Talk 2

Economics and Computation
May 22, 2013
Fabian Kolbe

Outline

- Introduction (Strategic Voting, Computational Complexity)
- The computational difficulty of manipulating an election

  1) J. Bartholdi, III, C. A. Tovey, and M. A. Trick. **The computational difficulty of manipulating an election.** Social Choice and Welfare, 6(3):227–241, 1989.

- When are elections with few candidates hard to manipulate?

  2) V. Conitzer, T. Sandholm, and J. Lang. **When are elections with few candidates hard to manipulate?** Journal of the ACM, 54(3), 2007.

Voting

- Voting: method for preference aggregation in multiagent settings
- Applications:

  - political elections

  - decision making

  - planning

  - automated group decisions (agents = software)

Definition of voting

$V = \{v_1, . . ., v_n\}$ – finite set of voters

$X = \{1, . . .,m\}$ – finite set of candidates

preferences of voter v_i are given by a linear order O_i on X: $(c_1, c_2, . . ., cm)$

A preference profile is a vector $P = <O_1, . . ., O_n>$

A voting scheme is a function from the set of all preference profiles to the set of candidates X.

Manipulating an election

- Manipulating an election means voting strategically
- Definition: The manipulator does not vote according to his true preferences, but he votes so that the outcome is most beneficial for him.

Question to the audience: *Should this manipulation really be avoided? Isn't it so that the wished outcome determines the preference of a voter? Let's discuss!*

Thoughts to the question

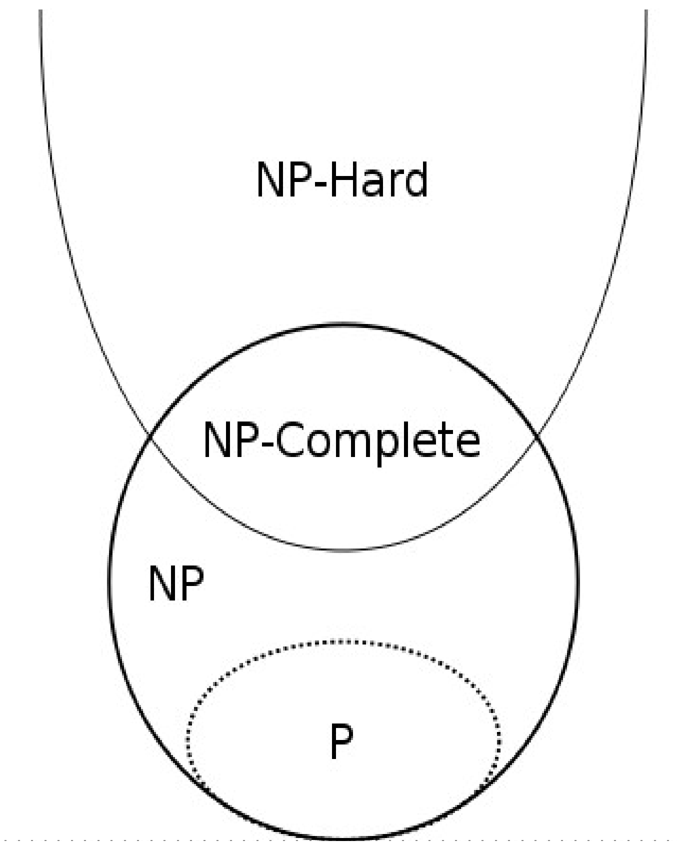
- Con: - voting preference is almost never caused by something else than the outcome

- Pro: - some people have an advantage due to their knowledge (math, other voters..) → unfair
  - voters could be forced/ bribed to vote not according to their true preference, what means to manipulate

How to prevent this manipulation

- Idea: Making it hard to achieve a beneficial manipulation
- → making it hard to computationally find such an beneficial manipulation
- → considering computational complexity

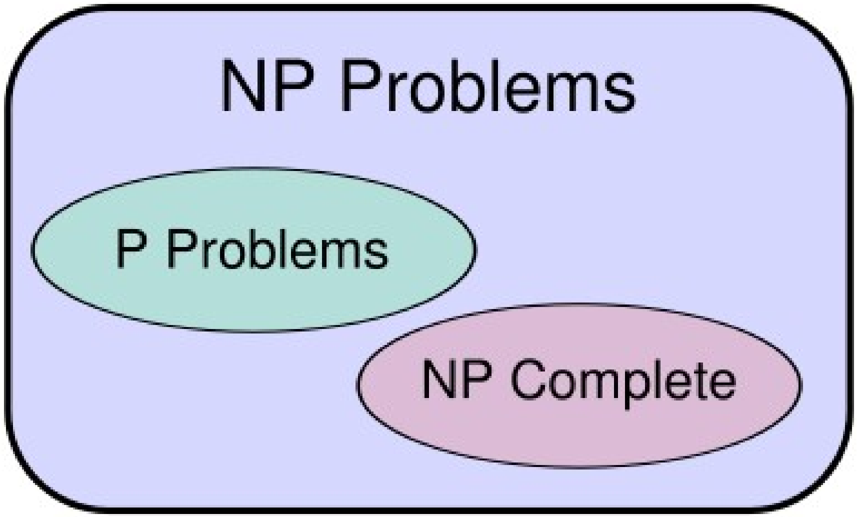
Computational Complexity



- Measured in computational steps for a worst case scenario
- Using a hierachy of complexity classes

Complexity classes

- P: polynomial
- NP: Contains P and NP-Complete problems
- NP-complete: hardest problems in NP. All NP problems can be reworded as NP-complete problem. No algorithm found to solve this alogorithms in polynomial time.



Computational Complexity

Generally:

- Problem is in P → tractable
- Problem is in NP-complete → intractable → hard to manipulate

Voting schemes

- Computational complexity of manipulating depends on used scheme
- Plurality: standard scheme. Every voter cast one vote for his preferred candidate.
- → more schemes presented later

The computational difficulty of manipulating an election

- Greedy-Manipulation algorithm
- Second-order Copeland, a resistant voting scheme

Greedy-Manipulation Algorithm

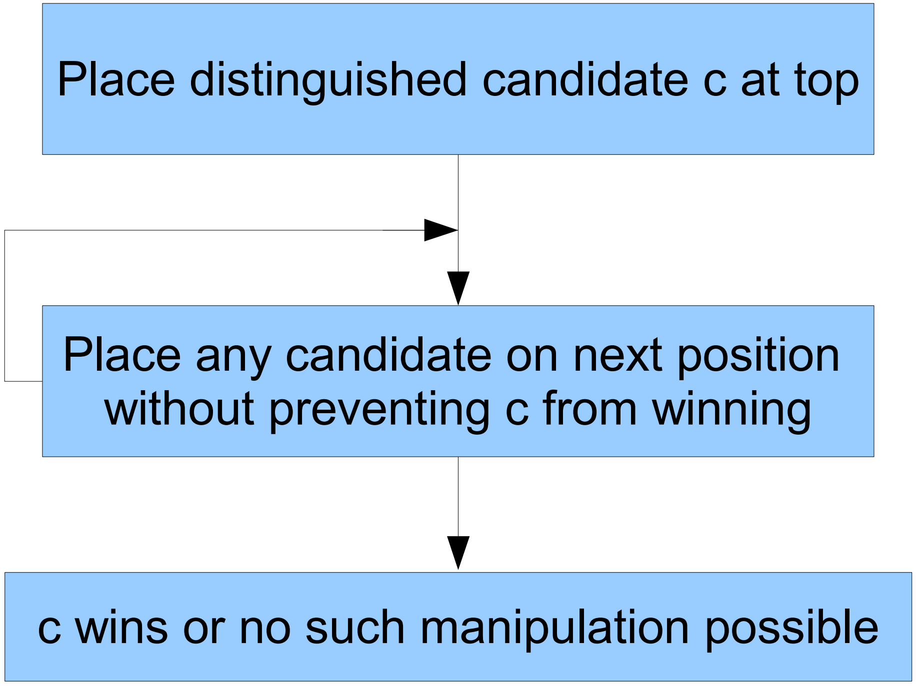
- *Theorem 1*: Greedy-Manipulation will find a preference order P that will make candidate c a winner (or conclude that it is impossible) for any voting scheme (represented as S(P)) is both:

  - Responsive - „candidate with most votes wins“

  - Monotone - „equal or more votes leads at least to the same position“

When is a scheme easy to manipulate?

- Use of Greedy Manipulation Algorithm
- Preocondition: manipulator knows all other votes



Proof of Greedy-Manipulation

- Note: If the algorithm successfully constructs an order, candidate c wins
- → Show that the algorithm will find the order, if such an order exists

  → Use of contradiction proof

- *Corollary*: Any voting system that satisfies the conditions of Theorem 1, and for which S is evaluatable in polynomial time, can be manipulated in polynomial time.

Manipuable by Greedy-Manipulation

- Plurality
- Borda
- Copeland's method

→ scheme needed that is resistant to this manipulation

Second-order Copeland scheme

- Introducing Second-order Copeland scheme
- Expectation: Voting scheme which is easy to use, but hard to manipulate

- *Theorem*: There exists a social choice function (Second Order Copeland) that is simultaneously

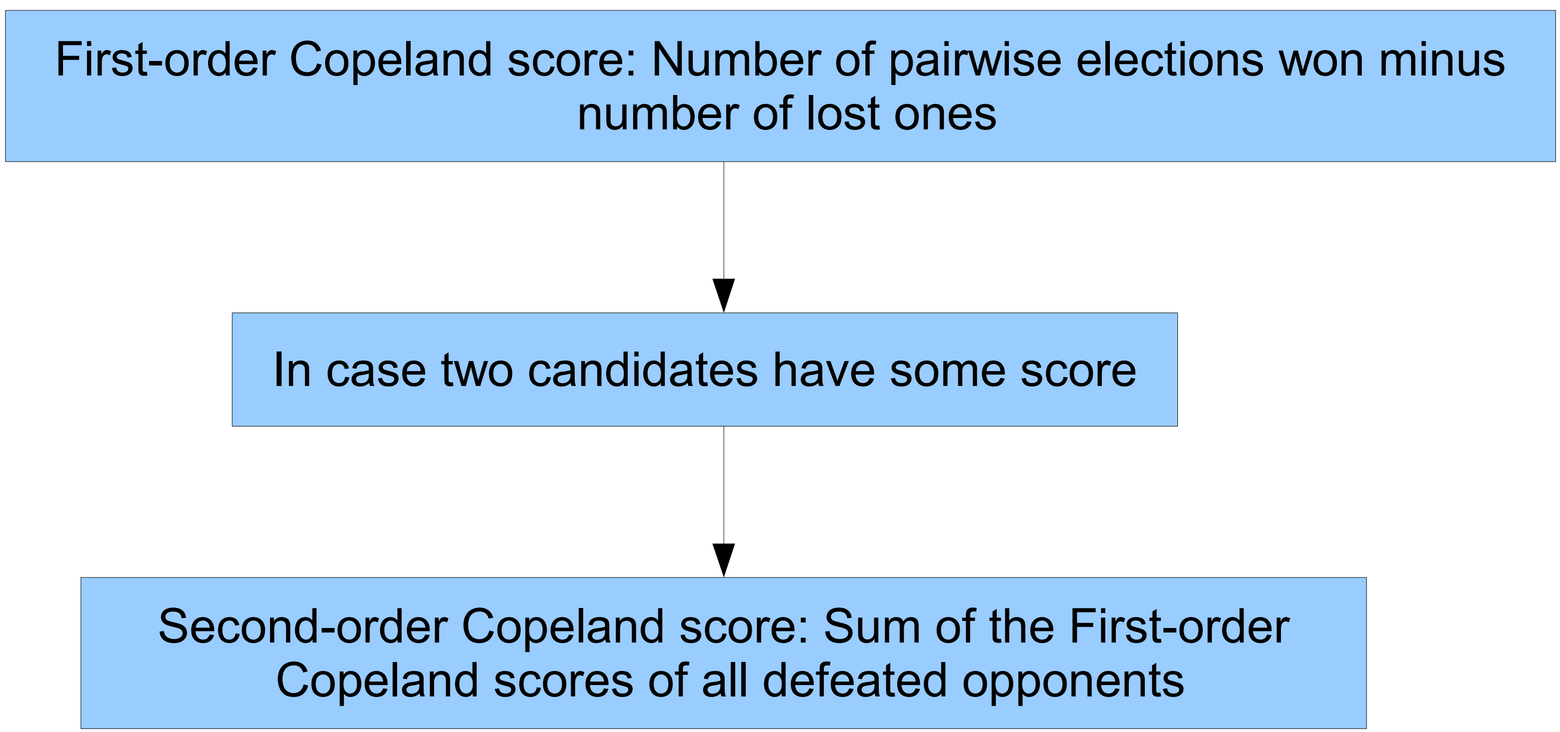
  (1) single-valued;

  (2) non-dictatorial;

  (3) easy to compute, but computationally difficult to manipulate.

Second-order Copeland

- First-order Copeland: pairwise elections
- Second-order Copeland adds Tie-breaking rule:



Theorem 2

- Theorem 2: Tournament outcome under second-order Copeland is NP-complete

- → Proof: It is as hard as a problem known to be NP-complete: 3,4-Satisfiability

3,4-Satisfiability

- Clauses: C1, …, Cm
- Variables: X1,X'1, …, Xn,X'n
- Each clause contains 3 different variables
- Each variable appears in 4 clauses

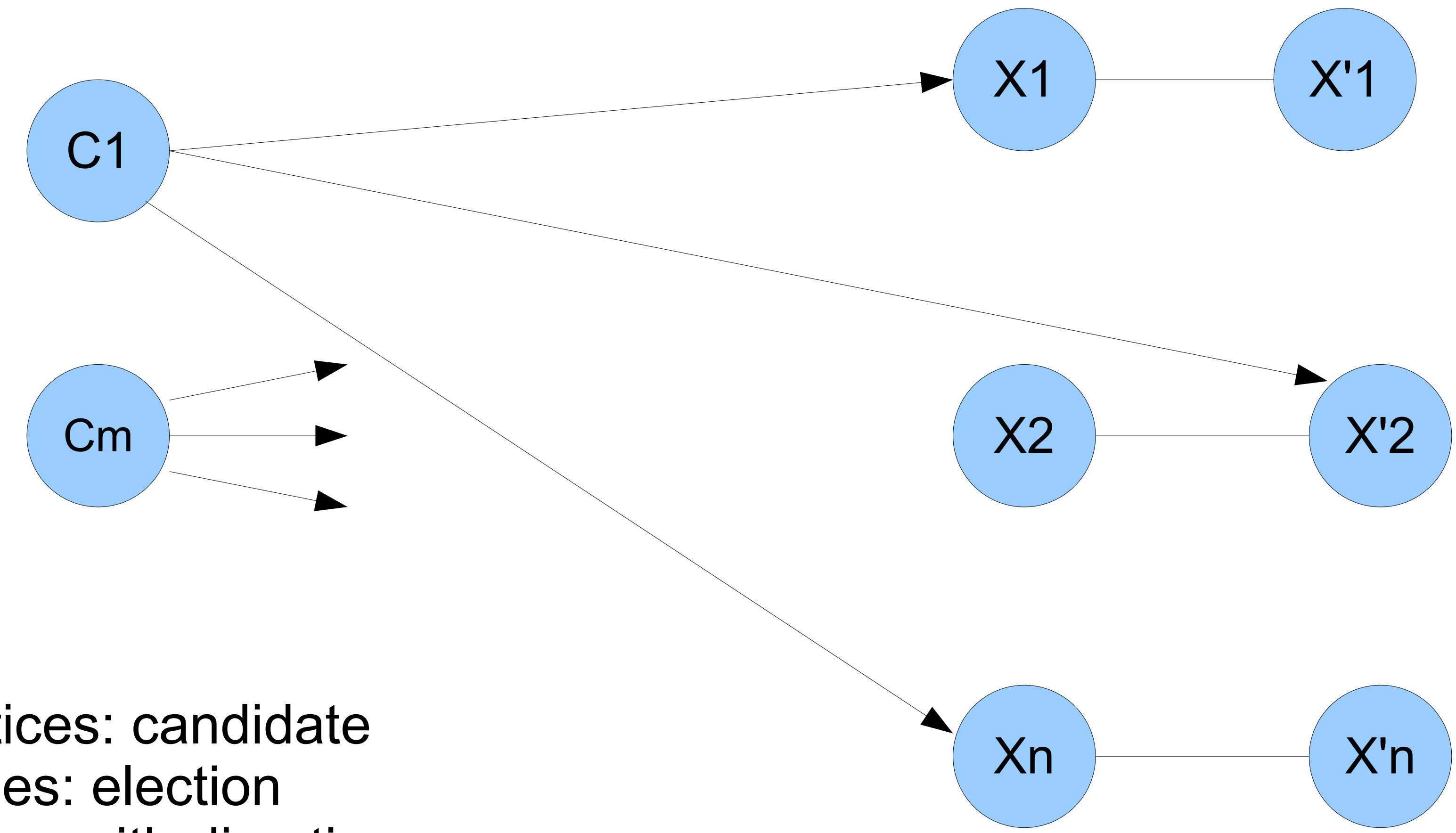
- → for second-order Copeland:

  - One Candidate c out of C is distinguished candidate

  - each Ci and each Xj is a candidate

  - each Ci defeated 3 of Xj, each Xj was defeated by 4 Ci

Graph



Vertices: candidate
Edges: election
Edges with direction: won
Edges without direction: not yet decided

Properties

- → now assign the graph to fulfill following properties:

  R: set of unidrected edges

  1) All candidates c will be tied in first Copeland score

  2)The second-order Copeland score of the distinguished candidate c is independent of R.

Properties 3-5

3)Without R each Candidate Ci has Second-order Copeland score 3 points lower than c

4)Each Ci defeated 3 Xj candidates

5)For any R, **c** wins with second-order Copeland

→ if all 5 properties are satisfied, our candidate is the winner

Theorem 3

- Leads to:
- Theorem 3: Existence of a winning preference for second-order Copeland is NP-complete
- Theorem 4: Manipulation of second-order Copeland is NP-complete

Conclusion of paper 1

- Second-order Copeland is easy to use but hard to manipulate
- → it satisfies our requirements

But...coming to paper 2

- Voting schemes might be only exponential to number of candidates?
- Considering more realistic voting situations

When are elections with few candidates hard to manipulate?

- Motivation:

  - Find the exact number of candidates that makes manipulation hard for different schemes

  - Examine different dimensions of elections and their impact

Considering Dimensions

What information do the manipulators have about the nonmanipulators' votes? → complete or incomplete settings

Who is manipulating: An individual voter or a coalition of voters?

Are the voters weighted or unweighted?

What is the goal of manipulation? → constructive (making someone win) or destructive (making someone not win) manipulation

Used Dimensions

- Main focus in the paper on: constructive and destructive coalitional weigthed manipulation with complete information (CW-Manipulation)
- Number of Candidates is a small constant
- Voters have different weight.

  → Unweigthed voters: all voters have the same weight
- → Used dimensions lead to other results than first paper

Single Transferable Vote (STV)

- Proceeds through a series of rounds.
- In each round the candidate with the lowest score of the remaining ones is eliminated
- The votes for that candidate transfer to the remaining candidates in order of the preference
- STV is used in many Australian political elections
- Plurality with runoff: Same principle like STV, but in the first round all except two candidates are eliminated

Easiness Results

- Plurality: constructive manipulation can be done in polynomial time for all candidates

  Proof: *manipulators simply check if their candidate will win if they vote for him.*

- All voting schemes with 2 candidates are easy to manipulate, because they are comparable to plurality.

Hardness Results

- General Procedure: Proofing that scheme with l+1 candidates is NP-complete, when it's P for l
- Using reductions of partition problem (which is NP-complete):

  given a set of *S* of integers, determine two disjoint subsets S_1 and S_2 where sum(S_1) = sum(S_2)
- Example: {1,8}={2,3,4}

Hardness

- Theorem: For the STV protocol, CONSTRUCTIVE CW-MANIPULATION is NP-complete for three candidates.

  Proof:

  A case, where 2 candidates must get exactly the same number of points, so that the 3. candidate drops out in the first round (else he would win in the second round) → partition problem → NP complete
- → Also applies for plurality with runoff

Construcive Manipulation

Number of candidates: Voting scheme:	2	3	4,5,6	≥ 7
Borda	P	NP-complete	NP-complete	NP-complete
Veto	P	NP-complete	NP-complete	NP-complete
STV	P	NP-complete	NP-complete	NP-complete
Plurality with runoff	P	NP-complete	NP-complete	NP-complete
Copeland	P	P	NP-complete	NP-complete
Maximin	P	P	NP-complete	NP-complete
Randomized Cup	P	P	P	NP-complete
Regular cup	P	P	P	P
plurality	P	P	P	P

Destructive manipulation

- Destructive manipulation can never be harder than constructive manipulation
- Only STV and plurality with runoff are NP-complete

Destructive Manipulation

Number of Candidates: Voting scheme:	2	3
STV	P	NP-complete
Plurality with runoff	P	NP-complete
Borda	P	P
Veto	P	P
Copeland	P	P
Maximin	P	P
Regular cup	P	P
Plurality	P	P

Unweigthed/Individual voting

- If there is an individual voter or a unweighted setting the manipulation problem is always in P for a small constant of candidates
- Comparison to paper 1 → doubts were true

Incomplete setting

- Only restricted probability distributions → uncertainity
- Effects of uncertainity:

  With weighted voters, whenever coalitional manipulation is hard, evaluating a candidate's probability to win is hard when there is uncertainty.
- In an incomplete setting unweigthed or individual voting can be NP-complete.

Conclusion

- STV and plurality with runoff are the schemes, which are the hardest to manipulate
- NP-complete means only weak gurantee of hardness →

  Outlook: Try to make manipulation hard for every or at least most instances

References

1) J. Bartholdi, III, C. A. Tovey, and M. A. Trick. **The computational difficulty of manipulating an election.** Social Choice and Welfare, 6(3):227–241, 1989.

2) V. Conitzer, T. Sandholm, and J. Lang. **When are elections with few candidates hard to manipulate?** Journal of the ACM, 54(3), 2007.

Images:

www.wikepedia.org