Universal Voting Protocol Tweaks to Make Manipulation Hard

BY V. CONITZER AND T. SANDHOLM

J. MARGOLIS, J. STERN, A. TAYLOR, A. RABINOLD
MARCH 4, 2015
A fundamental problem

- **Gibbard-Satterthwaite Theorem**: if there are 3+ candidates, then there are preferences under which an agent is better off voting strategically than according to true preference set.

- All nondictatorial protocols are manipulable
  - But one voter should not be able to dictate social choice regardless of other voters!

- In AI applications where voters are software agents:
  - Algorithm re-used by multitudes, rational strategic voting increases
  - Software agents have more computational power and can more easily find effective manipulations than humans can.
Rather than re-creating the wheel...

- ...why not add to the wheel?

- Proposal: tweak existing protocols by adding a single elimination preround

- Depending on preround schedule, manipulation becomes NP-hard, \#P-hard, or PSPACE-hard (definitions to come)

- Assuming that PSPACE \neq NP, manipulation is in a higher complexity class than NP.
BRUTE-FORCE SOLUTION:
\( O(n!) \)

DYNAMIC PROGRAMMING ALGORITHMS:
\( O(n^2 2^n) \)

SELLING ON EBAY:
\( O(1) \)

Still working on your route?

Shut the hell up.
Advantages of the Tweak

- Satisfiable for large family of election protocols (we will focus on plurality, Borda, STV, and Maximin)
- Leaves much of original protocol intact
- Easier to adjust existing protocol than to replace
New definitions

• Preround: protocol tweak in which candidates are matched in a pairwise election where only the winner continues to complete the rest of the election

• Schedule: determination of who faces who in the preround

• Bye: given to a candidate if there are an odd number of candidates in preround; candidate with a bye is never eliminated and automatically is in the running for the rest of the election

• Elicitation: collecting only parts of the voters’ preferences

• Maximin: election protocol in which a candidate’s number of points is the lowest score they receive in any pairwise election

• #P: The class of problems where the task is to count the number of solutions to a problem in NP, not just looking for if there are any

• PSPACE: the class of problems solvable in polynomial space
Preround

Definition 1: Given a protocol P, the new protocol obtained by adding a preround to it proceeds as follows:

1. Pair the candidates. If there is an odd number, one candidate gets a *bye*.
2. In each pairing, the loser of each pairwise election is eliminated (candidate with a bye is not eliminated).
3. On remaining candidates, execute P as usual to produce a winner.
Candidates

Dany  Margaery  Tyrion  Jon  Brienne
Preround

Votes:
- Dany > Jon > Brienne > Tyrion > Margaery
- Dany > Margaery > Jon > Brienne > Tyrion
- Margaery > Jon > Tyrion > Brienne > Dany
- Brienne > Dany > Margaery > Jon > Tyrion
- Tyrion > Dany > Brienne > Margaery > Jon

<table>
<thead>
<tr>
<th>Preround Match</th>
<th>Dany vs. Jon</th>
<th>Margaery vs. Brienne</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dany vs. Jon</td>
<td>Dany</td>
<td>Brienne</td>
</tr>
<tr>
<td>Margaery vs. Brienne</td>
<td>Dany</td>
<td>Margaery</td>
</tr>
<tr>
<td>Tyrion vs. bye</td>
<td>Jon</td>
<td>Margaery</td>
</tr>
</tbody>
</table>

Pairwise: Dany vs Jon, Margaery vs Brienne, Tyrion “vs” bye

Dany survives, Brienne survives, Tyrion survives
Maximin Example

Dany > Brienne > Tyrion
Dany > Brienne > Tyrion
Tyrion > Brienne > Dany
Brienne > Dany > Tyrion
Tyrion > Dany > Brienne

Winners: Tyrion, Brienne, Dany
Other victories

<table>
<thead>
<tr>
<th></th>
<th>Borda:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Dany 6; Brienne 5; Tyrion 4</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Plurality:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Dany, Tyrion</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>STV:</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Brienne knocked out; Dany victorious</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Dany</td>
<td>Brienne</td>
<td>Tyrion</td>
</tr>
<tr>
<td>Dany</td>
<td>Brienne</td>
<td>Tyrion</td>
</tr>
<tr>
<td>Tyrion</td>
<td>Brienne</td>
<td>Dany</td>
</tr>
<tr>
<td>Brienne</td>
<td>Dany</td>
<td>Tyrion</td>
</tr>
<tr>
<td>Tyrion</td>
<td>Dany</td>
<td>Brienne</td>
</tr>
</tbody>
</table>
Variations on the preround

• DPRE+P: protocol with preround where schedule is determined and published before votes are collected

• RPRE+P: protocol with preround where schedule is drawn randomly after votes are collected

• IPRE+P: protocol with preround where votes are elicited incrementally, interleaved with scheduling-and-publishing of the various face-offs
Deterministic Preround (DPRE+P)

**Theorem 1:** Given a voting protocol $P$, suppose that it is possible, for any Boolean formula $\phi$ in conjunctive normal form (i.e. a SAT instance) to construct in polynomial time a set of votes over a candidate set containing at least $\{p\} \cup C_L$ where $C_L = \{c_l : l \in L\}$ ($L$ is the set of literals $\{+v : v \in V\} \cup \{-v : v \in V\}$, where $V$ is the set of variables used in $\phi$), with the following properties:

1. If we remove, for each $v \in V$, one of $c_{+v}$ and $c_{-v}$, $p$ would win an election under protocol $P$ against the remaining candidates if and only if for every clause $k \in K$ ($K$: set of clauses in $\phi$), there is some $l \in L$ such that $c_l$ has not been removed, and $l$ occurs in $k$. This should hold even if a single arbitrary vote is added.

2. For any $v \in V$, $c_{+v}$ and $c_{-v}$ are tied in their pairwise election after these votes.
Plurality

- Candidate set $C$ includes required candidates from property 1 plus set of candidates corresponding to the set of clauses $K$ of $\phi$, $C_K = \{c_k \mid k \in K\}$

- Regardless to which of the candidates corresponding to literals are removed, p with get $4|K| + 2$ votes.

- Two potential cases:
  - If for some $k \in K$, all candidates $c_l$ with $l \in L$, $l \in k$ are removed, then $c_k$ receives $4|K|+4$ votes; p does not win
  - If for each $k \in K$, at least one candidate $c_l$ with $l \in k$ remains, then each $c_k$ will receive $4|K|$ votes, as will each remaining $c_l$ (at the most). Thus, p wins.
Borda

• Let there be M candidates.

• Set of votes: for every $c_i \in C_k$
  - 4M votes ranking $c_{i+1} > c_{i+2} > \ldots > c_{|K|} > p > c_1 > c_2 > \ldots > c_{i-1} > \{l \text{ occurs in the clause corresponding to } c_i\} > c_i > \{c_j \in L : j \neq c_i\}$
  - 4M votes ranking $c_1 > c_2 > \ldots > c_{|K|} > p > C_L$
  - 1 vote $c_1 > c_2 > \ldots > c_{|K|} > C_L > p$
  - 1 vote $c_{|K|} > c_{|K|-1} > \ldots > c_1 > C_L > p$
  - $4|K| M$ votes $p > c_1 > c_2 > \ldots > c_n > C_L$
  - $4|K| M$ votes $c_n > c_{n-1} > \ldots > c_1 > p > C_L$
Borda cont.

• None of the $c_i$ can win (only consider $c_i$ and $p$)
  ◦ 4M votes ranking $c_{i+1} > c_{i+2} > ... > c_{|K|} > p > c_1 > c_2 > ... > c_{i-1} > c_i$
  ◦ 4M votes ranking $c_1 > c_2 > ... > c_{|K|} > p$
  ◦ 1 vote $c_1 > c_2 > ... > c_{|K|} > p$
  ◦ 1 vote $c_{|K|} > c_{|K|-1} > ... > c_1 > p$
  ◦ 4$|K|$ votes $p > c_1 > c_2 > ... > c_n$
  ◦ 4$|K|$ votes $c_n > c_{n-1} > ... > c_1 > p$

• After first votes, all $c_i$ with $i \in k$ have been removed tied with $p$, all other $c_k$ at least 4M points behind $p$

• For last two points, $p$ wins if and only if for every clause $k \in K$, there is some $l \in L$ with $l \in k$ such that $c_i$ has not been removed.

• A margin of at least $M - |V|$ points exists
Maximin

• Regardless of which of the candidates corresponding to literals is removed, p’s worst score in a pairwise election is when it faces any $c_k$ $(16|K| + 2)$

• Any $c_k$ where all the $c_l$ with $l \in k$ have been removed will get its worst score against any of the $C_l$ $(16|K| + 4)$

• Any other $c_k$ will get its worst score against some $c_l$ with $l \in k$ $(16|K|)$

• p wins if and only if for every clause $k \in K$, there exists some $l \in k$ such that $c_l$ has not been removed.
Randomized Preround (RPRE+P)

- Manipulation becomes \#P-hard
  - The class of problems where the task is to count the number of solutions to a problem in NP, not just looking for if there are any

- Reduction of an arbitrary PERMANENT instance to set of votes satisfying certain properties (to be stated)
PERMANENT problem

- Computing the permanent of a matrix is #P-hard
- Given bipartite graph B with the same number of vertices k in both parts
- Asked how many perfect matchings there are from vertices sets \{1, 2, \ldots, k\} and \{k+1, k+2, \ldots, 2k\}
Conditions for \#P-hardness

**Theorem 7:** Given a voting protocol P, suppose that it is possible for any bipartite graph B with the same number of vertices in both parts, to construct in polynomial time a set of votes over the candidate set \{c_1, \ldots, c_{2k}, p\} (where c_i corresponds to vertex i in B) with the following properties:

1. If we remove k of the c_i, p would win an election under protocol P against the remaining c_i if and only if the removed c_i are exactly all the c_i with k+1 ≤ i ≤ 2k
2. p loses its pairwise election against all c_i with k+1 ≤ i ≤ 2k
3. For any 1 ≤ i ≤ k and k+1 ≤ j ≤ 2k, c_i defeats c_j in their pairwise election if and only if in B, there is an edge between vertices i and j
4. All the previous properties still hold with any additional single vote
A single reduction

Given a bipartite graph B with the same number of vertices k in both parts, we have $12k^3 + 2k^2$ votes divided in the following ways:

- **$6k^3$ votes** ranking $c_{k+1} > c_{k+2} > ... > c_{2k} > p > c_1 > c_2 > ... > c_k$

- **$3k^2$ votes** ranking $p > c_k > c_{k-1} > ... > c_1 > c_{2k} > c_{2k-1} > ... > c_{k+1}$

- **$6k^3-3k^2$ votes** ranking $c_k > c_{k-1} > ... c_1 > c_{2k} > c_{2k-1} > ... > c_{k+1} > p$

For each edge $(i,j)$ in B ($1 \leq i \leq k$, $k+1 \leq j \leq 2k$):

- One vote ranking $c_i > c_j > p > c_1 > c_2 > ... > c_{i-1} > c_{i+1} > ... > c_k > c_{k+1} > ... > c_{j-1} > c_{j+1} > ... c_{2k}$

- One vote ranking the inverse (except $c_i$ and $c_j$ have maintained order)

For each pair $i,j$ without an edge between them in B ($1 \leq i \leq k$, $k+1 \leq j \leq 2k$):

- One vote ranking $c_j > c_i > p > c_1 > c_2 > ... > c_{i-1} > c_{i+1} > ... > c_k > c_{k+1} > ... > c_{j-1} > c_{j+1} > ... c_{2k}$

- One vote ranking the inverse (except $c_i$ and $c_j$ have maintained order)
Interleaved Preround (IPRE-P)

- Scheduling and eliciting votes simultaneously increases complexity of manipulation to PSPACE-hardness
  - PSPACE: the class of problems solvable in polynomial space

- Hardness carries over from the specific to the general, so using the precise mechanical process of interleaving is not always necessary to achieve hardness
Interleaved Preround: algorithm

1. Label the matchups (a space in the preround for two yet-to-be-determined candidates to face each other) 1 through $\frac{|C|}{2}$

2. For each matchup $i$, assign one of the candidates to play in it. Denote them by $c(i,1)$. One of the candidates in each matchup is now know.

3. For some $k$ which is a multiple of 4, for each $i$ with $1 \leq i \leq k$, assign second candidate to play in matchup $i$. Denote this candidate $c(i,2)$.

<table>
<thead>
<tr>
<th>$C(1,1)$</th>
<th>$C(1,2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C(2,1)$</td>
<td>$C(2,2)$</td>
</tr>
<tr>
<td>$C(3,1)$</td>
<td>$C(3,2)$</td>
</tr>
<tr>
<td>$C(4,1)$</td>
<td>$C(4,2)$</td>
</tr>
</tbody>
</table>
4. For each pair of matchups \((2i-1, 2i)\) with \(i > \frac{k}{2}\), assign two more candidates to face the candidates already in these two matchups. Denote them \(c((2i-1, 2i), 1)\) and \(c((2i-1, 2i), 2)\)

5. For \(i = \frac{k}{2} + 1\) to \(\frac{|C|}{4}\)
   • Randomly decide which of the new matchups faces \((2i-1, 1)\) and which faces \((2i, 1)\). Denote these new matchups \(c(2i-1, 2)\) and \(c(2i, 2)\), respectively
   • Ask all voters for preference: \(c(i - \frac{k}{2}, 1)\) or \(c(i - \frac{k}{2}, 2)\). Even if there are \(k=0\) scheduled matchups, elicitation process trails behind the scheduling process by a factor 2.

6. Elicit the remainder of all the votes
Applying IPRE+P

- Preconditions for generic IPRE remarkably similar to those presented in Theorem 1
  - Removing $c_{+v}$ or $c_{-v}$ from each $v \in V$ will ensure $p$ wins an election iff for every clause $k \in K$ there exists some $l \in L$ such that $c_l$ has not been removed and $l$ occurs in $k$ (even if single arbitrary vote added)
  - For any $x \in X$, $c_{+v}$ and $c_{-v}$ are tied in their pairwise elections after these votes
  - For any $y \in Y$, $c_y$ and $c_{-y}$ both lose their pairwise elections against $c_{1_y}$ by at least 2 votes

- Add in the $c_{1_y}$ such that corresponding $c_y$ and $c_{-y}$ in slightly more than half the votes and ranking them as low as possible elsewhere

- Otherwise, follow proofs corresponding to theorem 1
Questions raised

• We have only described situations in which there is one single voter trying to manipulate an election
Questions raised

Furthermore, we rely on the condition that all other votes are known to this single manipulator.
Questions raised

How would these protocols hold under someone who is eyeing towards destructive manipulation?

“Please, literally anybody else.”
- Traditional Westerosi ballot
Merits

- First results in voting settings that put manipulation in classes higher than NP
- Prerounds very easy to implement
- Still works with major voting protocols