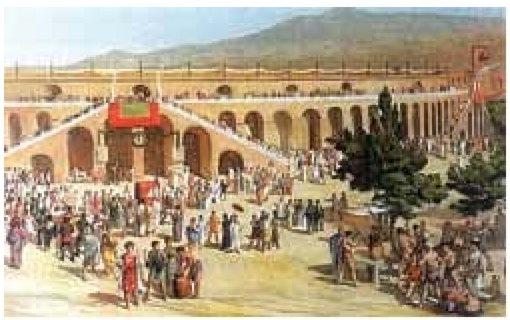
Complexity in elections

Dejan Novakovic
Game Theory and algorithms
EPFL, 2011

History ...

- Elections are important



. . .

Manipulations

Manipulations (examples)

- "Spendthrift election", 1768 (Northamptonshire)

- California gubernatorial election, 2002
  - Riordan/Simon/Davis

- Canadian general election, 2004
  - Liberal Party was able to convince many New Democratic voters to vote Liberal in order to avoid a Conservative government)

What should we do?

Outline

- Gibbard-Satterthwaite theorem
- Manipulations and Computational complexity
- Bartholdi's rules
  - Computational resistance
- More of the complexity
- Concerns

Gibbard–Satterthwaite theorem 1/3

- Recall Arrow's theorem:
  - Claim 1: Unanimity Preserving (if X>Y and candidate X is preferred by all voters then Y is not selected)
  - Claim 2: Independent Irrelevant Alternatives (If every voter's preference between X and Y remains unchanged, then the group's preference between X and Y will also remain unchanged (even if voters' preferences between other pairs like X and Z, Y and Z, or Z and W change).)

  - Arrow's Theorem: If players are truthful, claims 1 and 2 are only satisfied in a dictatorial system!

Gibbard–Satterthwaite theorem 2/3

- Theorem (GS): For three or more candidates, one of the following three things must hold for every voting rule:
  - The rule is dictatorial.
  - There is some candidate who can never win, under the rule.
  - The rule is susceptible to tactical voting.

Gibbard–Satterthwaite theorem 3/3

- Properties:
  - Candidate cannot win ☒
  - Dictatorial rule ☒
  - All voting systems which yield a single winner either are manipulable or do not meet the preconditions of the theorem.

- Conclusion:
  - Manipulation is always possible. There is no perfect voting system.

How hard is it, computationally, to find a manipulating ballot?

If it is computationally intractable to actually find out how to vote in order to manipulate successfully, then this may be deemed an acceptable risk!

Computational complexity

- An algorithm is considered formally efficient if it requires a number of computational steps that is at most polynomial in the size of the problem.

- If we find a polynomial time algorithm for any NP-complete problem, then we find for all (TSP, 3d matching ...).

Complexity of Manipulation in Voting

- Bartholdi, Tovey and Trick (1989)
  - "... manipulation is in fact easy for a range of commonly used voting rules ..."
  - Given: A set of candidates C, a set V of nonmanipulative voters, a set R of manipulative voters with $V \cap R = \emptyset$, and a distinguished member c in C.
  - Question: Is there a way to set the preference lists of the voters in R such that, under election system S, c is a winner of election $(C, V \cup R)$?
  - If this can be answered within polynomial time, then the voting scheme is ''easily manipulable''.

Manipulating the Plurality Rule

- Recall:
  - Each voter submits a ballot with exactly one name. The candidate receiving the most votes wins.

- It's easy to manipulate (trivial):
  - Vote for w (w is the candidate to be made winner by means of manipulation). If manipulation is possible at all, this will work.

- It can be done in polynomial time.

Manipulating the Borda Rule

- Recall:
  - complete ranking of all k candidates
  - calculate points based on ordering
  - add up the points and rank the candidates

- It's easy to manipulate as well (greedy algorithm):
  - put w at the top of your declared preference ordering.
  - check if any of the remaining candidates can be put next into the preference ordering without preventing w from winning.
  - If yes, do so. If no, terminate and say that manipulation is impossible.

- It can be done in polynomial time.

Greedy-Manipulation 1/2

- Theorem (Bartholdi, 1989):
  - Greedy-Manipulation will find a preference order **P** that will make **c** a winner (or conclude that it is impossible) for any voting scheme that can represented as function $S(P) = C \to R$ that is both:
    - responsive: a candidate with the largest $S(P, i)$ is a winner
    - monotone: for any two preference orders P and P' and for any candidate i, $\{j: iP'j\} \subseteq \{j: iPj\}$ implies that $S(P', i) \leq S(P, i)$.
- Corollary:
  - Any voting system that satisfies the conditions of Theorem, and for which **S** is evaluatable in polynomial time, can be manipulated in polynomial time.

Greedy-Manipulation 2/2

- Sketch proof (Corollary):
  - Greedy-Manipulation executes within polynomial time since no more than **n** iterations are required, and each iteration requires no more than **n** evaluations of **S** (by monotonicity of **S**) with each evaluation of **S** requiring only polynomial time (by assumption).

- Examples:
  – Plurality: $S(P, i) = b_i + 1$ if **i** is ranked 1st, else $= b_i$
  – Borda: $S(P, i) = b_i + |\{j : iPj\}| + 1$

Computational Resistance

- Recall
  - Gibbard–Satterthwaite

- <u>Claim</u>:
  - There exists a voting scheme that is simultaneously:
    - single valued
    - non-dictatorial
    - easy to compute, but computationally difficult to manipulate.

Single Transferable Vote (STV)

- Description:
  - Voters submit ranked preferences for all candidates.
  - If one of the candidates is the 1st choice for over 50% of the voters, he wins.
  - Otherwise, the candidate who is ranked 1st by the fewest voters ("plurality loser") gets eliminated from the race.
  - Votes for eliminated candidates get transferred:
    - delete removed candidates from ballots and "shift" rankings (e.g. if 1st choice got eliminated, then 2nd choice becomes 1st).

- In practice (Ireland, Malta, Australia, Canada, Cambridge …)

Intractability of Manipulating STV 1/2

- Bartholdi 1991: Manipulation of STV for electing a single winner is **NP**-complete.

- Proof sketch:
  - We need to prove NP-membership and NP-hardness.
    - winner determination can be done in polynomial time (# of rounds is limited)
    - if someone guesses a preference ordering to be used for manipulation, we only need to run the polynomial winner determination algorithm to check whether it worked

Intractability of Manipulating STV 2/2

- NP-hardness (3-Cover):
  - Instance: Sets $S_1, \dots, S_m$ with $|S_i| = 3$; $S = \bigcup_{i=1}^{m} S_i$ with $|S| = n$.
  - Question: Is there an $I \subseteq \{1..m\}$ with $|I| = n/3$ and $\bigcup_{i \in I} S_i = S$?

  - The proof for NP-hardness works by reducing 3-C to the former: Given any instance of 3-C, we can construct an election which a manipulator can manipulate successfully iff he can solve the 3-C problem.
    - First define a long list of voter preferences
    - Make sure the one we don't want to win does not gain transferred votes
    - This induces complex relationships between entries in the manipulator's ranking
    - This turns out to correspond to 3-Cover (see paper for details).

More on the Complexity of Voting

- What else is there:
  - winner determination
  - bribery
  - controlling an election

Winner determination

- For a given voting rule, what is the complexity of computing the winner?
- Why is this important?

- Description (Dodgson rule):
  - A Dodgson winner is a candidate minimising the number of “switches” in the voters’ linear preference orderings required to make that candidate a Condorcet winner.

- Checking whether a candidate's is a winner is NP-complete.

Bribery

- similar to manipulation
- constructive/destructive
- outside agent

- <u>Given</u>: A set **C** of candidates, a set **V** of voters, a distinguished candidate **c** in **C**, and a nonnegative integer **k**.

- <u>Question</u>: Is it possible to change the preference lists of at most **k** voters such that, under election system **S**, **c** is a winner of election **(C,V)**?

Control

- changing the structure
- adding or removing (candidates or voters)
- constructive/destructive

- Given: A set $\mathbf{C}$ of original candidates, a pool $\mathbf{D}$ of potential additional candidates, a distinguished candidate $\mathbf{c} \in \mathbf{C}$, and a set $\mathbf{V}$ of voters with preferences over $\mathbf{C} \cup \mathbf{D}$.
- Question: Is there a set $\mathbf{D}' \subseteq \mathbf{D}$ such that, under election system $\mathbf{S}$, $\mathbf{c} \in \mathbf{C}$ is not a winner of the election having candidates $\mathbf{C} \cup \mathbf{D}'$ with the voters being $\mathbf{V}$ with the preferences of $\mathbf{V}$ restricted to $\mathbf{C} \cup \mathbf{D}'$?

Concerns

- Manipulation is not an issue (complete information about **P** is never available).

- Very few among all are actually manipulable.

- Effective heuristic to manipulate an election even though manipulation is **NP**-complete.

Summary

- Complexity results:
  - winner determination, manipulation, bribery, control
  - winner determination should be computationally easy
  - for the manipulation, bribery, and control problems, intractability results are positive results.

References

- J.J. Bartholdi III, C.A. Tovey, and M.A. Trick.
  The Computational Difficulty of Manipulating an Election.
  Social Choice and Welfare, 1989

- J.J. Bartholdi III, and J.B. Orlin.
  Single Transferable Vote Resists Strategic Voting.
  Social Choice and Welfare, 1991

- P. Faliszewski, E. Hemaspaandra, L.A. Hemaspaandra, and J. Rothe.
  A Richer Understanding of the Complexity of Election Systems.
  Tech. Report, Dept. of CS, Universety of Rochester, 2006