# Efficient Computation of IRV Parliamentary Election margins

Michelle Blom, Peter J. Stuckey and Vanessa Teague

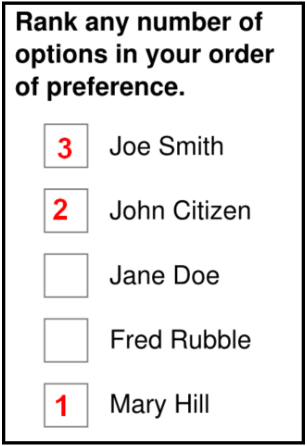
The University of Melbourne

August 31$^{st}$, 2016

THE UNIVERSITY OF MELBOURNE

# Instant Runoff Voting (IRV)

- Preferential voting scheme
- A set of candidates $\mathcal{C}$, one winner
- Each vote is a ranking over $\mathcal{C}$
- Each vote can be a *partial* ranking
  e.g., [Mary Hill, Joe Smith, John Citizen]



# Instant Runoff Voting (IRV) – An Example

4 candidates, 60 votes

| Ranking | Count |
|---|---|
| $[c_2, c_3]$ | 4 |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| $[c_2, c_3, c_4]$ | 6 |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

(a) Initial tallies

| Candidate | Rnd1 | Rnd2 | Rnd3 |
|---|---|---|---|
| $c_1$ | 26 | 26 | 26 |
| $c_2$ | 10 | 10 | — |
| $c_3$ | 9 | — | — |
| $c_4$ | 15 | 24 | 30 |

(b) Tallies after each round of counting

# Margin of Victory

- By *how much* did winner $w \in \mathcal{C}$ win?
- **Definition (Last Round Margin):** *The difference between the tallies of the last two remaining candidates, divided by 2 and rounded up.*

# Margin of Victory

- By *how much* did winner $w \in \mathcal{C}$ win?
- **Definition (Last Round Margin):**
  *The difference between the tallies of the last two remaining candidates, divided by 2 and rounded up.*
- **Definition (Margin of Victory):**
  *The number of votes we would have to* **manipulate** *to ensure that a candidate other than $w$ wins the election.*
- A manipulation *changes* the ranking of a vote
  e.g., a vote with ranking $[c_1, c_2, c_4]$ changed to $[c_2, c_3]$

# Margin of Victory – An Example

**Last Round Margin** is 2 votes

| Ranking | Count |
|---|---|
| $[c_2, c_3]$ | 4 |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| $[c_2, c_3, c_4]$ | 6 |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

(c) Initial tallies

| Candidate | Rnd1 | Rnd2 | Rnd3 |
|---|---|---|---|
| $c_1$ | 26 | 26 | 26 |
| $c_2$ | 10 | 10 | — |
| $c_3$ | 9 | — | — |
| $c_4$ | 15 | 24 | 30 |

(d) Tallies after each round of counting

# Margin of Victory – An Example

**Margin of Victory** is 1 vote! Change one $[c_2, c_3]$ vote to $[c_3, c_4]$

| Ranking | Count |
|---|---|
| $[c_2, c_3]$ | **3** |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | **10** |
| $[c_2, c_3, c_4]$ | 6 |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

(e) Initial tallies

| Candidate | Rnd1 | Rnd2 | Rnd3 |
|---|---|---|---|
| $c_1$ | 26 | 26 | 41 |
| $c_2$ | 9 | — | — |
| $c_3$ | 10 | 19 | 19 |
| $c_4$ | 15 | 15 | — |

(f) Tallies after each round of counting

# Distance to a particular order

- Treat the outcome of an IRV election as a *candidate elimination order* $(\pi)$
  e.g., $\pi = [c_3, c_2, c_1, c_4]$

# Distance to a particular order

- Treat the outcome of an IRV election as a *candidate elimination order* ($\pi$)
  e.g., $\pi = [c_3, c_2, c_1, c_4]$
- Given candidates $\mathcal{C}$, votes $\mathcal{B}$, and candidate order $\pi$, and all the votes, a linear program (LP) [$\textsc{DistanceTo}$] computes the smallest manipulation of $\mathcal{B}$ required to achieve $\pi$

# A simple (but very inefficient) algorithm

- Compute the last-round margin $M_{\text{Last}}$
- $M = M_{\text{Last}}$
- For every candidate elimination order $\pi$,
  - If $\textsc{DistanceTo}(\pi) < M$
  - Update $M$

This works, but you have to check all $n!$ elimination orders, where $n$ is the number of candidates.

# Computing Margins – Existing Work

Algorithm by Magrino, Rivest, Shen and Wagner (2011) [MRSW]

- Search the space of *alternate orders* (ending in a different winner) for one requiring the *least manipulation*
- Applies branch and bound to the tree of elimination orders

# Computing Margins – Existing Work

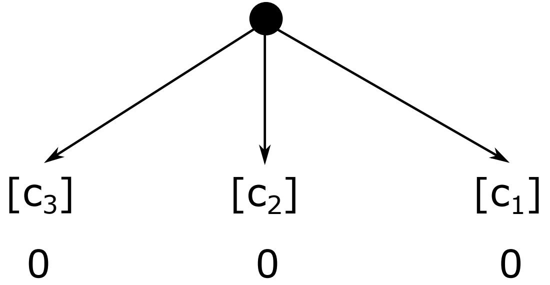
**Key observation:**
Apply DistanceTo LP to *partial order* $\pi'$ to get *lower bound* on manipulations required to achieve any order *ending in* $\pi'$

Candidates $c_1$, $c_2$, $c_3$, and winner $c_4$.

# Computing Margins – Existing Work

**Key observation:**
Apply DistanceTo LP to *partial order* $\pi'$ to get *lower bound* on manipulations required to achieve any order *ending in* $\pi'$

Candidates $c_1$, $c_2$, $c_3$, and winner $c_4$.

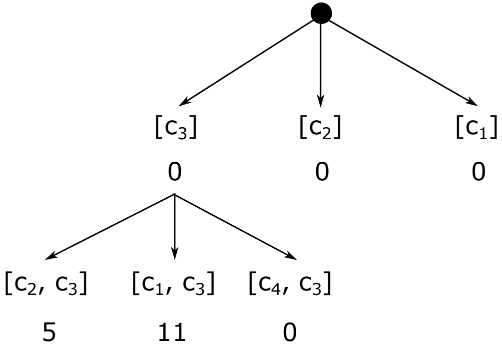



Add nodes to represent each possible *alternate winner*

# Computing Margins – Existing Work

**Key observation:**
Apply DistanceTo LP to *partial order* $\pi'$ to get *lower bound* on manipulations required to achieve any order *ending in* $\pi'$

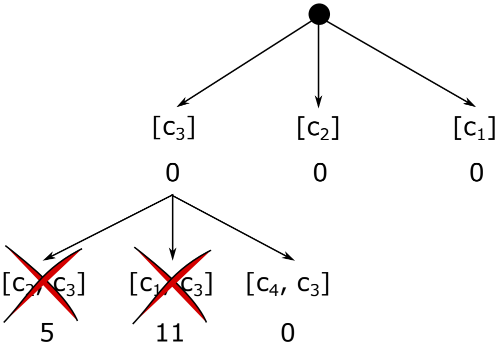
Candidates $c_1$, $c_2$, $c_3$, and winner $c_4$.



# Computing Margins – Existing Work

**Key observation:**
Apply DistanceTo LP to *partial order* $\pi'$ to get *lower bound* on manipulations required to achieve any order *ending in* $\pi'$

Candidates $c_1$, $c_2$, $c_3$, and winner $c_4$. Initial upper bound of 2.



# Improvements (BST, ECAI '17)

- Two new rules for computing lower bounds for each partial order
  - Lower bounds are typically *tighter* than those of DistanceTo
- Reduces the number of orders explored and LPs solved
- Compute margins in elections for which MRSW times out

# New Lower Bounding Rule

Consider a partial order $\pi' \subset \mathcal{C}$ (i.e., an order ending in $\pi'$)

- All candidates $e \in \mathcal{C} \setminus \pi'$ must be eliminated prior to those in $\pi'$

# New Lower Bounding Rule

Consider a partial order $\pi' \subset \mathcal{C}$ (i.e., an order ending in $\pi'$)

- All candidates $e \in \mathcal{C} \setminus \pi'$ must be eliminated prior to those in $\pi'$
- Let $\Delta_{\mathcal{S}}(c, e)$ denote the number of votes where $c \in \pi'$ is ranked higher than $\{e\} \cup \pi' \setminus \{c\}$ (or $c$ appears and the others do not)

# New Lower Bounding Rule

Consider a partial order $\pi' \subset \mathcal{C}$ (i.e., an order ending in $\pi'$)

- All candidates $e \in \mathcal{C} \setminus \pi'$ must be eliminated prior to those in $\pi'$
- Let $\Delta_{\mathcal{S}}(c, e)$ denote the number of votes where $c \in \pi'$ is ranked higher than $\{e\} \cup \pi' \setminus \{c\}$ (or $c$ appears and the others do not)
- Let $f(c)$ denote the number of votes in which $c$ is ranked first

# New Lower Bounding Rule

Consider a partial order $\pi' \subset \mathcal{C}$ (i.e., an order ending in $\pi'$)

- All candidates $e \in \mathcal{C} \setminus \pi'$ must be eliminated prior to those in $\pi'$
- Let $\Delta_{\mathcal{S}}(c, e)$ denote the number of votes where $c \in \pi'$ is ranked higher than $\{e\} \cup \pi' \setminus \{c\}$ (or $c$ appears and the others do not)
- Let $f(c)$ denote the number of votes in which $c$ is ranked first
- To ensure $e \in \mathcal{C} \setminus \pi'$ is eliminated before all $c \in \pi'$, we require:

$$\Delta_{\mathcal{S}}(c, e) \geq f(e)$$

# New Lower Bounding Rule

Consider a partial order $\pi' \subset \mathcal{C}$ (i.e., an order ending in $\pi'$)

- All candidates $e \in \mathcal{C} \setminus \pi'$ must be eliminated prior to those in $\pi'$
- Let $\Delta_{\mathcal{S}}(c, e)$ denote the number of votes where $c \in \pi'$ is ranked higher than $\{e\} \cup \pi' \setminus \{c\}$ (or $c$ appears and the others do not)
- Let $f(c)$ denote the number of votes in which $c$ is ranked first
- To ensure $e \in \mathcal{C} \setminus \pi'$ is eliminated before all $c \in \pi'$, we require:

$$\Delta_{\mathcal{S}}(c, e) \geq f(e)$$

$$l(c, e) = \max(\ 0,\ \lceil\ \tfrac{f(e) - \Delta_{\mathcal{S}}(c,e)}{2}\ \rceil\ )$$

# New Lower Bounding Rule

Consider a partial order $\pi' \subset \mathcal{C}$ (i.e., an order ending in $\pi'$)

- All candidates $e \in \mathcal{C} \setminus \pi'$ must be eliminated prior to those in $\pi'$
- Let $\Delta_{\mathcal{S}}(c, e)$ denote the number of votes where $c \in \pi'$ is ranked higher than $\{e\} \cup \pi' \setminus \{c\}$ (or $c$ appears and the others do not)
- Let $f(c)$ denote the number of votes in which $c$ is ranked first
- To ensure $e \in \mathcal{C} \setminus \pi'$ is eliminated before all $c \in \pi'$, we require:

$$\Delta_{\mathcal{S}}(c, e) \geq f(e)$$

$$l(c, e) = \max(\ 0,\ \lceil\ \tfrac{f(e) - \Delta_{\mathcal{S}}(c,e)}{2}\ \rceil\ )$$

$$lbound(\pi') = \max\{l(c, e) \mid c \in \pi',\ e \in \mathcal{C} \setminus \pi'\}$$

# New Lower Bounding Rule

Consider a partial order $\pi' = [c_2]$ (i.e., an order ending in $c_2$)

| Ranking | Count |
|---|---|
| **$[\mathbf{c_2}, \mathbf{c_3}]$** | **4** |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| **$[\mathbf{c_2}, \mathbf{c_3}, \mathbf{c_4}]$** | **6** |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

$$\Delta_{\mathcal{S}}(c_2, c_1) = 10$$

# New Lower Bounding Rule

Consider a partial order $\pi' = [c_2]$ (i.e., an order ending in $c_2$)

| Ranking | Count |
|---|---|
| **[c₂, c₃]** | **4** |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| **$[c_2, c_3, c_4]$** | **6** |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

$$\Delta_{\mathcal{S}}(c_2, c_1) = 10$$
$$\Delta_{\mathcal{S}}(c_2, c_3) = 10$$
$$\Delta_{\mathcal{S}}(c_2, c_4) = 10$$

# New Lower Bounding Rule

Consider a partial order $\pi' = [c_2]$ (i.e., an order ending in $c_2$)

| Ranking | Count |
|---|---|
| **$[\mathbf{c_2}, \mathbf{c_3}]$** | **4** |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| **$[\mathbf{c_2}, \mathbf{c_3}, \mathbf{c_4}]$** | **6** |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

$$\Delta_{\mathcal{S}}(c_2, c_1) = 10 \quad f(c_1) = 26$$
$$\Delta_{\mathcal{S}}(c_2, c_3) = 10 \quad f(c_3) = 9$$
$$\Delta_{\mathcal{S}}(c_2, c_4) = 10 \quad f(c_4) = 15$$

# New Lower Bounding Rule

Consider a partial order $\pi' = [c_2]$ (i.e., an order ending in $c_2$)

| Ranking | Count |
|---|---|
| **$[\mathbf{c_2}, \mathbf{c_3}]$** | **4** |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| **$[\mathbf{c_2}, \mathbf{c_3}, \mathbf{c_4}]$** | **6** |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

$$\Delta_{\mathcal{S}}(c_2, c_1) = 10 \quad f(c_1) = 26$$
$$\Delta_{\mathcal{S}}(c_2, c_3) = 10 \quad f(c_3) = 9$$
$$\Delta_{\mathcal{S}}(c_2, c_4) = 10 \quad f(c_4) = 15$$

$$l(c_2, c_1) = \max(0, \lceil \tfrac{f(c_1) - \Delta_{\mathcal{S}}(c_2, c_1)}{2} \rceil) = 8$$
$$l(c_2, c_3) = 0$$
$$l(c_2, c_4) = 3$$

# New Lower Bounding Rule

Consider a partial order $\pi' = [c_2]$ (i.e., an order ending in $c_2$)

| Ranking | Count |
|---|---|
| **[c₂, c₃]** | **4** |
| $[c_1]$ | 20 |
| $[c_3, c_4]$ | 9 |
| **[c₂, c₃, c₄]** | **6** |
| $[c_4, c_1, c_2]$ | 15 |
| $[c_1, c_3]$ | 6 |

$$\Delta_{\mathcal{S}}(c_2, c_1) = 10 \quad f(c_1) = 26$$
$$\Delta_{\mathcal{S}}(c_2, c_3) = 10 \quad f(c_3) = 9$$
$$\Delta_{\mathcal{S}}(c_2, c_4) = 10 \quad f(c_4) = 15$$

$$l(c_2, c_1) = \max(0, \lceil \tfrac{f(c_1) - \Delta_{\mathcal{S}}(c_2, c_1)}{2} \rceil) = 8$$
$$l(c_2, c_3) = 0$$
$$l(c_2, c_4) = 3$$

$$bound(\pi') = \max\{8, 0, 3\} = 8$$

DistanceTo LP gives a lower bound of 0 for $\pi' = [c_2]$!
We take the maximum of the LP and lower bounding rule result

# This paper - computing the overall parliamentary margin

*What is the total number of votes that need to change in order to change the overall Parliamentary election outcome?*

- In Australian Parliaments, each seat in the lower house is typically elected using IRV.
- This requires a modification to the algorithm: how many votes does it take to change *to a particular outcome?*
- Add up the margins for as many seats as need to change.

# Example: New South Wales state election, 2015

- 4.56 million votes
- 93 seats in Parliament
- Overall conservatives won 46% of first prefences and 54 seats
- 34% for Labor (34 seats), 10% Green (3 seats), 2 independents.
- It would take
  - 22,746 vote changes for the Labor party to gain the 13 additional seats they need to win government (with 47 seats),
  - 16,349 vote changes for a Labor/Greens coalition to gain the 10 additional seats they need to win government, and
  - 10,398 vote changes to lose the Liberal/National coalition 8 seats and hence produce a hung parliament.

**Questions?**