Mukesh Tiwari (University of Melbourne, Melbourne), Dirk Pattinson (Australian National University, Canberra)

Machine Checked Properties of Schulze Method

Ongoing Work

Developing Correct Software is Hard

Flaws found in NSW iVote system yet again

Analysis of source code published at the request of the NSW Electoral Commission shows that the state's election system software was still vulnerable to attack.

International Conference on Interactive Theorem Proving

ITP 2017: Interactive Theorem Proving pp 410-426 | Cite as

Schulze Voting as Evidence Carrying Computation

Authors | Authors and affiliations

Dirk Pattinson, Mukesh Tiwari

- In this ongoing work, we push the limit of correctness further by proving that our Coq implementation of Schulze method follows Condorcet Winner and Monotonicity property.

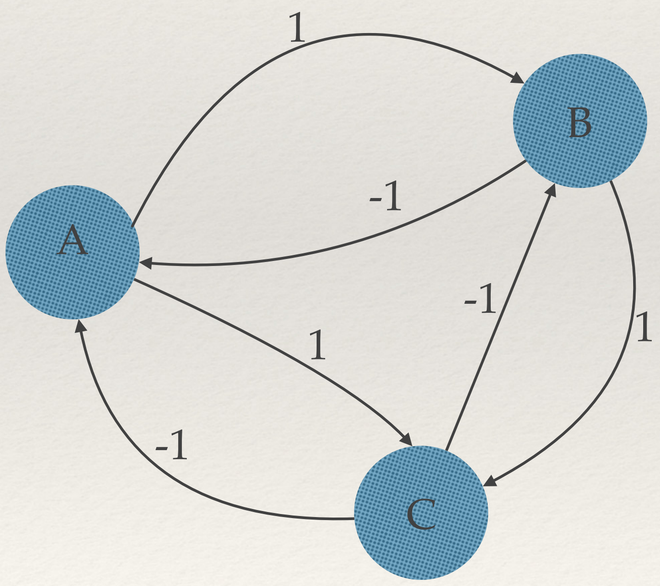
Schulze Method

A	1
B	2
C	3

Ballot

- Construct a margin function (matrix) $m : C \times C \to \mathbb{Z}$.

$$m(c, d) = \sharp\{b \in P \mid c >_b d\} - \sharp\{b \in P \mid d >_b c\}$$



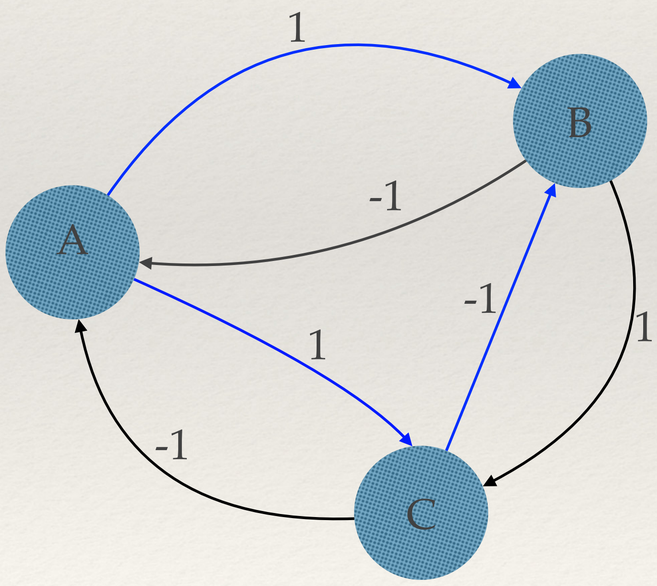
G

	A	B	C
A	0	1	1
B	-1	0	1
C	-1	-1	0

Margin Matrix (m)

- The strength, st, of a path in G is

$$st(c_0, \ldots, c_{n+1}) = \min\{m(c_i, c_{i+1}) \mid 0 \leq i \leq n\}.$$



G

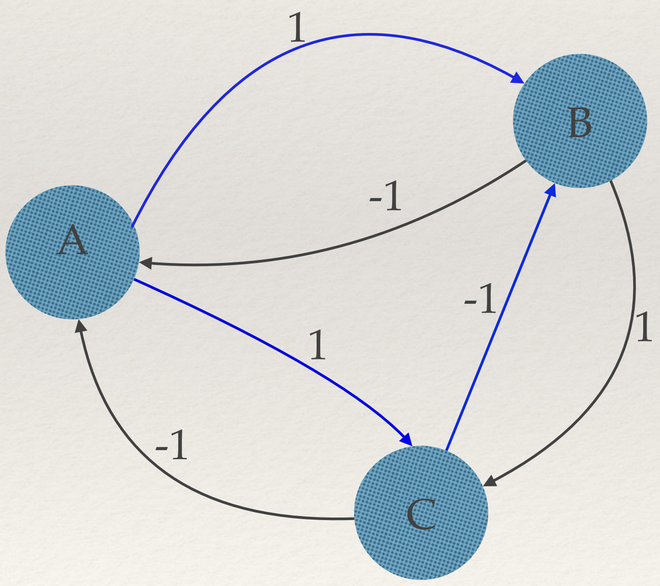
st (A - B) = min {m (A, B)}
= 1

st (A - C - B) = min {m (A, C), m (C, B)}
= min {1, -1}
= -1

- We compute the generalized margin, M, between two candidate c d as

$$M(c, d) = \max\{st(p) : \text{p is path from c to d in G}\}$$

M(A, B) = max {st (A, B), st (A-C-B)}
= max {1, -1}
= 1



G

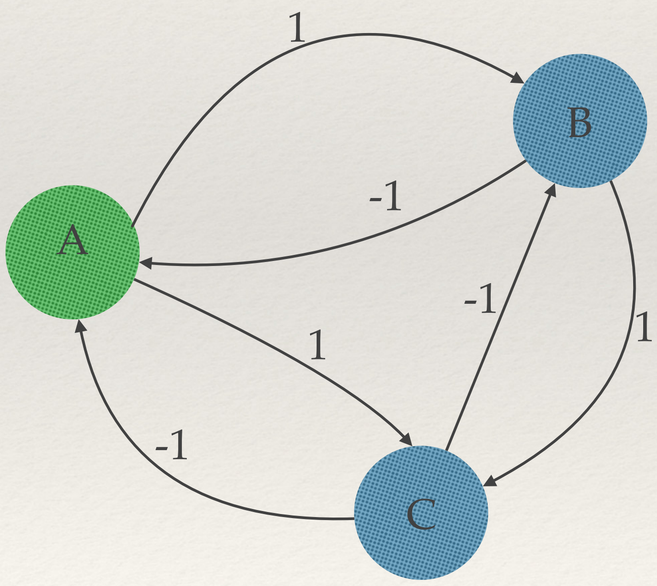
	A	B	C
A	0	1	1
B	-1	0	1
C	-1	-1	0

Generalised Margin Matrix (M)

- The winning set is defined as

$$W = \{c \in C : \forall d \in C \setminus \{c\}, M(c,d) \geq M(d,c)\}$$

A wins



G

	A	B	C
A	0	1	1
B	-1	0	1
C	-1	-1	0

Generalised Margin Matrix (M)

Condorcet Winner

If there is a Condorcet Winner, then Schulze Method elects it

```
(* if candidate c is condercet winner then it's winner of election *)
    Lemma condercet_winner_implies_winner (c : cand) (marg : cand -> cand -> Z) :
      condercet_winner marg c = true -> c_wins marg c = true.
    Proof.
      intros Hc.
```

A	1
B	2
C	3

Ballot

A	1
B	2

A beats B

A	1
C	3

A beats C

A is the Condorcet Winner

Monotonicity

```
Lemma winner_reversed :
  forall marg c, unique_winner marg c ->
           (exists d, c_wins marg d = false /\ c <> d) ->
           c_wins (rev_marg marg) c = false.
Proof.
```

A is the winner

A	1
B	2
C	3

Ballot A

	A	B	C
A	0	1	1
B	-1	0	1
C	-1	-1	0

Margin Matrix (m)

A is the loser

A	3
B	2
C	1

Reversed Ballot A

	A	B	C
A	0	-1	-1
B	1	0	-1
C	1	1	0

Reversed Margin Matrix (m)

Thank You