] (also see Algorithm 1 below).

Example of demoting transitions for a candidate b in a three-candidate election is shown in Figure 3.

Objective:

$$\min \sum_{\forall(s,\sigma)\in\mathcal{D}} d_{(s,\sigma)} \tag{12}$$

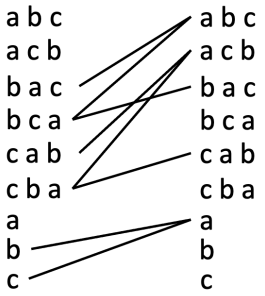

Figure 2: Example of signature transitions promoting candidate a in a three-candidate election

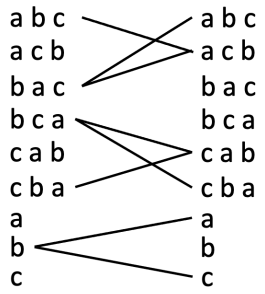

Figure 3: Example of signature transitions demoting candidate b in a three-candidate election

B.3 Participation Failure: "Bottom-Add"

Adding votes that rank a candidate B last should not improve B's chances of winning the election[12]

- $\mathcal{C}$ - set of candidates
- s - a ballot signature (tuple), e.g., (a, b, c) with $a, b, c \in \mathcal{C}$
- $\mathcal{B}$ - set of all possible signatures that rank a loser $l \in \mathcal{C} \setminus w$ at the last position, i.e., $\mathcal{B}\{(s_i) : \text{s.t. signature ranks candidate l bottom}\}$
- n_s - original count for signature s
- y_s - count for signature s after modification
- a_s - number of ballots with signature s added to the election
- n - total votes in original election

Set up a separate integer linear program (ILP) for each loser $l \in \mathcal{C} \setminus w$ with the following constraints and objective.

Constraints: $\forall s \in \mathcal{B}$

$$y_s = n_s + a_s \tag{13}$$

$$0 \leq y_s \leq 2n \tag{14}$$

$$0 \leq a_s \leq n \tag{15}$$

Add constraints enforcing specific elimination order as in [16] (also see Algorithm 1 below).

Objective:

$$\min \sum_{\forall s \in \mathcal{B}} a_s \tag{16}$$

B.4 Participation Failure: "Bottom-Remove"

Removing votes that rank winner W last should not decrease W's chances of winning the election[12]

- $\mathcal{C}$ - set of candidates
- s - a ballot signature (tuple), e.g., (a, b, c) with $a, b, c \in \mathcal{C}$
- $\mathcal{B}$ - set of all possible signatures that rank the winner $w \in \mathcal{C}$ at the bottom, i.e., $\mathcal{B}\{(s_i) : \text{s.t. signature ranks candidate w bottom}\}$
- n_s - original count for signature s
- y_s - count for signature s after modification
- a_s - number of ballots with signature s added to the election
- n - total votes in original election

Set up an integer linear program (ILP) with the following constraints and objective.

Constraints: $\forall s \in \mathcal{B}$

$$y_s = n_s - a_s \tag{17}$$

$$0 \leq y_s \leq 2n \tag{18}$$

$$0 \leq a_s \leq n_s \tag{19}$$

Add constraints enforcing specific elimination order as in [16] (also see Algorithm 1 below).

Objective:

$$\min \sum_{\forall s \in \mathcal{B}} a_s \tag{20}$$

B.5 Branch-and-Bound Search

The search adopts a tree structure to efficiently encode the ILP elimination constraints: a non-terminal node represents a partial elimination sequence, e.g., $\pi = [c_3, c_2]$ means candidate c_3 is eliminated in the first IRV round, followed by c_2 eliminated in the next; a leaf node corresponds to a complete elimination sequence (of length $C-1$, with last candidate—the winner—implied). Each node is an expansion of its parent by exactly one candidate, e.g., $[c_3, c_2] \rightarrow [c_3, c_2, c_4]$. At each node, we solve an ILP with the objective function, $\mathcal{L}$, and constraints, $\mathcal{F}$, defined in Sections B.1, B.2, B.3, and B.4, only modifying the corresponding elimination portion of the constraints from node to node. Recall that the ILP returns the minimum number of ballots that need to be modified to achieve a corresponding anomaly (e.g., a monotonicity failure) or returns a flag to indicate the problem has no solution. In the following, we refer to this return value as "score", denoted by l, i.e., $l = ILP(\mathcal{F}, \mathcal{L})$. So, $l \in \{\mathbb{N}^+, \infty\}$ with ∞ symbolizing an infeasible problem. The following holds true:

Property 1 *Let node n' be an expansion of a parent node n, with respective constraints $\mathcal{F}'$ and $\mathcal{F}$, and $\mathcal{F} \subseteq \mathcal{F}'$. Then $l' \geq l$, where $l' = ILP(\mathcal{F}', \mathcal{L})$ and $l = ILP(\mathcal{F}, \mathcal{L})$.*

In other words, the score cannot decrease when new constraints are added. Note that the ILP objective function, $\mathcal{L}$, is identical across nodes.

Pruning infeasible paths: Property 1 implies that if an ILP associated with a parent node is infeasible, so will be that of any of its descendant nodes, thus offering computational savings: We only need to prove infeasibility for the shortest elimination sequence, say, $[e_0], e_0 \in \mathcal{C}$ to be able to prune away all $(C-1)!$ elimination sequences with prefix e_0.

Pruning by upper bounding: We use a priority queue (similar to [16, 10]) to store nodes ordered by their score. Any new or queued node whose score exceeds an active upper bound is removed (and hence all its future descendants are pruned). For this purpose, an active upper bound is the smallest score ($l < \infty$) of a leaf node (with a complete elimination sequence). If no such node exists (yet), the upper bound is inactive and not tested against.

Algorithms 1 and 2 capture the entire process formally, whereby Algorithm 2 (`ManipulateElection`) uses Algorithm 1 (`EliminationConstraints`) as a sub-routine.

Notation: Projection of a candidate sequence π onto a set $\mathcal{S} \subseteq \mathcal{C}$ is an order-preserving transform yielding a new sequence, π', containing only candidates in $\mathcal{S}$. We use the following notation for projection:

$$\pi' = \pi|_{\mathcal{S}} \tag{21}$$

For example, for $\pi = [2, 4, 1, 3, 0]$ and $\mathcal{S} = \{0, 3, 4\}$, $\pi|_{\mathcal{S}} = [4, 3, 0]$.

Algorithm 1 `EliminationConstraints`

Input: π: Desired elimination sequence; $\mathcal{Y}$: Set of all signature counts after manipulation ($\mathcal{Y} = \{y_s\}_{\forall s}$)
Output: $\mathcal{F}$: set of ILP elimination constraints

$\mathcal{D} \leftarrow \{\}$ ▷ Candidates defeated
$\mathcal{F} \leftarrow \{\}$
for r $\leftarrow 1$ **to** $\min\{|\pi|, C-1\}$ **do**
  $e \leftarrow \pi_r$ ▷ Candidate eliminated at round r
  $\mathcal{S} \leftarrow \mathcal{C} \setminus \mathcal{D}$ ▷ Candidates standing
  $\mathcal{V}_e \leftarrow \{\text{signature } s \mid e \text{ is first in } s|_{\mathcal{S}}\}$ ▷ Signatures where e is top
  $t_e \leftarrow \sum_{s \in \mathcal{V}_e} y_s$ ▷ e's tally after manipulation
  for $o \leftarrow \mathcal{S} \setminus e$ **do** ▷ Loop over opponents
    $\mathcal{V}_o \leftarrow \{\text{signature } s \mid o \text{ is first in } s|_{\mathcal{S}}\}$
    $t_o \leftarrow \sum_{s \in \mathcal{V}_o} y_s$ ▷ Opponent's tally
    $\mathcal{F} \leftarrow \mathcal{F} \cup (t_e < t_o)$ ▷ e must lose to o
  end for
  $\mathcal{D} \leftarrow \mathcal{D} \setminus e$
end for

Algorithm 2 `ManipulateElection`

Input: $\mathcal{F}$: initial set of fixed ILP constraints; $\mathcal{L}$: ILP objective; c^* : Candidate to win; $\mathcal{Y}$: Set of all signature counts after manipulation ($\mathcal{Y} = \{y_s\}_{\forall s}$)
Output: U : minimum number of ballots to modify, or ∞
for $c \in \mathcal{C} \setminus c^*$ **do** ▷ Initialize priority queue
  $\pi \leftarrow [c]$
  $\mathcal{F}' \leftarrow \mathcal{F} \cup$ `EliminationConstraints`$(\pi, \mathcal{Y})$
  $l = ILP(\mathcal{F}', \mathcal{L})$
  if $l \neq \infty$ **then**
    Insert node (l, π) into Q
  end if
end for
$U \leftarrow \infty$ ▷ Initial upper bound
while $|Q| > 0$ **do**
  $(l, \pi) \leftarrow$ lowest-score node from Q
  $Q \leftarrow Q \setminus (l, \pi)$
  for $c \in \mathcal{C} \setminus \pi \setminus c^*$ **do**
    $\pi' \leftarrow \pi + [c]$
    $\mathcal{F}' = \mathcal{F} \cup$ `EliminationConstraints`$(\pi', \mathcal{Y})$
    $l' = ILP(\mathcal{F}', \mathcal{L})$
    if $l' < U$ **then**
      Insert node (l', π') into Q
      if $|\pi'| == C - 1$ **then** ▷ A complete elimination seq.
        $U \leftarrow \min(U, l')$ ▷ Update bound
      end if
    end if
  end for
end while

Candidate c^* input to `ManipulateElection` may represent a current IRV loser who, by manipulation, is made to a winner. This covers the cases of Demotion Non-Monotonicity, and Bottom-Add Participation Failure, which need to be run separately for each loser candidate. For Promoting Non-Monotonicity and Bottom-Remove Participation Failure we only need to ensure that the current IRV winner, w, loses, which requires a minor modification of Algorithm 2 enforcing that w is always placed to any but the last position in an elimination sequence. Such an algorithm is only required to be run once.

In processing the Utah data set we observed the above search algorithm yielding pruning rates between 80% and 99%, relative to the full set of possible complete sequences - a significant computational saving.