Computational Voting Theory, Algorithmic Game Theory and the Connections Between Them

Progress Lecture by Michael Zuckerman
Advisor: Prof. Jeffrey S. Rosenschein

Agenda

- Introduction
- Computational Voting Theory (CVT)
  - Manipulation under Maximin
- Algorithmic Game Theory (AGT)
  - Transformation Games
  - Effort Games
- The connections between CVT and AGT
  - Manipulating the quota in WVG
  - An NTU cooperative game theoretic view of manipulating elections
- Summary

Computational Voting Theory – Introduction

- Voting enables a group of agents (voters) to make a joint choice from a set of alternatives
- Each voter reports his preferences over the alternatives
- A voting rule is applied to aggregate the preferences of the agents

Manipulation in Voting

- Sometimes a subset of the agents can gain by reporting their preferences insincerely
- A voting rule is strategy-proof if there is never a (beneficial) manipulation under this rule.

Gibbard-Satterthwaite Theorem

- When there are at least 3 alternatives, there is no voting rule that satisfies all of the following conditions:
  - Non-imposition: every alternative wins under some profile
  - Non-dictatorship: there is no voter such that we always choose that voter's most preferred alternative
  - Strategy-proof (non-manipulable)

Computational Complexity as a Barrier Against Manipulation

- Second order Copeland and STV are NP-hard to manipulate [Bartholdi et al. 89, Bartholdi & Orlin 91]
- Many hybrids of voting rules are NP-hard to manipulate [Conitzer & Sandholm 03, Elkind and Lipmaa 05]
- Many common voting rules are hard to manipulate for weighted coalitional manipulation [Conitzer et al. 07]
- All of these are *worst-case* results: it could be that *most* instances are easy to manipulate
  - Some evidence that this is indeed the case [Procaccia & Rosenschein 06, Conitzer & Sandholm 06, Zuckerman et al. 09, Friedgut et al. 08, Xia & Conitzer 08a, Xia & Conitzer 08b, Isaksson et al. 10]

Algorithmic Game Theory – Introduction

- Combines computer science, game theory and social choice
- Studies interactions among electronic agents
- The agents are often designed to be:
  - Intelligent
  - Self-interested
  - Computationally efficient

Introduction

- Non-cooperative game theory
  - Individually acting agents
  - Nash equilibrium
  - Dominant-strategy equilibrium
- Cooperative game theory
  - Agents forming coalitions
  - What coalitions are most likely to arise
  - How the gains from cooperation are distributed among the agents

Cooperative Game Theory

- Stability in cooperative games
  - The core
  - ε-core
  - Least-core
- Power indices
  - Shapley-Shubik index
  - Banzhaf index
  - Always exist
  - Unique
  - In general are hard to compute
    - There are efficient approximation algorithms

An Algorithm for the Coalitional Manipulation Problem under Maximin

AAMAS'11

Joint work with Omer Lev and Jeff Rosenschein

Agenda

- Introduction
- Constructive Coalitional Unweighted Manipulation (CCUM) problem
- Algorithm for CCUM under Maximin
- 5/3-approximation of the optimum
- Lower bound on approximation ratio
- Simulation results
- Conclusions

Constructive Coalitional Unweighted Manipulation (CCUM) problem

- Given
  - A voting rule *r*
  - The Profile of Non-Manipulators *PNM*
  - Candidate *p* preferred by the manipulators
  - Number of manipulators $|M|$
- We are asked whether or not there exists a Profile of Manipulators *PM* such that *p* is the winner of *PNM* ∪ *PM* under *r.*

Constructive Coalitional Unweighted Optimization (CCUO) problem

- Given
  - A voting rule *r*
  - The Profile of Non-Manipulators *PNM*
  - Candidate *p* preferred by the manipulators
- We are asked to find the minimum *k* such that there exists a set of manipulators *M* with $|M| = k$, and a Profile of Manipulators *PM* such that *p* is the winner of $PNM \cup PM$ under *r.*

Our setting, Maximin

- $C = \{c_1,\ldots,c_m\}$ – the set of candidates
- S, $|S| = N$ – the set of N non-manipulators
- T, $|T| = n$ – the set of n manipulators
- $N_i(c, c') = |\{ k \mid c >_k c', >_k \in S \cup \{1,\ldots,i\}\}|$ – the number of voters from S and from the i first manipulators, which prefer c over c'
- $S_i(c) = \min_{c' \neq c} N_i(c, c')$ – the Maximin score of c from S and the i first manipulators
- Maximin winner $= \mathrm{argmax}_c\{S_n(c)\}$
- Denote $MIN_i(c) = \{c' \in C \mid S_i(c) = N_i(c, c')\}$

Maximin example


a > b > c > d


b > d > a > c


c > b > d > a

	a	b	c	d	score
a	-	1	2	1	1
b	2	-	2	3	2
c	1	1	-	2	1
d	2	0	1	-	0

CCUM Complexity

- CCUM under Maximin is NP-complete for any fixed number of manipulators (≥ 2)

  (Xia et al. ’09 [7])
- It follows that the CCUO is not approximable by constant better than 3/2, unless P = NP
  - Otherwise, if opt = 2, then the output of the algorithm would be < 3, i.e., 2
  - Hence, it would solve the CCUM for n = 2, a contradiction

The heuristic / approximation algorithm

- Fix some order on manipulators
- The current manipulator i
  - Ranks p first
  - Builds a digraph $G^{i-1} = (V, E^{i-1})$, where
    - $V = C \setminus \{p\}$;

      $(x, y) \in E^{i-1}$ iff $(y \in MIN_{i-1}(x)$ and $p \notin MIN_{i-1}(x))$
  - Iterates over the candidates who have not yet been ranked
  - If there is a candidate with out-degree 0, then it adds such a candidate with the lowest score
  - Otherwise, adds a vertex with the lowest score
  - Removes all the outgoing edges of vertices who had outgoing edge to newly added vertex

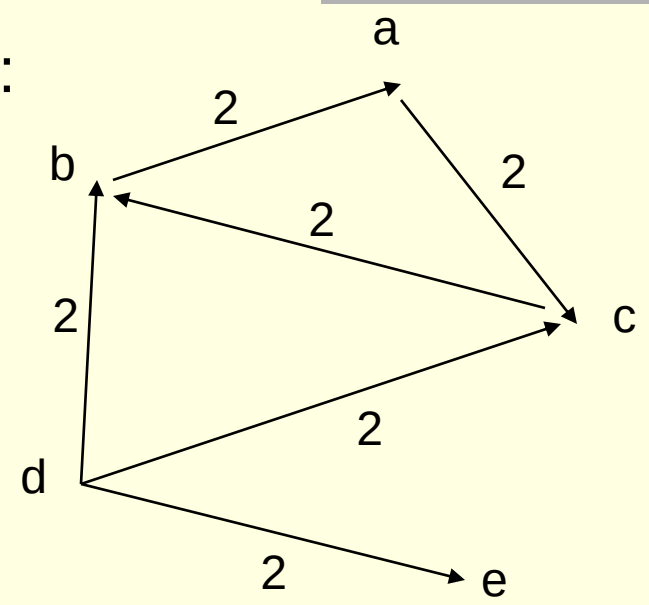
Additions to the algorithm

- The candidates with out-degree 0 are kept in stacks in order to guarantee a DFS-like order among the candidates with the same scores
- If there is no candidate (vertex) with out-degree 0, then it first searches for a cycle, with 2 adjacent vertices having the lowest scores
  - If it finds such a pair of vertices, it adds the front vertex

Example – 1/4

- C = {a, b, c, d, e, p}
- |S| = 6
- |T| = 2
- The non-manipulators' votes:
  - a > b > c > d > p > e
  - a > b > c > d > p > e
  - b > c > a > p > e > d
  - b > c > p > e > d > a
  - e > d > p > c > a > b
  - e > d > p > c > a > b

G^0:



$S_0(p) = N_0(p, b) = 2$

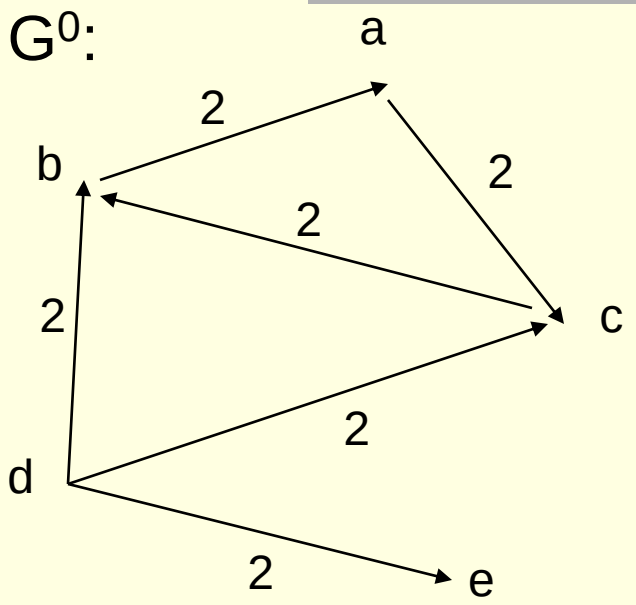
$S_0(e) = N_0(e, p) = 2$

Example - 2/4

- The non-manipulators' votes:
  - a > b > c > d > p > e
  - a > b > c > d > p > e
  - b > c > a > p > e > d
  - b > c > p > e > d > a
  - e > d > p > c > a > b
  - e > d > p > c > a > b
- The manipulators' votes:

p > e > d > b > c > a

G^0:



$S_0(p) = N_0(p, b) = 2$

$S_0(e) = N_0(e, p) = 2$

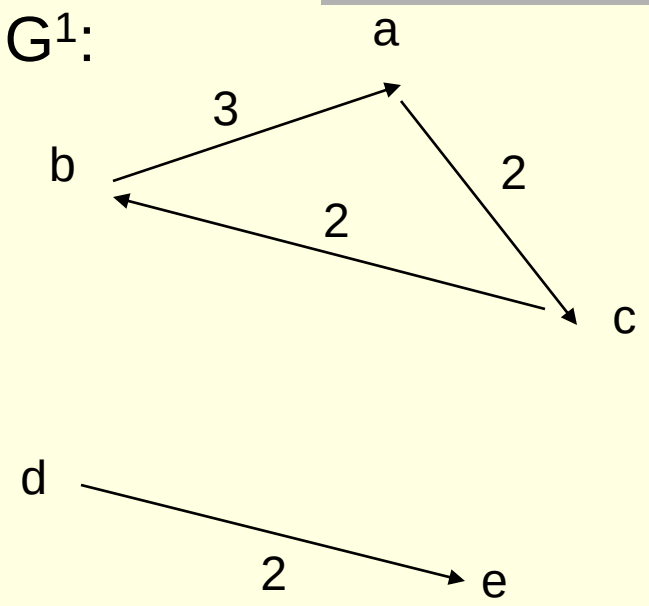
Example - 3/4

- The non-manipulators' votes:
  - a > b > c > d > p > e
  - a > b > c > d > p > e
  - b > c > a > p > e > d
  - b > c > p > e > d > a
  - e > d > p > c > a > b
  - e > d > p > c > a > b
- The manipulators' votes:

p > e > d > b > c > a

p > e > d > c > a > b

G^1:



$S_1(p) = N_1(p, b) = 3$

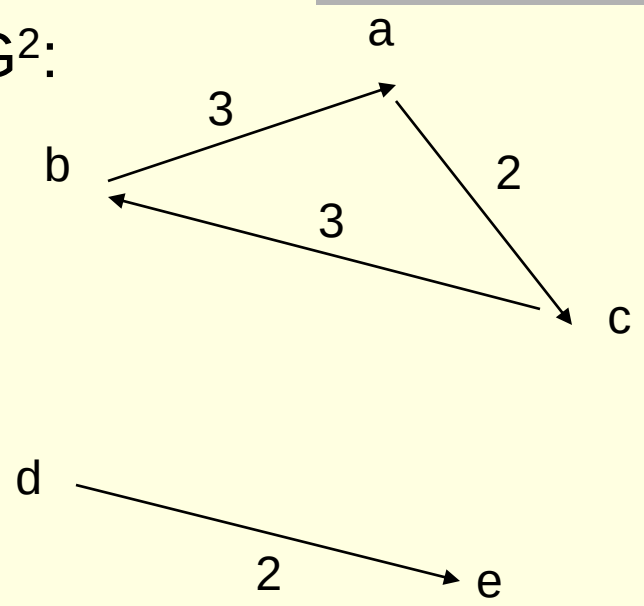
$S_1(e) = N_1(e, p) = 2$

Example - 4/4

- The non-manipulators' votes:
  - a > b > c > d > p > e
  - a > b > c > d > p > e
  - b > c > a > p > e > d
  - b > c > p > e > d > a
  - e > d > p > c > a > b
  - e > d > p > c > a > b
- The manipulators' votes:

  p > e > d > b > c > a
  p > e > d > c > a > b

G^2:



$S_2(p) = N_2(p, b) = 4$
$\max_{x \neq p} S_2(x) = 3$

p is the winner!

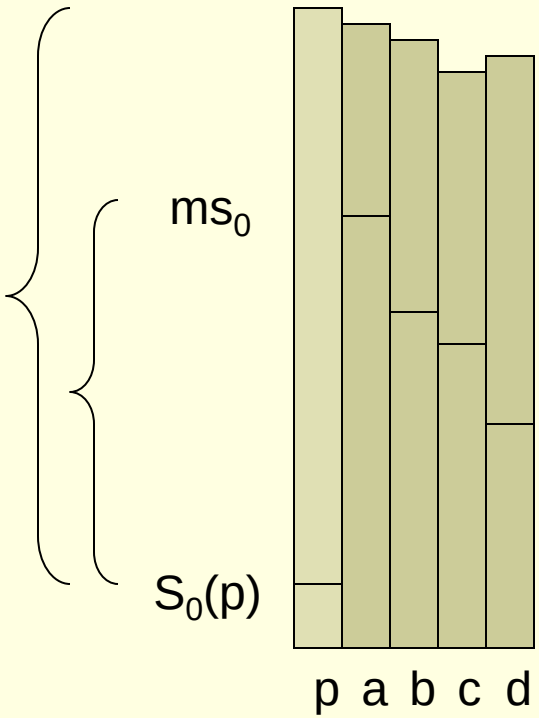
Instances without 2-cycles

- Denote $ms_i = \max_{c \neq p} S_i(c)$
  - The maximum score of *p*'s opponents after *i* stages
- Lemma: If there are no 2-cycles in the graphs built by the algorithm, then for all i, $0 \leq i \leq n-3$ it holds that $ms_{i+3} \leq ms_i + 1$
- Theorem: If there are no 2-cycles, then the algorithm gives a 5/3-approximation of the optimum

Proof of Theorem

$n = \lceil (3\, ms_0 - 3S_0(p) + 3)/2 \rceil$

$opt \geq ms_0 - S_0(p) + 1$

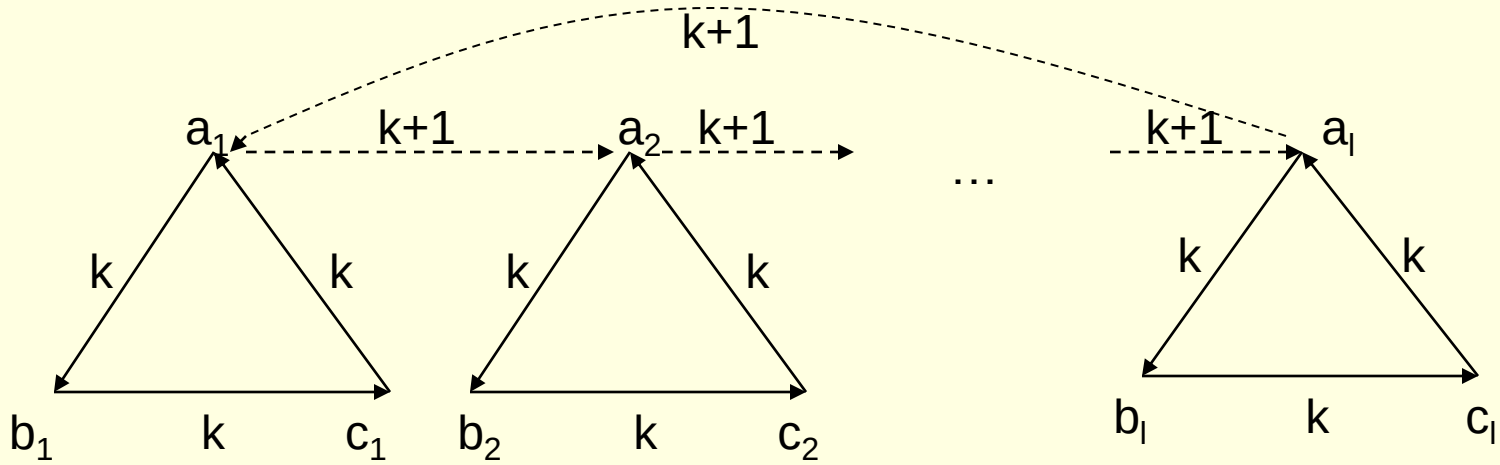


The ratio n/opt is the biggest when opt = 3, $n = \lceil 3/2 * 3 \rceil = 5$

Eliminating the 2-cycles

- Lemma: If at a certain stage i there are no 2-cycles, then for all j > i, there will be no 2-cycles at stage j
- We prove that the algorithm performs optimally while there are 2-cycles
  - Intuitively, if there is a 2-cycle, then one of its vertices has the highest score, and it will always be placed in the end – until the cycle is eliminated
- Once the 2-cycles have been eliminated, our algorithm performs a 5/3-approximation on the number of stages left
- Generally we have 5/3-approximation of the optimal solution

Lower bound on approximation ratio



- When voted:

$p > a_l > c_l > b_l > \ldots > a_1 > c_1 > b_1$

…

- ms^*_i grows by 1 every (m-1)/3 voters

- Our algorithm can vote:

$p > a_1 > c_1 > b_1 > \ldots > a_l > c_l > b_l$

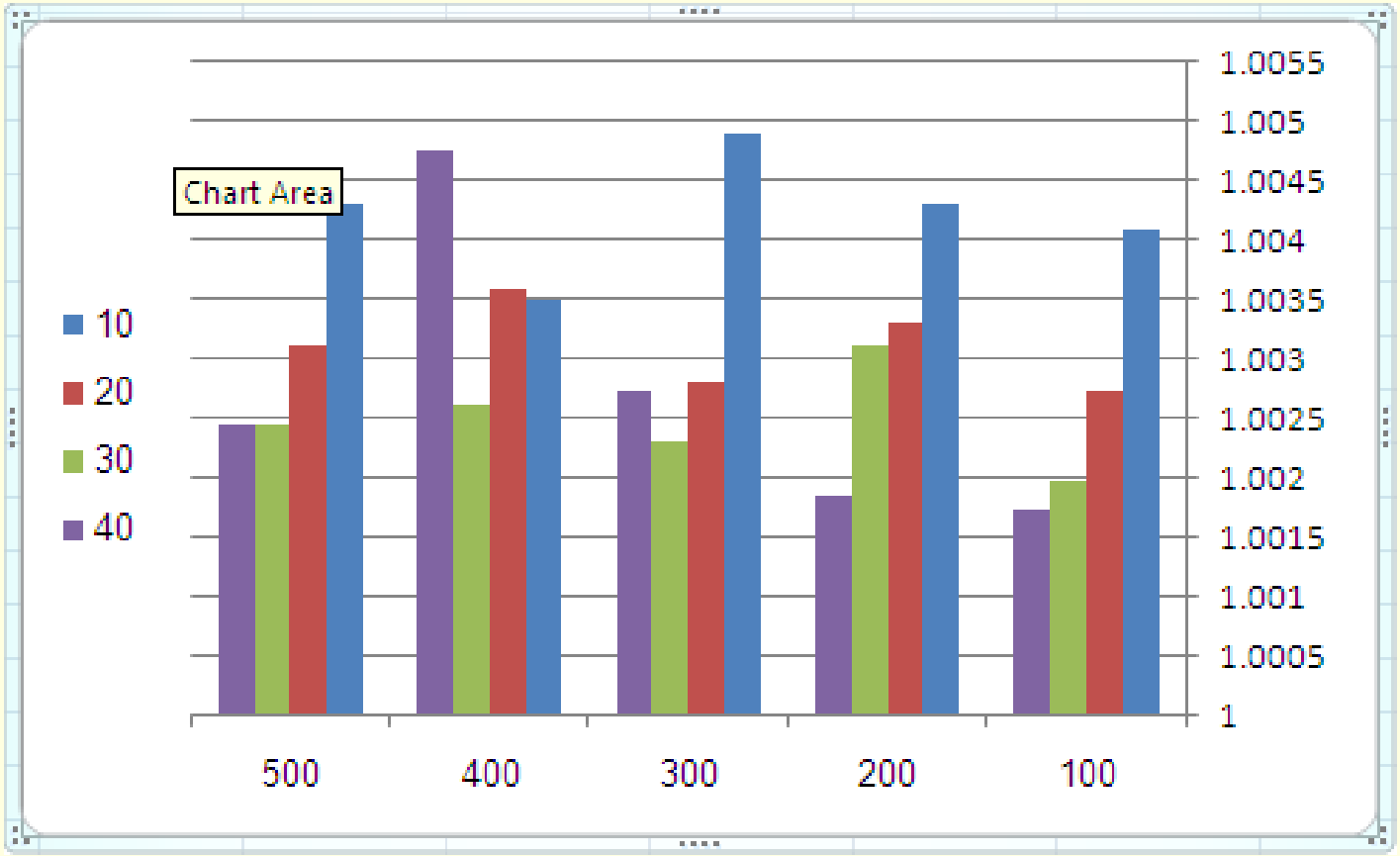
…

- Here ms_i grows by 1 every 3 voters

Simulation results

- Implemented
  - this algorithm
  - the simple greedy algorithm in [8]
- On average, this algorithm is a little better
- On some instances, the simple greedy is better
- No difference between the performance of this algorithm with and without the “additions”
- Difficulties in calculating the optimum

Simple greedy alg / this alg



10 – 40 candidates
100 – 500 non-manipulators

Conclusions

- A new heuristic / approximation algorithm for CCUM / CCUO under Maximin
- Gives a 5/3-approximation of the optimum
- The lower bound on the approximation ratio of the algorithm (and any algorithm) is 1½
- Simulation results – comparison between this algorithm and the simple greedy algorithm in [8]

- Future work
  - Prove the approx. ratio without the additions

Manipulation with Randomized Tie-Breaking under Maximin

AAMAS'12

Joint work with Jeff Rosenschein

The setting

- Dealing with Maximin voting rule
- There is only one manipulator
- Ties are broken uniformly at random
- The manipulator has utilities for the candidates: (1,…,1,0,…,0) with k 1's
- The problem is to maximize the manipulator's expected utility

The results

- The problem is in FPT when parameterized by k
  - Relies on algorithm of Obraztsova et al. (AAMAS'11)
- Under certain conditions it is approximable by a constant factor
- Under certain other conditions it can be solved exactly

Proof Systems and Transformation Games

CoopMAS'10

Joint work with Yoram Bachrach, Michael Wooldridge and Jeff Rosenschein

Transformation Games – 1/2

- Cooperative game
- Players are endowed with
  - initial resources
  - capabilities of deriving certain output resources given certain input resources
- The aim is to generate a particular target resource
- It is achieved by forming coalitions

Transformation Games – 2/2

- Can model cooperative proof systems and supply chains
- We considered possible restrictions on transformation chains
- We considered the computational complexity of:
  - Testing whether a coalition wins
  - Checking whether a player is a dummy or a veto player
  - Computing the core of the game
  - Computing power indices
- Extension: the transformations have associated costs

Effort Games and the Price of Myopia

MLQ’09

Joint work with Yoram Bachrach and Jeff Rosenschein

Effort Games and the Price of Myopia (PoM) – 1/3

- Multi-agent environment
- A common project depends on various tasks
  - Winning and losing subsets of tasks
- The probability of carrying out a task is higher when the agent in charge of it exerts effort
- There is a certain cost for exerting effort
- The principal tries to incentivize agents to exert effort
- The principal can only reward agents based on success of the entire project

Effort Games and the PoM – 2/3

- The PoM is a measure of the influence the model of rationality has on the minimal payments the principal has to make in order to motivate the agents to exert effort.
- We considered the computational complexity of testing whether exerting effort is a dominant strategy for an agent
- The computational complexity of finding a reward strategy, using:
  - Dominant strategy equilibrium, or
  - Iterated elimination of dominated strategies

Effort Games and the PoM - 3/3

- Proved that the above problems are generally #P-hard
  - At least as hard as calculating the Banzhaf power index
- In a certain restricted domain all of the above problems are in P
- Gave bounds on PoM in WV Effort Games
- Provided simulation results in EG played over Series-Parallel Graphs

Manipulating the Quota in Weighted Voting Games

AIJ'12

Joint work with Piotr Faliszewski, Yoram Bachrach and Edith Elkind

Manipulating the Quota in Weighted Voting Games – 1/2

- WVG are described by
  - A set of players
  - A list of players' weights
  - A quota
- The power of a player is identified with
  - His Shapley-Shubik index, or
  - His Banzhaf index

Manipulating the Quota in Weighted Voting Games – 2/2

- Investigated, how much the central authority can change a player's power, by modifying the quota
- Studied, how the choice of quota can affect the relative power of players
- Provided an efficient algorithm for determining whether there is a value of the quota that makes a given player a dummy
- Hardness result for checking which of the two values of the quota makes a player more powerful.

An NTU Cooperative Game Theoretic View of Manipulating Elections

WINE'11

Joint work with Piotr Faliszewski, Vincent Conitzer and Jeff Rosenschein

Constructive Coalitional Unweighted Manipulation (CCUM) Problem

- Given
  - A voting rule *R*
  - The Profile of Non-Manipulators *PNM*
  - Candidate *c* preferred by the manipulators
  - Number of manipulators $|M|$
- We are asked whether or not there exists a Profile of Manipulators *PM* such that *c* is the winner of *PNM* ∪ *PM* under *R.*

Motivation

- In CCUM
  - We assume that the manipulators are single-minded
  - We do not ask how they choose the candidate for whom to vote
- Here
  - We use cooperative game theory to fill this gap
  - Explore the Non-Transferable Utility (NTU) setting

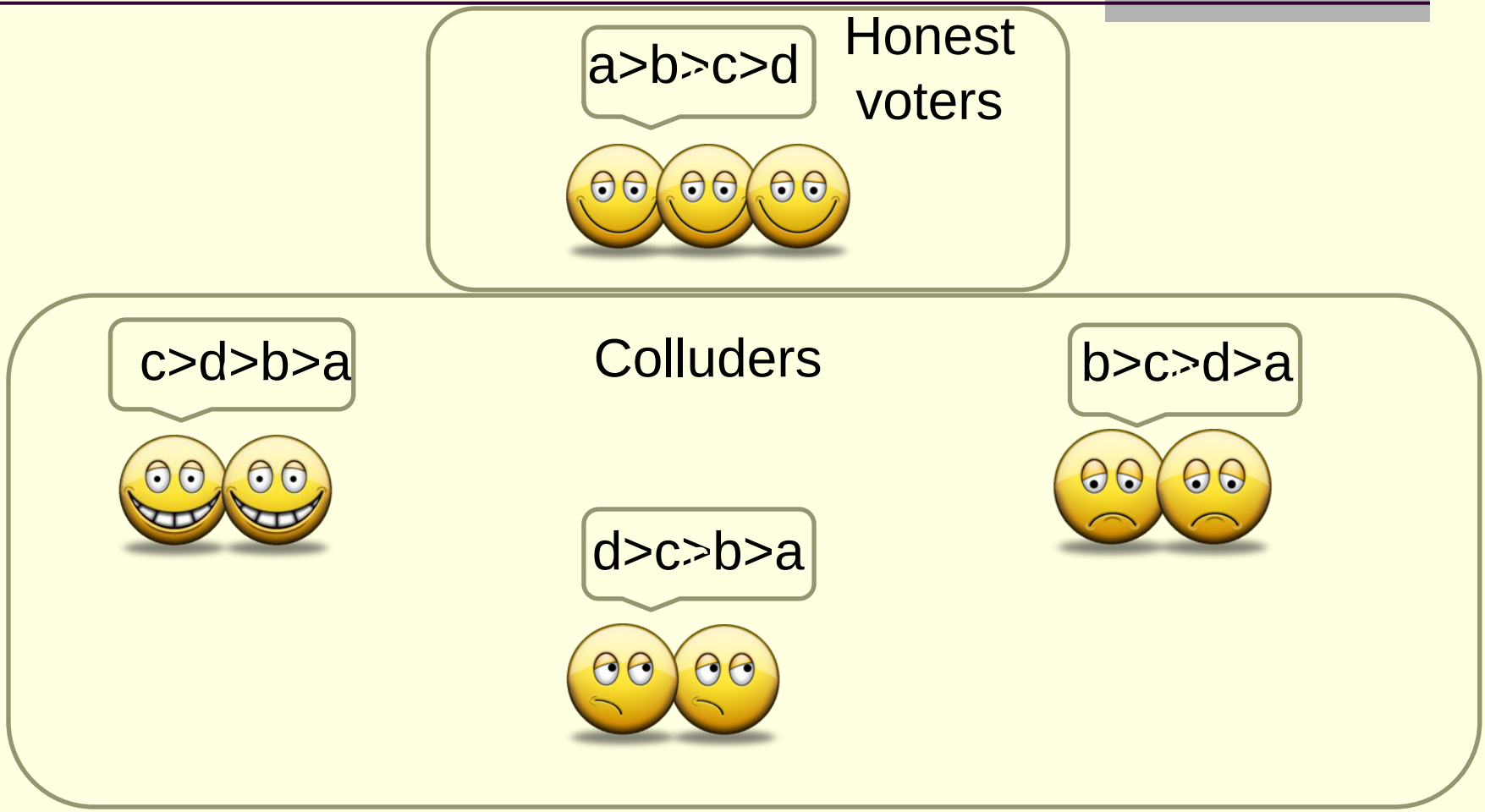
The three settings

- Assume that the manipulators can communicate with one another
- What a (sub)coalition of the manipulators can achieve depends on the votes of manipulators outside the coalition
- Consider 3 settings:
  - Strong Nash equilibrium
  - Pessimistic (α-core)
  - Adaptive (β-core)

Notation

- $C = \{c_1, \ldots, c_m\}$ – the candidates/alternatives
- $V = \{1, 2, \ldots, n\}$ – the voters
- $P = \{P_1, \ldots, P_n\}$ – a preference profile of the voters
- $R(P)$ – the winner under profile P
- $M \subseteq V$ – the colluders/manipulators
- $H = V \setminus M$ – the honest voters
- $W_{ab} \subseteq M$ – all the colluders who prefer a to b

Example



The truthful plurality winner is a

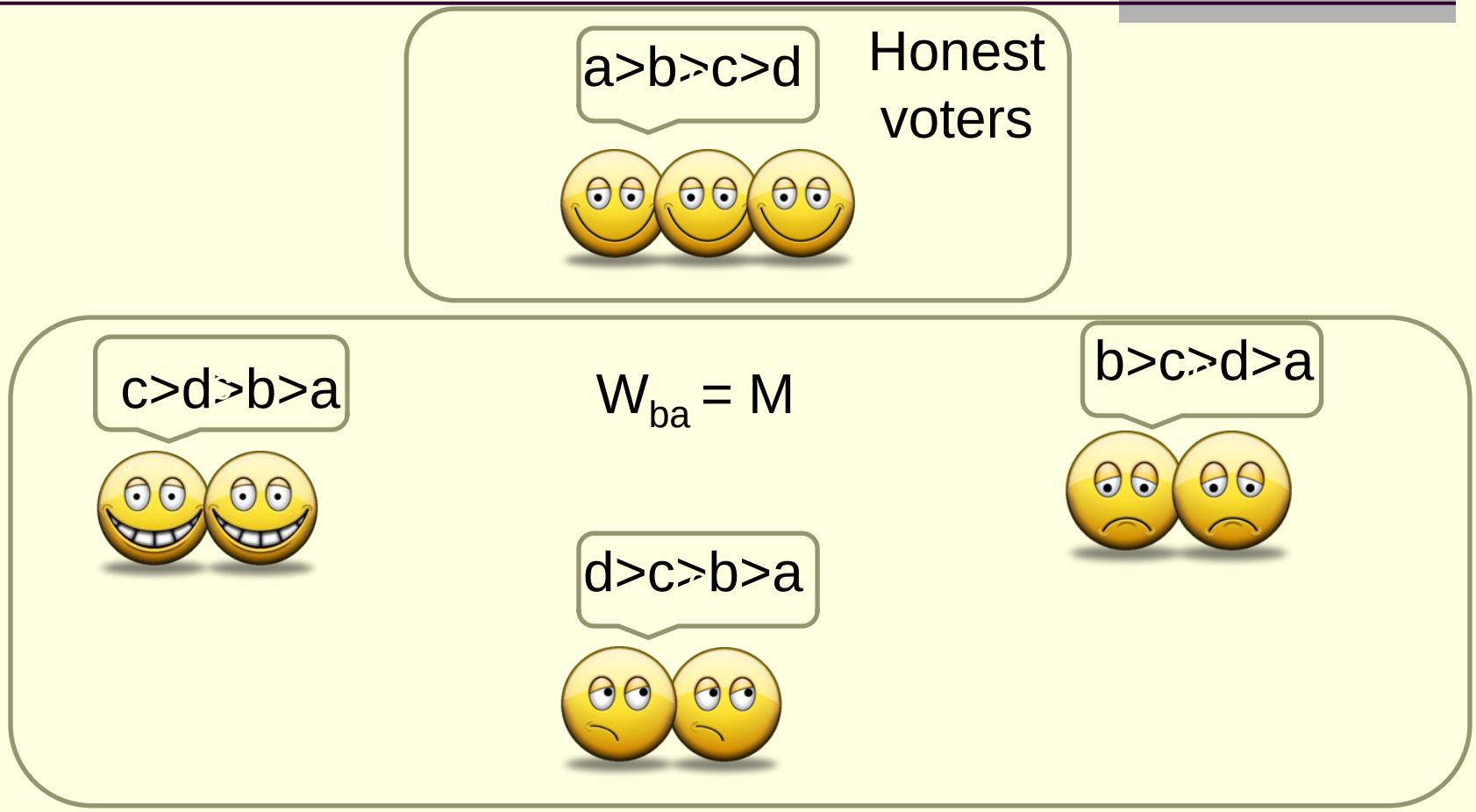
Strong Nash Equilibrium (SNE)

- SNE is a Nash equilibrium in which no coalition, taking the actions of its complements as given, can cooperatively deviate in a way that benefits all of its members
- When a coalition coordinates a deviation, the remaining players are unaware of it → they stick to their agreed-upon strategies

The SNE (more formally)

- Let P_H be a profile of the honest voters
- A profile P'_M is in an SNE if there is no coalition $W \subseteq M$ that can cause an alternative to win that is better for them than $c = R(P_H \cup P'_M)$, when the colluders in $M \setminus W$ vote according to $P'_{M \setminus W}$

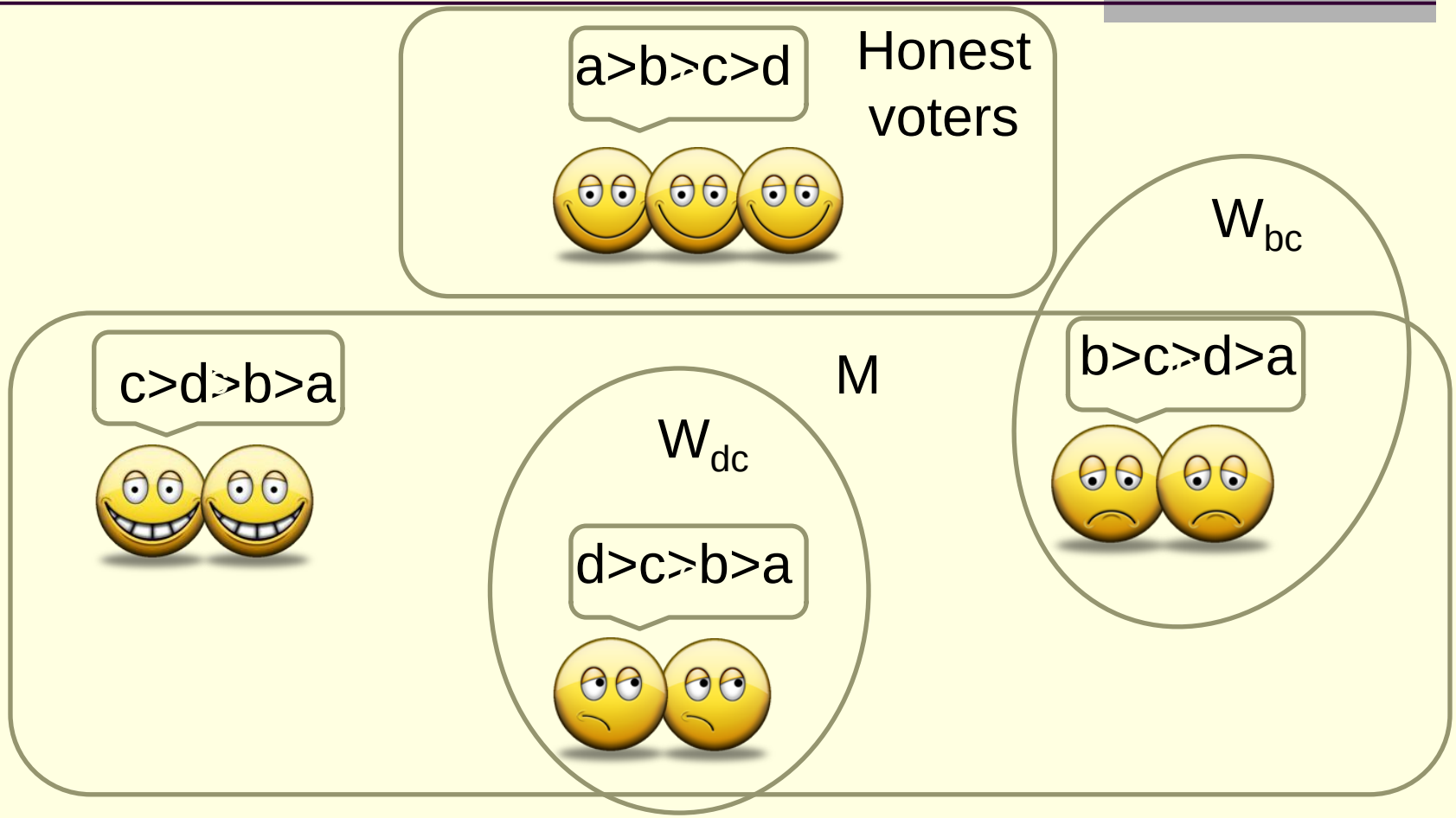
The SNE – example for Plurality - 1/2



Is the profile of truthful preferences an SNE?

No, since everybody in M can deviate and vote for b, for example

The SNE – example for Plurality - 2/2



So, does there exist an SNE?

Yes! When everybody in M votes for c

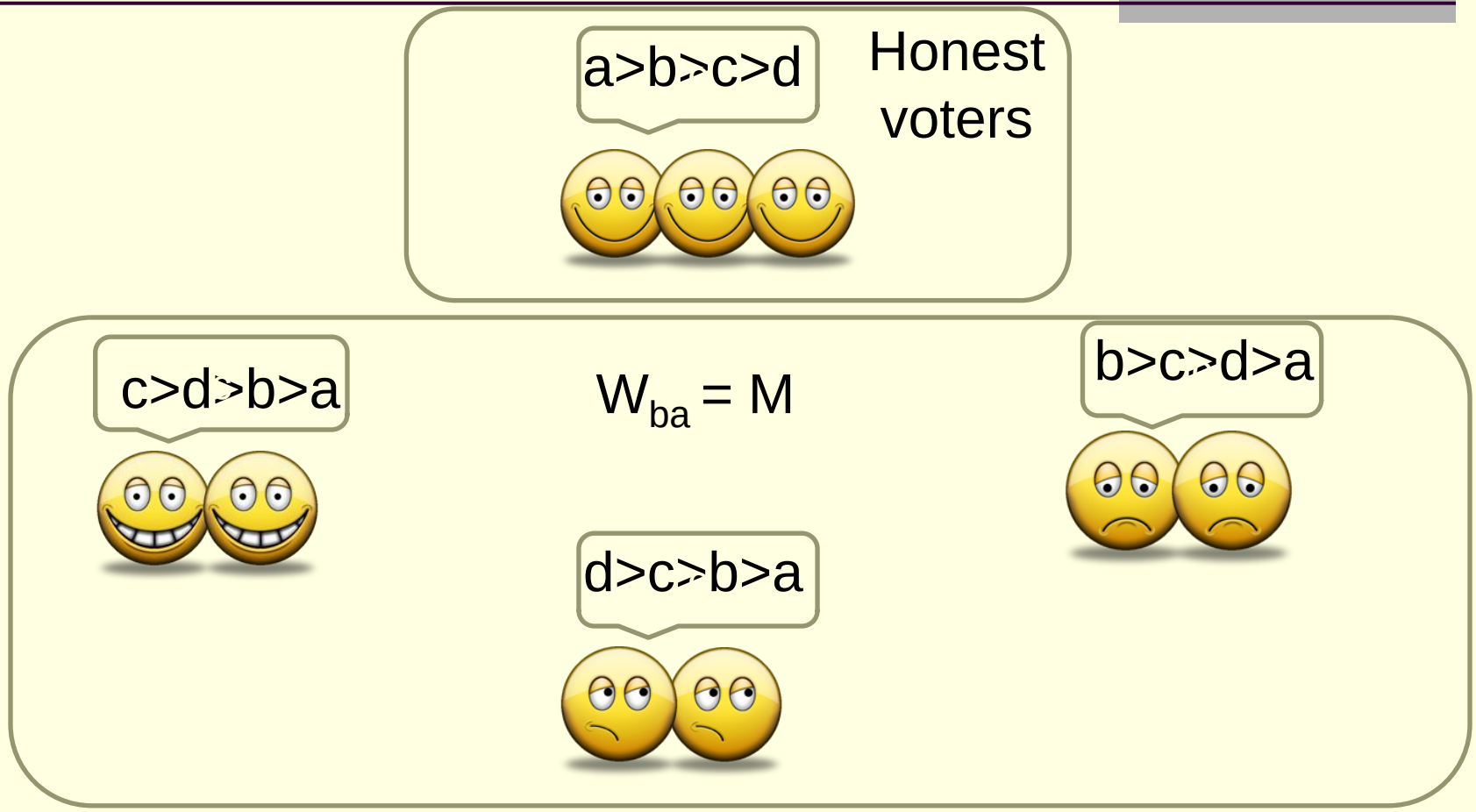
Feasibility

- We say that alternative $c \in C$ is *feasible* for a coalition $W \subseteq M$ if W can make c get elected when everybody in $M \setminus W$ votes truthfully.
- We say that alternative $c \in C$ is *α-feasible* for a coalition $W \subseteq M$ if there exists a profile for W making c get elected for all profiles of $M \setminus W$.
- We say that alternative $c \in C$ is *β-feasible* for a coalition $W \subseteq M$ if for every profile of $M \setminus W$ there exists a profile of W making c get elected.

The Cores

- Let c be an alternative that is feasible for M. We say that c is in a *core* if no coalition $W \subseteq M$ can make a better alternative for them win
  - For α-core we require *α-feasibility*
  - For *β-core* we require *β-feasibility*

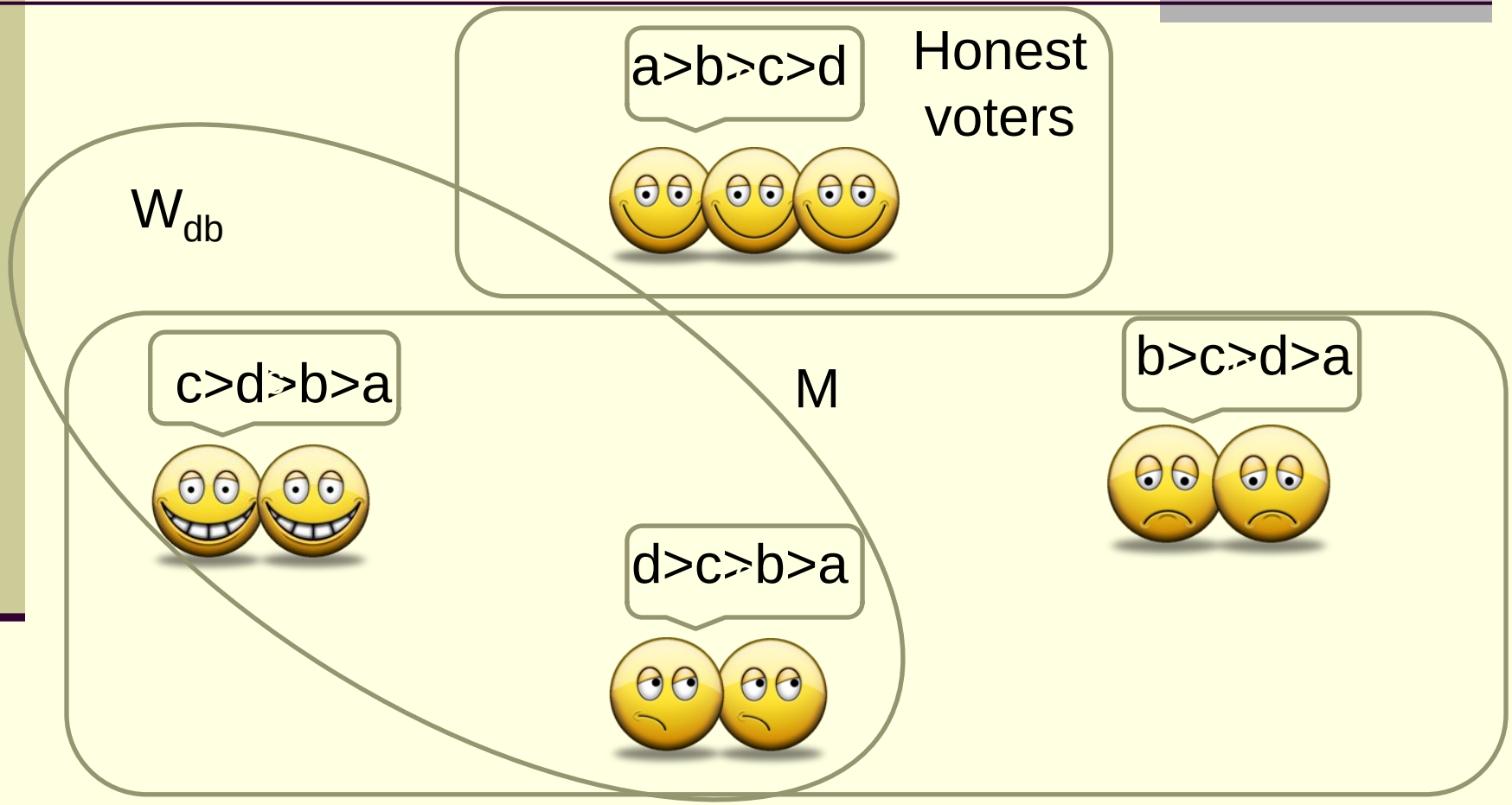
The Cores – example for Plurality - 1/2



Is a∈ α-core?

b is α-feasible for $W_{ba} = M$ → a ∉ α-core

The Cores – example for Plurality - 2/2



Is b∈ β-core?

d is not β-feasible for W_{db}, the same for c, a → b ∈ β -core

SNE vs. β-core

- Lemma: If P'_M is an SNE, then $c = R(P_H, P'_M) \in$ *β-core*
- Proof: If $c \notin$ *β-core* then there exists $c' \neq c$ and $W' \subseteq M$ such that everybody in W' prefers c' to c, and c' is β-feasible for W', then by definition W' can benefit by deviating from $P'_M \rightarrow P'_M$ is not an SNE.

α-core vs. β-core

- Lemma: *β-core* $\subseteq$ α-core
- Proof: Let c be a candidate. If c $\notin$ α-core then there exists a candidate c' ≠ c and a coalition W' $\subseteq$ M such that everybody in W' prefers c' to c, and c' is α-feasible for W' → c' is β-feasible for W' → c $\notin$ *β-core*.

Our results - 1/3

- Theorem 1: The problem of determining whether a given profile P'_M is in an SNE is Turing-reducible to the CCUM problem.
- Proof: Let $c = R(P_H, P'_M)$. For each $x \neq c$, solve the CCUM problem where W_{xc} is the set of manipulators and $P_H \cup P'_{M \setminus W_{xc}}$ are the profiles of the non-manipulators. If for some x and W_{xc} the answer of CCUM is "yes" then P'_M is not an SNE. Otherwise, P'_M is an SNE.

Our results - 2/3

- Theorem 2: The problem of determining whether an alternative c is in the α- or β-core is at least as hard as the CCUM problem.
- Proof: in the reduction from the CCUM, let all the manipulators prefer c. Then c is in the α- and β-core if and only if it is feasible for M.

Our results - 3/3

- Theorem 3: There are polynomial-time algorithms solving the α- and the β-core problems for k-approval and Bucklin voting rules.
- Theorem 4: There is a polynomial-time algorithm solving the Plurality-with-Runoff-β-core problem.

Related work

- The Transferable Utility (TU) setting is studied in “Coalitional Voting Manipulation: A Game-Theoretic Perspective” by Yoram Bachrach, Edith Elkind and Piotr Faliszewski (IJCAI11) [9]
- Peleg and Peters in “Strategic Social Choice” [10] define a rule called “feasible elimination procedure” (f.e.p.)
- F.e.p. elects an alternative which is an outcome of an SNE, and therefore in the α-core and in the β-core

Conclusions

- We explored the stability issue in coalitional manipulation voting games in the NTU setting
- We defined and studied the complexity of the core problems, and an SNE

Future work

- Investigate the complexity of α-core problem in Plurality with Runoff, and α- and β-core problems in Cup voting rule
- What is the complexity, under various voting rules, of finding a profile for the colluders which is an SNE, if one exists?

Summary

- We gave some background on Computational Voting Theory and Algorithmic Game Theory
- We presented algorithms for manipulation problems under Maximin
- We briefly discussed
  - Transformation Games
  - Effort Games
  - Manipulating Quota in WVGs
- We explored manipulation from cooperative game theoretic point of view

Thank you!

References – 1/3

- [1] An algorithm for the coalitional manipulation problem under Maximin, Michael Zuckerman, Omer Lev, and Jeffrey S. Rosenschein. In The Tenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011), Taipei, Taiwan, 2011.
- [2] Manipulation with randomized tie-breaking under Maximin (extended abstract), Michael Zuckerman and Jeffrey S. Rosenschein. In The Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012), Valencia, Spain, 2012.
- [3] Proof systems and transformation games, Yoram Bachrach, Michael Zuckerman, Michael Wooldridge, and Jeffrey S. Rosenschein. In The First Workshop on Cooperative Games in Multiagent Systems (CoopMAS'10), at The Ninth International Joint Conference on Autonomous Agents and Multiagent Systems, Toronto, 2010.
- [4] Effort Games and the Price of Myopia, Yoram Bachrach, Michael Zuckerman and Jeffrey S. Rosenschein. Mathematical Logic Quarterly, Special Issue on Logic and Complexity within Computational Social Choice. 2009, pp. 377-396.

References – 2/3

- [5] Manipulating the quota in weighted voting games, Michael Zuckerman, Piotr Faliszewski, Yoram Bachrach, and Edith Elkind. Journal of Articial Intelligence, 180-181:1-19, April 2012.
- [6] An NTU cooperative game theoretic view of manipulating elections, Michael Zuckerman, Piotr Faliszewski, Vincent Conitzer, and Jeffrey S. Rosenschein. In The 7th Workshop on Internet and Network Economics (WINE 2011), pages 363-374, Singapore, December 2011.
- [7] Complexity of Unweighted Coalitional Manipulation Under Some Common Voting Rules, Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer and Jeffrey S. Rosenschein. The Twenty-First International Joint Conference on Artificial Intelligence (IJCAI 2009), July 2009, Pasadena, California, pp. 348-353.
- [8] Algorithms for the Coalitional Manipulation Problem, Michael Zuckerman, Ariel D. Procaccia and Jeffrey S. Rosenschein. Journal of Artificial Intelligence. Volume 173, Number 2, February 2009, pp. 392-412.

References – 3/3

- [9] Coalitional Voting Manipulation: A Game-Theoretic Perspective, Yoram Bachrach, Edith Elkind and Piotr Faliszewski. The Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011), pp. 49--54, 2011.
- [10] Strategic Social Choice, Bezalel Peleg, Hans Peters. Springer, Berlin, 2010.

Backup slides

Manipulation with Randomized Tie-Breaking under Maximin

AAMAS'12

Joint work with Jeff Rosenschein

The classic manipulation problem

- Given
  - A voting rule *R*
  - The Profile of Non-Manipulators *PNM*
  - Candidate *c* preferred by the (single) manipulator
- We are asked whether or not there exists a preference order *L* for the manipulator such that *c* is the winner of *PNM* ∪ *{L}* under *R.*

Our setting

- $C = \{c_1,\ldots,c_m\}$ – the set of candidates
- $V = \{1,\ldots,n\}$ – the set of voters
- $V' = \{1,\ldots,n-1\}$ – the set of non-manipulators
- The voter n is the manipulator
- The Maximin score of c is:

  $$\min_{c' \neq c} |\{i \mid c >_i c',\ i \in V\}|$$

- The candidates with maximum score win

Notation

- $N(c, c') = |\{ i \mid c >_i c', i \in V'\}|$ – the number of non-manipulators who prefer c to c'
- $S(c) = \min_{c' \neq c} N(c,c')$ – the Maximin score of c from the non-manipulators

Tie-breaking

- Most of the literature assumes tie-breaking relative to the manipulator
  - In favor of manipulator, or
  - Adversarial to the manipulator
- Sometimes some lexicographic tie-breaking is assumed

Randomized tie-breaking

- We assume that ties are broken uniformly at random
- Suppose that manipulator's preference is

  $c_1 > c_2 > \ldots > c_m$
- What is better for the manipulator: c_1 tied with c_m, or c_2?
- Need a notion of utilities

Utility of the manipulator

- $u(c) \in \{0,1\}$ – the manipulator's utility for candidate c
- We assume that the utilities are compatible with the manipulator's preferences
- Let S be the set of the winning candidates
- Then the manipulator's expected utility is

$$\hat{u}(S) = \frac{1}{|S|}\sum_{c \in S} u(c)$$

Manipulation problem with randomized tie-breaking (MRTB)

- Given
  - A voting rule *R* (here: Maximin)
  - The Profile of Non-Manipulators *PNM*
  - The manipulator's utilities of the candidates $u(c_j)$
  - A rational number $q \geq 0$
- We are asked whether or not there exists a vote L for the manipulator such that

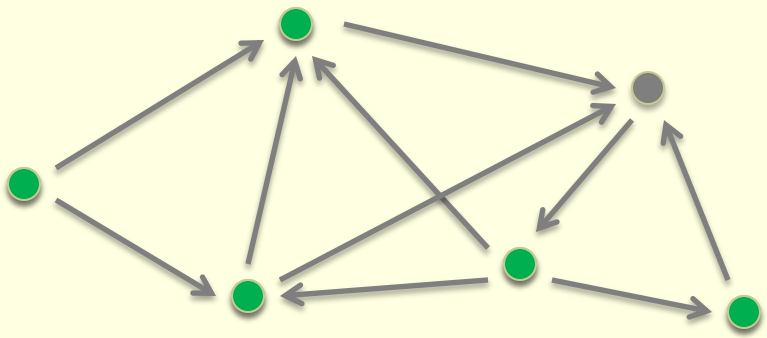
  $$\hat{u}(R(PNM \cup \{L\})) \geq q$$
- In the optimization version we are asked to find the optimal vote for the manipulator

Related work

- Obraztsova et al. (AAMAS'11) showed an algorithm for finding an optimal manipulation under Maximin when the utilities are (1, 0,…, 0)
- Obraztsova and Elkind (IJCAI'11) proved that this problem is NP-hard when the utilities are (1, …, 1, 0)
- What if the utilities are (1, …, 1, 0, …, 0) with k 1's?

Maximum Directed Acyclic Subset (Max-DAS) Problem

- Given digraph G = (W, E) we are asked to find a maximum subset of vertices $S \subseteq W$ which induce acyclic sub-graph of G



Parameterized Complexity

- Classifies problems' complexity as a function of multiple parameters of the input
- Finer scale classification of NP-hard problems
- A parameterized problem is a language $L \subseteq \sum^* \times \mathrm{N}$
- *L* is called fixed-parameter tractable (in FPT) if the question "$(x, k) \in L$?" can be decided in time $f(k) \cdot |x|^{O(1)}$

Main result

- Suppose the utility vector of the manipulator is (1,…,1,0,…,0) with k 1's
- Theorem 1: If there is α–approx. algorithm for Max-DAS problem on graph with k vertices running O(f(k)) time, then there is α–approx. algorithm for MRTB under Maximin running $O(f(k) + (n+m)m^2)$ time
- Corollary 1: MRTB under Maximin can be done in time $O(2^k k^2 + (n+m)m^2)$ (➔ in FPT)
- Corollary 2: When k = O(log(m) + log(n)) then it is in P

Election Graph

- We define the election digraph $G = (W, E)$ as follows:
  - $W = C$
  - $(c, c') \in E$ iff $N(c', c) = S(c')$
- Let $X = \text{argmax}_c\{S(c) \mid u(c) = 1\}$
- Let $H = (X, E')$ be the sub-graph of G induced by vertices X

Proof sketch of the Theorem 1

- First run the algorithm of Obraztsova et al. for utilities vector (1,0,…,0) to minimize the number of winning candidates with utility 0
  - Get $L_1 :: L_2$, where L_1 is some order on X
- Then run Max-DAS approximation algorithm on the digraph H = (X, E'); let $S \subseteq X$ be its output
- Topologically sort the vertices of S: $[x_1,\ldots,x_{|S|}]$
- Return $[x_{|S|},\ldots,x_1] :: L' :: L_2$, where L' is some order on X \ S

An approximation result

- Lemma: If in digraph G = (W, E) there are at least |W|/b vertices having in-degree or out-degree ≤ b’, then we can approximate Max-DAS problem up to the factor of b(b’+1).
- Corollary: If in the digraph H at least |X|/b vertices have in-degree or out-degree ≤ b’, then we can approximate the optimal expected utility up to the factor of b(b’+1).

Approximation algorithm for Max-DAS problem

- $S = \emptyset$
- While $W \neq \emptyset$
  - Let $x = \text{argmin}_{v \in W}\{\min\{d_{in}(v), d_{out}(v)\}\}$
  - $S = S \cup \{x\}$
  - If $d_{in}(x) \leq d_{out}(x)$ then remove x from W together with all the incoming neighbors of x
  - Else remove x from W together with all the outgoing neighbors of x
- Return S

Proof of Lemma – 1/2

- The set S returned by the algorithm induces acyclic sub-graph G’ = (S, E’):
  - Suppose for contradiction there is a cycle T in G’
  - Arrange the vertices of S on a line from left to right in the order of addition to S
  - Let a be the leftmost vertex of T
  - a has an edge to the right and an edge from the right, a contradiction

Proof of Lemma - 2/2

- If at least $|W|/b$ of the vertices have in-degree or out-degree $\leq b'$, then the above algorithm gives $b(b'+1)$-approximation to the Max-DAS problem:
  - When processing the first $|W|/b$ vertices, with every vertex added to S we delete at most $b'+1$ vertices from W.
  - Therefore, we will do at least $|W|/(b(b'+1))$ iterations of the main loop
  - I.e., we will add at least $|W|/(b(b'+1)) \geq OPT/(b(b'+1))$ vertices to S

An exact result

- Theorem 2: If in the digraph H = (X, E') induced by the set $X = \text{argmax}_c\{S(c) \mid u(c) = 1\}$ there is a 2-cycle, then the approximation algorithm attains the optimum
- Proof idea: We prove that when there is a 2-cycle in H, then H is complete
- Therefore, in any permutation of X, there is only one candidate in X whose score increases

Conclusions

- We defined the manipulation with randomized tie-breaking problem
- We proved that when the utilities of the manipulator are (1,…,1,0,…,0) with k 1's, it is in FPT when parameterized by k
- We identified conditions which allow to approximate MRTB up to a constant factor
- We investigated when this problem could be solved exactly and efficiently
- What about general utilities?

Proof Systems and Transformation Games

CoopMAS'10

Joint work with Yoram Bachrach, Michael Wooldridge and Jeff Rosenschein

Transformation Games - Introduction

- Cooperative game
- Players are endowed with
  - initial resources
  - capabilities of deriving certain output resources given certain input resources
- The aim is to generate a particular target resource
- It is achieved by forming coalitions

Example 1 – supply chains

- Firm A – drills for crude oil
- Firm B – produces refined oil from crude oil
- Firm C – packages refined oil in barrels
- Firm D – transforms refined oil into petrol sold to consumers

Questions for supply chains

- Which supply chains are more likely to be formed?
- How should the firms share the revenue?
- Which firm is most important in the supply chain?

Example 2 – cooperative proof systems

- A set of experts tries to prove a complex theorem
- Each expert knows facts and proofs for simpler theorems
- Suppose they can prove the theorem together
- How can they quantify the contribution of each expert?

Notation

- $I = \{a_1,\ldots,a_n\}$ – is the set of agents
- $R = \{r_1,\ldots,r_k\}$ – is the set of resources
- $r_g \in R$ is the goal resource
- $R_i \subseteq R$ is the resource set of a_i
- A transformation is a pair <B, r> where $B \subseteq R$, $r \in R$
- D is the set of all possible transformations over R
- $D_i \subseteq D$ are the transformations of ai

Assumptions

- We do not model resource *quantity*
- We do not model resource *consumption*
- In the basic model there are no *costs* associated with transformations
  - The model extension contains such costs

Transformations

- $\Rightarrow \subseteq 2^I \times R$ – is a relation where $C \Rightarrow r$ means that coalition C can produce resource r
  - Defined recursively
- TG Γ with a goal resource r_g induces a simple TU coalitional game:

$$v_\Gamma(C) = \begin{cases} 1 & \text{if } C \Rightarrow r_g \\ 0 & \text{otherwise} \end{cases}$$

- It is called *Unrestricted* TG (UTG)

Restrictions on TGs

- Definition: A *Transformation restricted* TG (DTG) with goal resource r_g and transformation bound k is the game where coalition C wins if it can derive r_g using at most k transformations, and loses otherwise
- Definition: A *Time limited* TG (TTG) with goal resource r_g and time limit t is the game where coalition C wins if it can derive r_g with time of at most t, and loses otherwise
  - Allows for simultaneous transformations

Results

Problem	UTG	DTG	TTG
CV	P	P (NPH)	P
VETO	P	P (co-NPH)	P
DUMMY	co-NPC	co-NPC(co-NPH)	co-NPC
CORE	P	P (co-NPH)	P
SHAPLEY	co-NPH	co-NPH	co-NPH
BANZHAF	#P-Hard	#P-Hard	#P-Hard

- If the results differ for simple and complex transformations, the results for complex transformations are given in parentheses

TGs with associated costs

- Every transformation d has cost $c(d) \in \Re^+$
- h(C, r) is the minimum cost needed to obtain r from R_C.
  - The sum of trans. costs in the minimum sequence of trans. from R_C to r.
- $v(r_g) \in \Re^+$ is the value of the goal resource
- Definition: TGs *with costs* (CTG) with the goal resource r_g and cost function $c: D \rightarrow \Re^+$ is the game where:

$$v(C) = \max(0, v(r_g) - h(C, r_g))$$

Results for CTGs

Problem	CTGs
Coalitional Value (CV)	P
DUMMY	co-NPC
SHAPLEY	co-NPH
BANZHAF	#P-Hard

Conclusions

- We introduced Transformation Games
- TGs are kind of cooperative games
- Can model cooperative proof systems and supply chains
- We defined some restrictions on TGs
- Studied computational complexity of various related problems
- Extended the model by adding costs to transformations

Future Work

- Extend the model to consumable resources
- More refined solution concepts:
  - Least core
  - Nucleolus
- Uncertain domains: domains where transformations may fail