Voting and Complexity

Rob LeGrand
legrand@cse.wustl.edu

Oral Qualifying Examination
28 September 2005

Washington
University in St. Louis

Outline

- Introduction
- Hardness of finding the winner(s)
  - Polynomial systems
  - NP-hard systems
    - The minimax procedure [Brams *et al.*]
- Hardness of voter manipulation
  - What is manipulation?
  - Polynomial systems
  - NP-hard systems
    - Second-order Copeland [Bartholdi *et al.*]
  - Tweaks to make manipulation NP-hard [Conitzer and Sandholm]
- Approximating minimax [Gąsieniec *et al.*]

Introduction: Computer science and voting

How can computer science improve the quality of elections?

- Common view: computers . . .
  - automate tedious counting
  - increase accuracy and reliability
  - reduce/eliminate spoiled ballots
- Computational view: CS makes possible new analysis of election systems
  - measure hardness of finding the winner(s)
  - measure hardness of manipulation by voters

Outline

- Introduction
- Hardness of finding the winner(s)
  - Polynomial systems
  - NP-hard systems
    - The minimax procedure
- Hardness of voter manipulation
  - What is manipulation?
  - Polynomial systems
  - NP-hard systems
    - Second-order Copeland
  - Tweaks to make manipulation NP-hard
- Approximating minimax

Easy for some election systems

Single-winner systems using simple ballots
(k alternatives, n voters)

- Plurality (first-past-the-post)
  - vote for one alternative, one with most votes wins
  - finding winner takes $O(k+n)$ time
- Approval voting
  - vote for up to k alternatives, one with most votes wins
  - finding winner takes $O(kn)$ time

Easy for some election systems

Single-winner systems using ranked ballots

- Borda
  - give $k-1$ points to one alternative, $k-2$ to another, and so on down to 0 for last
  - one with most points wins
  - finding winner takes $O(kn)$ time
- Copeland
  - rank all alternatives
  - one with highest Copeland score (pairwise victories minus pairwise defeats) wins
  - finding winner takes $O(k^2n)$ time

Easy for some election systems

Multiwinner systems
(k alternatives, m winners, n voters)

- Single non-transferable vote (SNTV)
  - vote for one alternative, m with most votes win
  - finding winners takes $O(km + n)$ time
- Single transferable vote (STV)
  - rank all alternatives
  - m winners found by quota/elimination scheme
  - finding winners takes $O(k^2n)$ time

Hard for some election systems

- Dodgson's method (single-winner)
  - rank all alternatives
  - winner is the alternative that requires fewest pairwise swaps among the ranked ballots to become Condorcet winner
  - finding winner is NP-hard [Bartholdi *et al.*]
- Brams *et al.*'s minimax procedure (multiwinner)
  - vote for up to k alternatives
  - winner set is that which has smallest maximum distance over all ballots
  - finding winners is NP-hard [Frances and Litman]

Minimax: Approval ballots

Approval ballot example: **010101**

- Voter approves three out of six alternatives (b, d, f)
- Voter's most preferred outcome: **010101** ($\{b, d, f\}$)
- Voter's least preferred outcome: **101010** ($\{a, c, e\}$)
- Voter prefers outcomes with smaller Hamming distances from **010101**
- Voter is indifferent among outcomes with equal Hamming distances from **010101**, *e.g.* **000000** and **111111**

Minimax: Hamming distance

- Used as measure of disagreement between a ballot and winner set
- Hamming distance between two sets S and T:

$$d_H(S, T) = |S - T| + |T - S|$$

- $d_H(\{a, b\}, \{a, c\}) = |\{b\}| + |\{c\}| = 2$
- Hamming distance between two bitstrings S and T:

$$d_H(S, T) = |S \oplus T|$$

- $d_H(\mathbf{010101}, \mathbf{111000}) = |\mathbf{101101}| = 4$

The minimax procedure

[Brams *et al.*]

- Finds a winner set that minimizes the dissatisfaction of the least satisfied voters
- Equivalent to choosing the winner set W with minimal "maxscore"
  - maxscore of a set is the largest Hamming distance between the set and any ballot:

$$\mathrm{maxscore}(S) = \max_{b \in B} d_H(S, b)$$

Minimax example

b_1	**000011**	$\{e, f\}$
b_2	**000111**	$\{d, e, f\}$
b_3	**001011**	$\{c, e, f\}$
b_4	**010011**	$\{b, e, f\}$
b_5	**111100**	$\{a, b, c, d\}$
W	**011111**	$\{b, c, d, e, f\}$

- All voters are relatively satisfied with the minimax outcome
- $\text{maxscore}(W) = 3$; all other sets have maxscore at least 4

Outline

- Introduction
- Hardness of finding the winner(s)
  - Polynomial systems
  - NP-hard systems
    - The minimax procedure
- Hardness of voter manipulation
  - What is manipulation?
  - Polynomial systems
  - NP-hard systems
    - Second-order Copeland
  - Tweaks to make manipulation NP-hard
- Approximating minimax

Can insincere voters manipulate?

Sincere ordinal preferences:

	7 voters	2 voters	6 voters
1st choice	a_1	a_2	a_3
2nd choice	a_2	a_3	a_2
3rd choice	a_3	a_1	a_1

- Under plurality voting, a_1 wins with 7 votes when all are sincere
- If a_2 voters voted for a_3 instead, a_3, their second choice, would win
- They can improve the outcome from their point of view by voting insincerely

Manipulation by insincere voters

- According to Gibbard and Satterthwaite, all election systems I discuss are sometimes vulnerable to manipulation by such insincere voting when $k \geq 3$
- General problem: Given the ballots of the other $n - 1$ voters, find the ballot (sincere or not) that will maximize your satisfaction with the result
- Another formulation: Given the ballots of the other $n - 1$ voters, find a ballot (if possible) that will elect a given alternative a

Manipulating minimax

Sincere votes:

b_1	**000011**	$\{e, f\}$
b_2	**000111**	$\{d, e, f\}$
b_3	**001011**	$\{c, e, f\}$
b_4	**010011**	$\{b, e, f\}$
b_5	**011111**	$\{b, c, d, e, f\}$
W_1	**000111**	$\{d, e, f\}$
W_2	**001011**	$\{c, e, f\}$
W_3	**010011**	$\{b, e, f\}$

- All voters approve e and f and disapprove a
- Voter 5 has Hamming distance 2 from each minimax winner set

Manipulating minimax

voter 5 is unscrupulous:

b_1	**000011**	$\{e, f\}$
b_2	**000111**	$\{d, e, f\}$
b_3	**001011**	$\{c, e, f\}$
b_4	**010011**	$\{b, e, f\}$
b_5	**111100**	$\{a, b, c, d\}$
W	**011111**	$\{b, c, d, e, f\}$

- By voting insincerely, voter 5 has manipulated the election to give his most preferred outcome decisively

Easy for some election systems

Single-winner systems

(k alternatives, n voters)

- Plurality (first-past-the-post)
  - vote for one alternative, one with most votes wins
  - finding most effective ballot takes $O(k+n)$ time
- Approval voting
  - vote for up to k alternatives, one with most votes wins
  - finding most effective ballot takes $O(kn)$ time
- Borda
  - assign points to alternatives based on ranked ballots
  - one with most points wins
  - finding most effective ballot takes $O(k^2+kn)$ time

Hard for some election systems

- Second-order Copeland [Bartholdi *et al.*]
  - rank all alternatives
  - winner is that whose defeated competitors have the largest sum of Copeland scores
  - finding most effective ballot is NP-hard
- Single transferable vote (STV)
  - rank all alternatives
  - m winners found by quota/elimination scheme
  - finding most effective ballot is NP-hard [Bartholdi and Orlin]
- Brams *et al.*'s minimax?
  - not proved, but NP-hard to find winners—perhaps same for manipulation

Manipulation decision problem

EXISTENCE OF A WINNING PREFERENCE (EWP)

INSTANCE: Set A and a distinguished member a of A; set B of transitive preference orders on A.
QUESTION: Does there exist a preference order b_0 on A such that a wins according to the election system with $B \cup \{b_0\}$?

- Assumes an election system that takes a set of preference orders and returns a winning alternative
- Alternatives A, ballots B; $|A| = k$, $|B| = n$

Greedy-Manipulation algorithm

[Bartholdi *et al.*]

Input preferences of all other voters; a distinguished alternative a

Output either a preference order that will elect a or a claim that none exists

Initialization Place a at the top of the preference order.

Iterative step Determine whether any alternative can be placed in the next lower position without preventing a from winning. If so, place such an alternative in the next position; otherwise terminate claiming that a cannot win.

Greedy-Manipulation algorithm (cont.)

- Poly-time algorithm to find a preference order that will elect a given alternative [Bartholdi *et al.*]
- Can be used to show that plurality, Borda and Copeland are manipulable in polynomial time
- Will work for any single-winner ranked-ballot election system that is "responsive and monotone"

Second-order Copeland

- Rank all alternatives
- Winner is that whose defeated competitors have the largest sum of Copeland scores (pairwise victories minus pairwise defeats)
- Greedy-Manipulation algorithm doesn't work (method fails monotonicity, unlike regular Copeland)
- Can elect nonintuitive winners

Second-order Copeland example

- a_1 pairwise defeats a_2, a_3, a_4
- a_2 pairwise defeats a_3, a_4, a_5
- a_3 pairwise defeats a_4, a_5
- a_4 pairwise defeats a_5
- a_5 pairwise defeats a_1

- Copeland scores: a_1: 2, a_2: 2, a_3: 0, a_4: -2, a_5: -2
- 2nd-order Copeland scores: a_1: 0, a_2: -4, a_3: -4, a_4: -2, a_5: 2

Second-order Copeland (cont.)

- Finding most effective ballot is NP-hard [Bartholdi *et al.*]
  - problem stated graph-theoretically
  - proof is reduction from 3,4-SAT (exactly 3 different variables in each clause, each variable appears in exactly 4 clauses)
  - 3,4-SAT expression is satisfiable iff there is a way to make a win

Tweaks to make manipulation hard

- Copeland with 2nd-order Copeland tiebreaks is also NP-hard to manipulate
  - so Copeland (a simple, well-known system) can be simply tweaked to be NP-hard to manipulate
- Adding a *preround* tweak to many ranked-ballot systems can make them NP-hard to manipulate [Conitzer and Sandholm]
  - alternatives are paired and the pairwise loser of each pair is eliminated before the main election protocol is executed

Deterministic preround tweak

[Conitzer and Sandholm]

1. The alternatives are paired before voting takes place. If there is an odd number of alternatives, one gets a bye.
2. In each pairing of two alternatives, the one losing the pairwise election between the two is eliminated. An alternative with a bye is never eliminated.
3. The original ranked-ballot system is used on the remaining alternatives to produce a winner.

- Adding this tweak to plurality, Borda, Simpson-Kramer and STV make them NP-hard to manipulate

Deterministic preround tweak (cont.)

EXISTENCE OF A WINNING PREFERENCE (EWP)

INSTANCE: Set A and a distinguished member a of A; set B of transitive preference orders on A.
QUESTION: Does there exist a preference order b_0 on A such that a wins according to the election system with $B \cup \{b_0\}$?

- For many systems with the deterministic preround tweak, solving EWP is NP-hard
- Proof idea: an arbitrary SAT instance is converted to a set of ranked votes over an alternative set that include one for each literal such that a can be made to win iff each clause can be satisfied by an assignment (implied by the manipulating ballot)

Randomized preround tweak

[Conitzer and Sandholm]

- Same as deterministic preround tweak, except alternatives are paired randomly after voting
- Applying to many ranked-ballot systems makes them #P-hard to manipulate
- Proof shows that a manipulating algorithm must solve PERMANENT (finding the number of matchings in a bipartite graph)

Interleaved preround tweak

[Conitzer and Sandholm]

- Same again, except alternative-pairing and voting are interleaved
- Applying to many ranked-ballot systems makes them PSPACE-hard to manipulate
- Proof shows that a manipulating algorithm must solve STOCHASTIC-SAT

Outline

- Introduction
- Hardness of finding the winner(s)
  - Polynomial systems
  - NP-hard systems
    - The minimax procedure
- Hardness of voter manipulation
  - What is manipulation?
  - Polynomial systems
  - NP-hard systems
    - Second-order Copeland
  - Tweaks to make manipulation NP-hard
- Approximating minimax

Approximating minimax

- Conitzer and Sandholm's tweaks made a system hard to manipulate
- It may be acceptable to find a "good enough" minimax winner set
  - effectively tweaking minimax to make easier to compute the winner(s)
- Minimax can be approximated in polynomial time
  - one PTAS is due to Li, Ma and Wang

Approximating minimax (cont.)

- Gąsieniec *et al.* give a $(1+\epsilon)$-approximation for the Hamming radius p-clustering problem (p-HRC)
  - minimax is equivalent to 1-HRC
  - their algorithm yields a $(1+\epsilon)$-approximation for minimax that runs in $2^{O(\varrho/\epsilon)}n^{O(1/\epsilon)}k^2$ time where ϱ is the "maxscore" of the optimal solution ($\varrho \leq k$)
  - runs in polynomial time if $\varrho = O(\log(k+n))$
- Gąsieniec *et al.* also give a simple 2-approximation algorithm for p-HRC that works for minimax

Is it desirable to be easy or hard to find the winner(s)?

- Better to be easy?
  - Ease and transparency of counting process is desirable for public elections
- Better to be hard?
  - Easy to find winner(s) $\Longrightarrow$ easy to manipulate?
    - proved false by 2nd-order Copeland
  - Hard to find winner(s) $\Longrightarrow$ hard to manipulate?
    - seems true intuitively but not yet proved
- Perhaps ideal: a system for which it's easy to find winner(s) but hard to manipulate

What does it mean to be hard to manipulate?

- This work has shown that some systems are NP-hard to manipulate
- To be NP-hard to manipulate is to be computationally intractable *in the worst case* to find a ballot that will be *certain* to elect a given alternative
- It may still be easy to find a manipulating ballot in certain common cases
- It may still be easy to find a ballot that is very *likely* to elect a given alternative (or at least very unlikely to backfire) in all cases
- Effective manipulation heuristics may still be found for any given system

Paper references

- Brams, Steven J., D. Mark Kilgour and M. Remzi Sanver. *A Minimax Procedure for Negotiating Multilateral Treaties*. October 2003.
- Gąsieniec, Leszek, Jesper Jansson and Andrzej Lingas. Approximation Algorithms for Hamming Clustering Problems. *Proceedings of the 11th Symposium on Combinatorial Pattern Matching*, 108–118, 2000.
- Bartholdi III, John J., C. A. Tovey and M. A. Trick. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare*, 6:227–241, 1989.
- Conitzer, Vincent, and Tuomas Sandholm. Universal Voting Protocol Tweaks to Make Manipulation Hard. *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, 2003.

Other references

- Bartholdi III, John J., and James B. Orlin. Single Transferable Vote Resists Strategic Voting. *Social Choice and Welfare*, 8:341–354, 1991.
- Bartholdi III, John J., C. A. Tovey and M. A. Trick. Voting Schemes for which It Can Be Difficult to Tell Who Won the Election. *Social Choice and Welfare*, 6:157–165, 1989.
- Frances, M., and A. Litman. On Covering Problems of Codes. *Theory of Computing Systems*, 30:113–119, March 1997.
- Gibbard, Allan. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41:587–602, 1973.
- Li, Ming, Bin Ma and Lusheng Wang. Finding Similar Regions in Many Strings. *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, 473–482, 1999.
- Satterthwaite, Mark A. Strategyproofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. *Journal of Economic Theory*, 10:187–217, 1975.

Thanks to Ron Cytron, Steven Brams and my committee