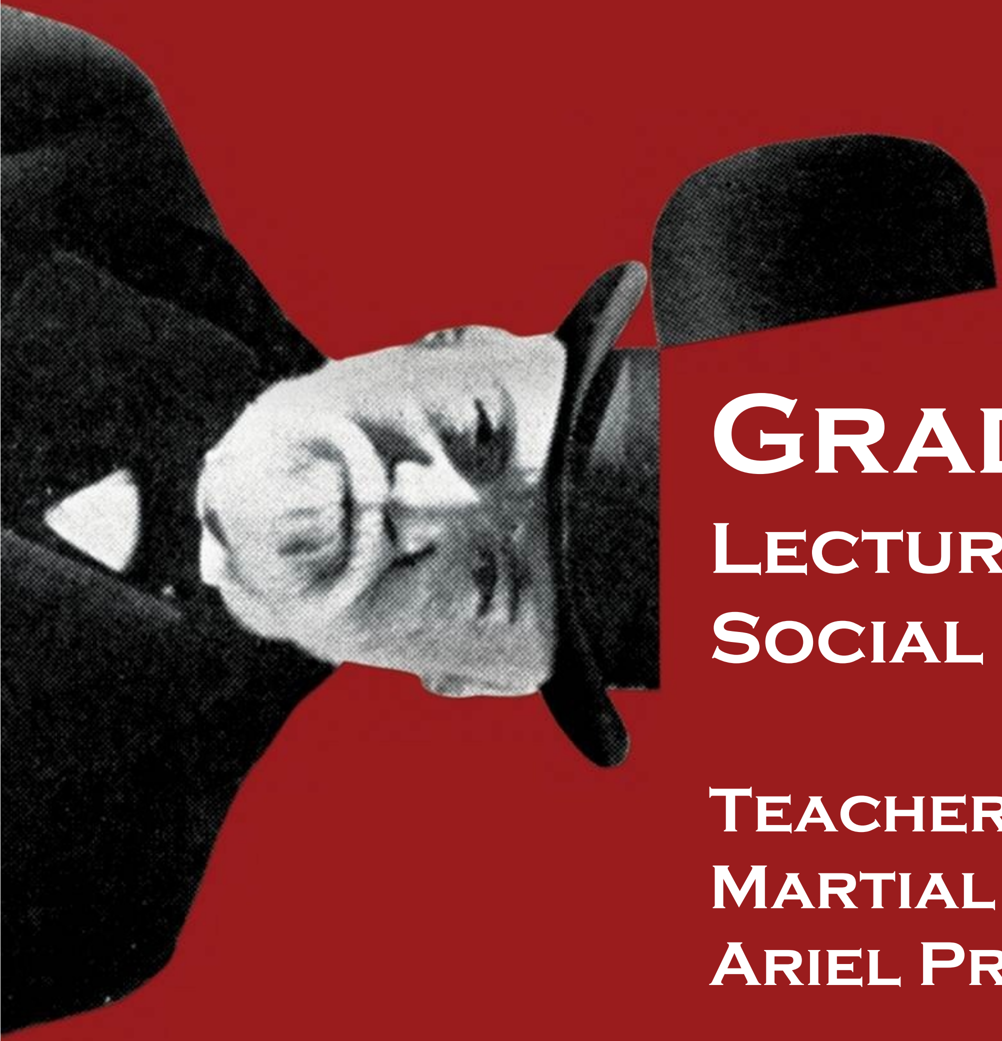


Graduate AI

Lecture 21: Social Choice II

Teachers:
Martial Hebert
Ariel Procaccia (this time)

Reminder: Voting

- Set of *voters* N={1,...,n}
- Set of alternatives A, |A|=m
- Each voter has a ranking over the alternatives
- $x >_i y$ means that voter i prefers x to y
- *Preference profile* = collection of all voters' rankings
- *Voting rule* = function from preference profiles to alternatives

Reminder: Manipulation

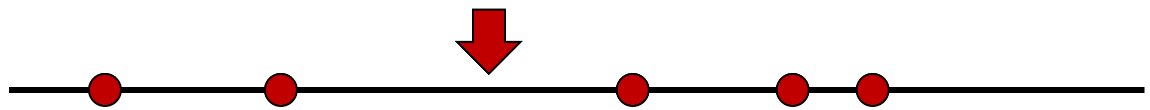
- A voting rule is *strategyproof (SP)* if a voter can never benefit from lying about his preferences:
  $\forall <, \forall i \in N, \forall <'_i, f(<) \geq_i f(<'_i, <_{-i})$
- **Theorem (Gibbard-Satterthwaite)**: If $m \geq 3$ then any voting rule that is SP and onto is dictatorial

Circumventing G-S

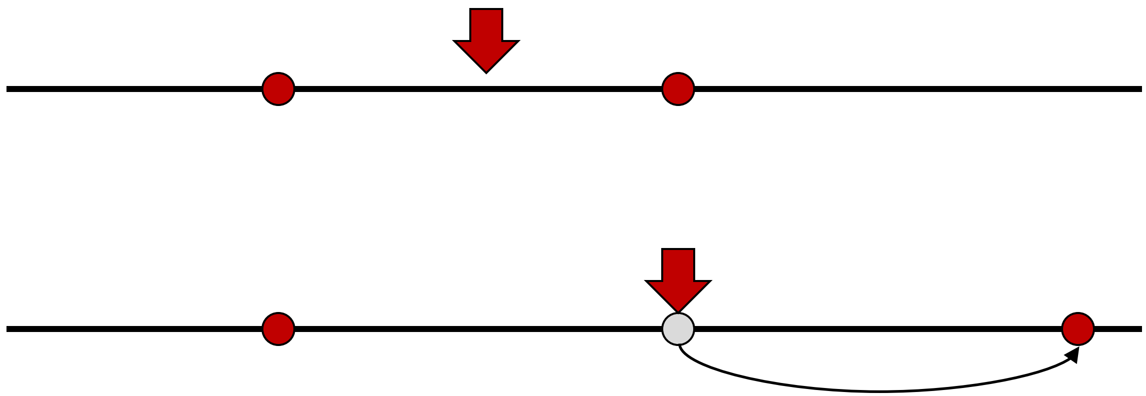
- Restricted preferences
- Money $\Rightarrow$ mechanism design
- Computational complexity

Single peaked preferences

- We want to choose a location for a public good (e.g., library) on a street
- Alternatives = possible locations
- Each voter has an ideal location (peak)
- The closer the library is to a voter's peak, the happier he is
- Suggestion: midpoint



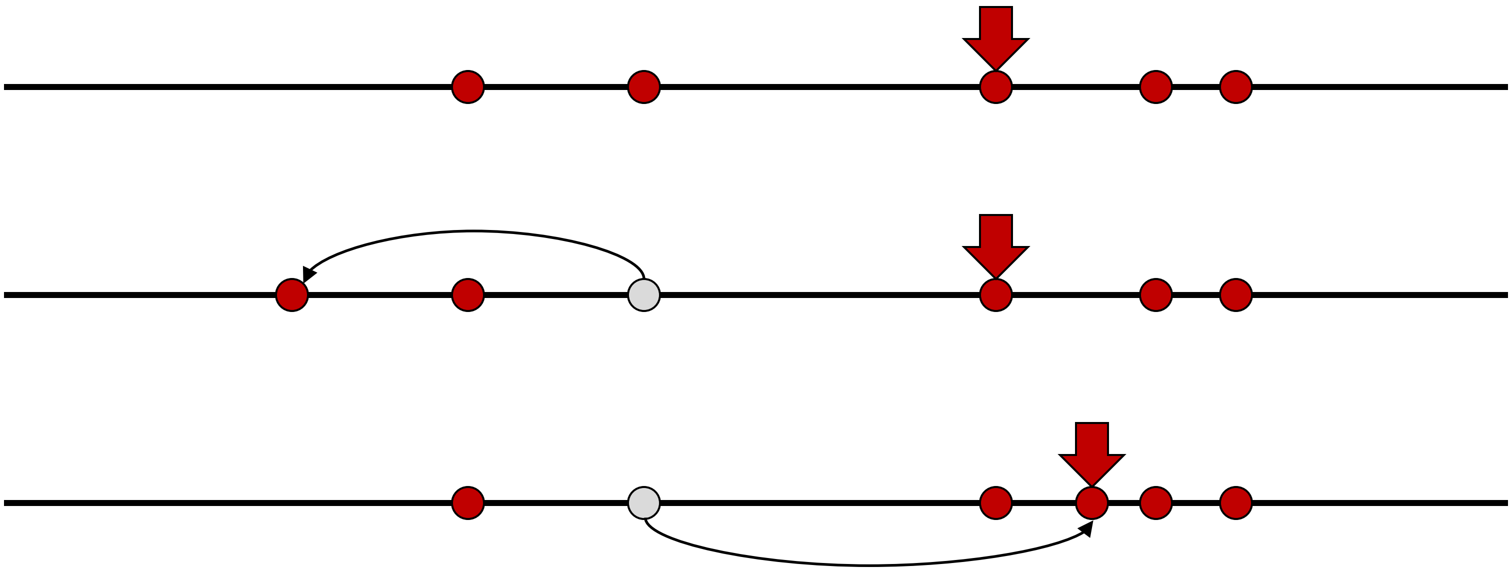
Midpoint is not SP



The median

- Select the median peak
- The median is a Condorcet winner!
- The median is onto
- The median is nondictatorial

The median is SP



Complexity of manipulation

- Manipulation is always possible in theory
- But can we design voting rules where it is difficult in practice?
- Are there "reasonable" voting rules where manipulation is a hard computational problem? [Bartholdi et al., SC&W 1989]

The computational problem

- R-Manipulation problem:
  - Given votes of nonmanipulators and a preferred candidate p
  - Can manipulator cast vote that makes p (uniquely) win under R?
- Example: Borda, p=a

1	2	3
b	b	
a	a	
c	c	
d	d	

1	2	3
b	b	a
a	a	c
c	c	d
d	d	b

A greedy algorithm

- Rank p in first place
- While there are unranked alternatives:
  - If there is an alternative that can be placed in next spot without preventing p from winning, place this alternative
  - Otherwise return false

Example: Borda

1	2	3
b	b	a
a	a	
c	c	
d	d	

1	2	3
b	b	a
a	a	b
c	c	
d	d	

1	2	3
b	b	a
a	a	c
c	c	
d	d	

1	2	3
b	b	a
a	a	c
c	c	b
d	d	

1	2	3
b	b	a
a	a	c
c	c	d
d	d	

1	2	3
b	b	a
a	a	c
c	c	d
d	d	b

Example: Copeland

1	2	3	4	5
a	b	e	e	a
b	a	c	c	
c	d	b	b	
d	e	a	a	
e	c	d	d	

Preference profile

	a	b	c	d	e
a	-	2	3	5	3
b	3	-	2	4	2
c	2	2	-	3	1
d	0	0	1	-	2
e	2	2	3	2	-

Pairwise elections

Example: Copeland

1	2	3	4	5
a	b	e	e	a
b	a	c	c	c
c	d	b	b	
d	e	a	a	
e	c	d	d	

Preference profile

	a	b	c	d	e
a	-	2	3	5	3
b	3	-	2	4	2
c	2	3	-	4	2
d	0	0	1	-	2
e	2	2	3	2	-

Pairwise elections

Example: Copeland

1	2	3	4	5
a	b	e	e	a
b	a	c	c	c
c	d	b	b	d
d	e	a	a	
e	c	d	d	

Preference profile

	a	b	c	d	e
a	-	2	3	5	3
b	3	-	2	4	2
c	2	3	-	4	2
d	0	1	1	-	3
e	2	2	3	2	-

Pairwise elections

Example: Copeland

1	2	3	4	5
a	b	e	e	a
b	a	c	c	c
c	d	b	b	d
d	e	a	a	e
e	c	d	d	

Preference profile

	a	b	c	d	e
a	-	2	3	5	3
b	3	-	2	4	2
c	2	3	-	4	2
d	0	1	1	-	3
e	2	3	3	2	-

Pairwise elections

Example: Copeland

1	2	3	4	5
a	b	e	e	a
b	a	c	c	c
c	d	b	b	d
d	e	a	a	e
e	c	d	d	b

Preference profile

	a	b	c	d	e
a	-	2	3	5	3
b	3	-	2	4	2
c	2	3	-	4	2
d	0	1	1	-	3
e	2	3	3	2	-

Pairwise elections

When does the alg work?

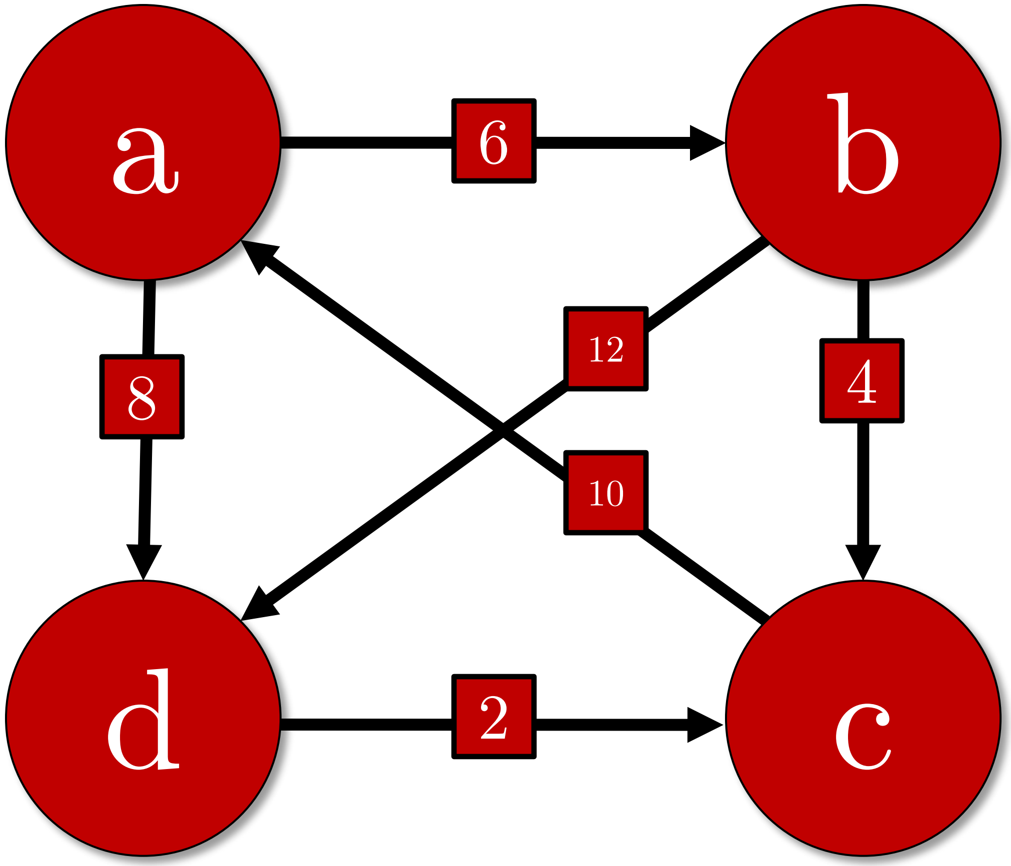
- **Theorem [Bartholdi et al., SCW 89]:** Let R be a rule s.t. $\exists$function $s(<,x)$ such that:
  - For every $<$ chooses a candidate that maximizes $s(<,x)$
  - $\{y\colon y < x\} \subseteq \{y\colon y <' x\} \Rightarrow s(x,<) \leq s(x,<')$

  Then the algorithm always decides R-Manipulation correctly
- Captures:
  - All scoring rules, e.g., Borda
  - Copeland: s is number of pairwise elections x wins
  - Maximin: s is the worst pairwise election of x
- We prove the theorem on the board
- Proof appears in: Bartholdi, Tovey, and Trick. The computational difficulty of manipulating an election. SC&W 1989, Theorem 1 (available on the course website)

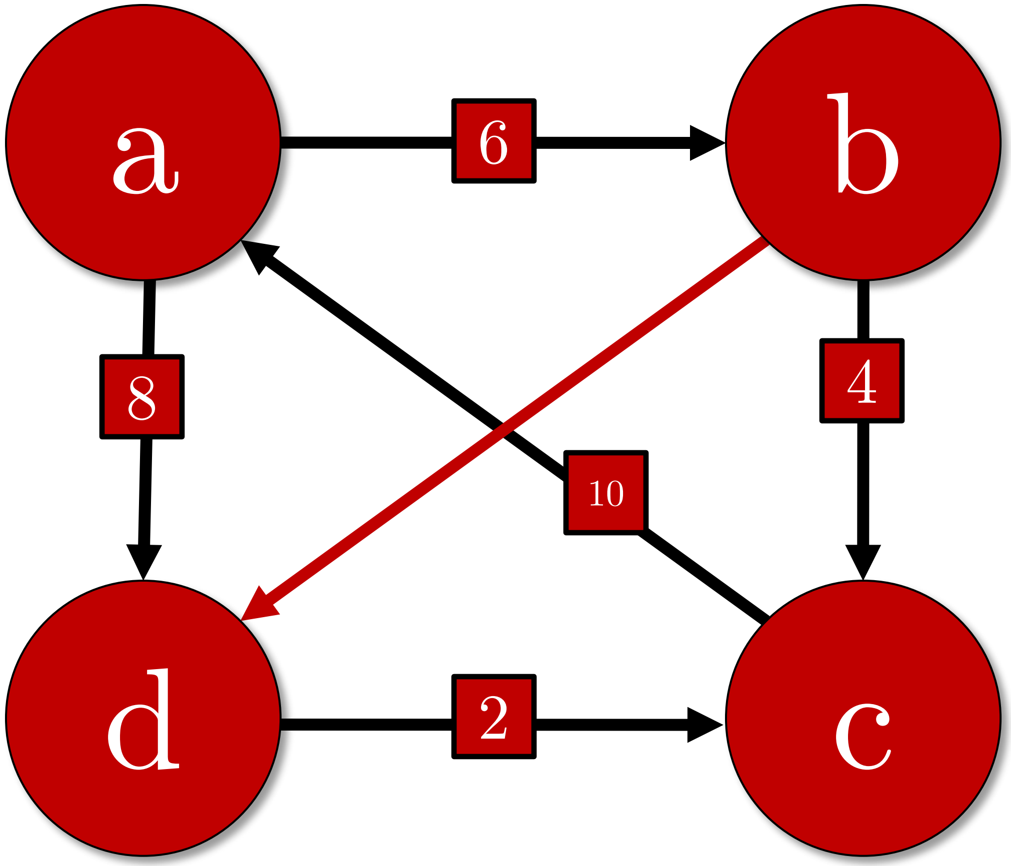
Voting rules that are hard to manipulate

- Natural rules
  - Copeland with second order tie breaking [Bartholdi et al., SCW 89]
  - STV [Bartholdi&Orlin, SCW 91]
  - Ranked Pairs [Xia et al., IJCAI 09]
    - Order pairwise elections by decreasing strength of victory
    - Successively lock in results of pairwise elections unless it leads to cycle
    - Winner is the top ranked candidate in final order
- Can also "tweak" easy to manipulate voting rules [Conitzer&Sandholm, IJCAI 03]

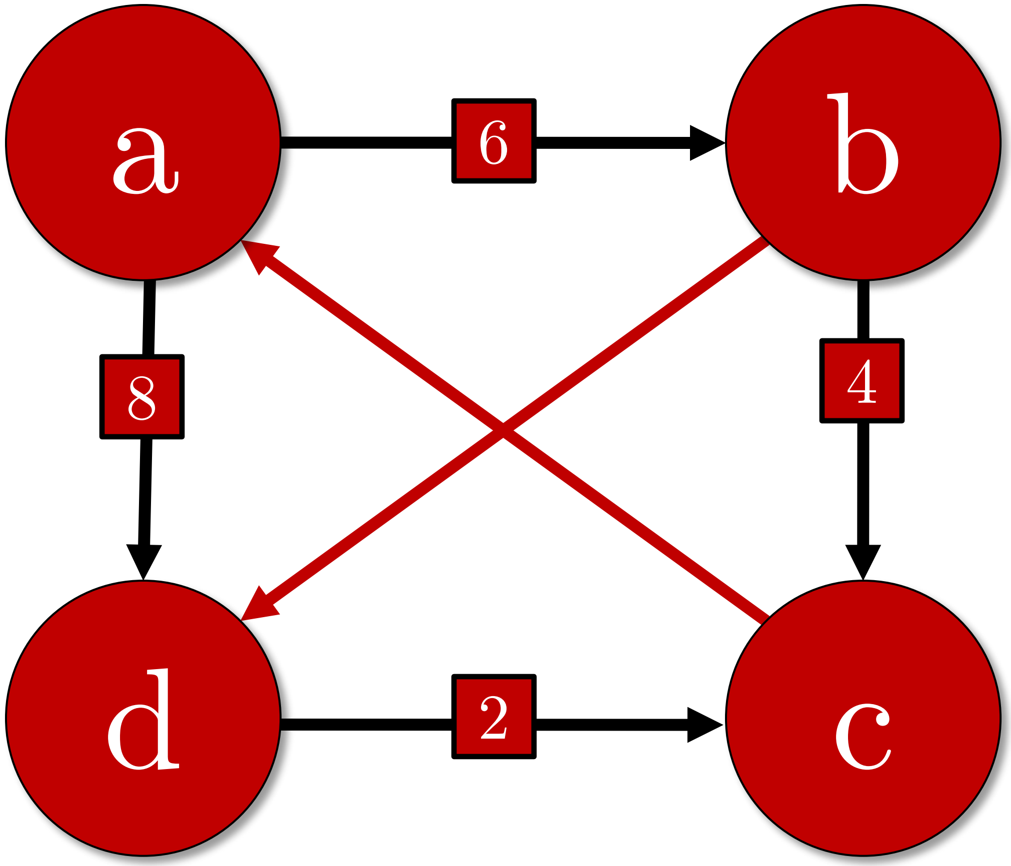
Example: ranked pairs



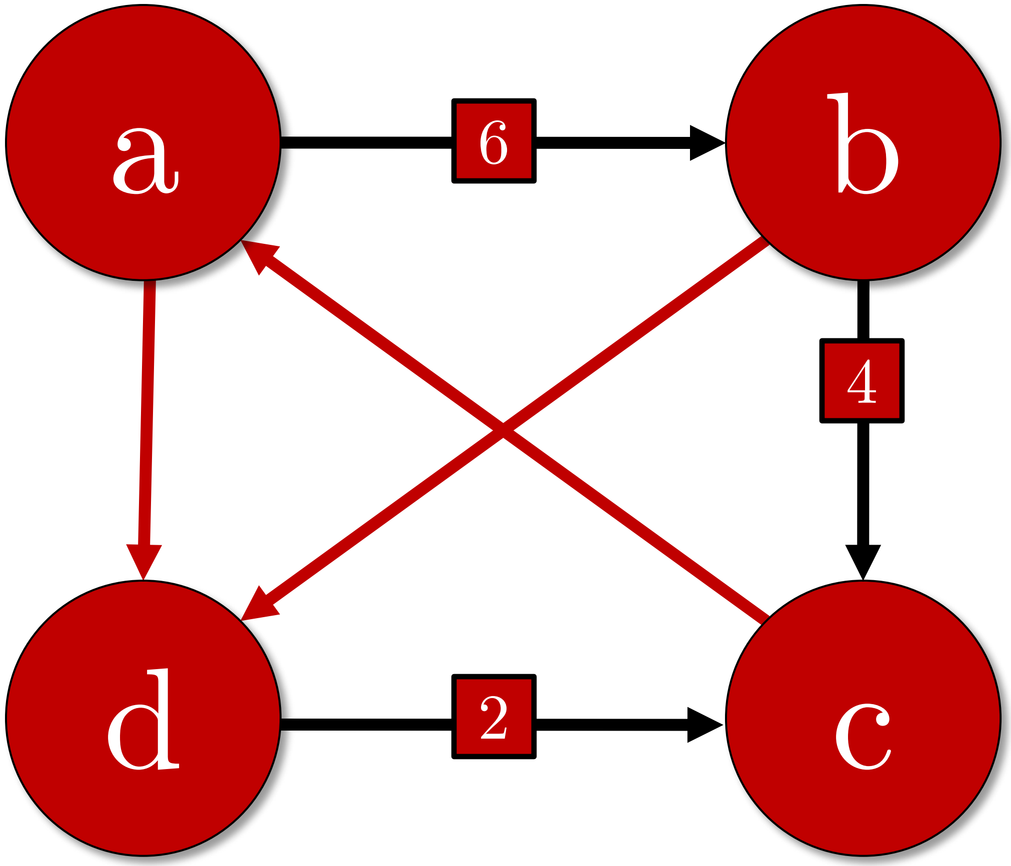
Example: ranked pairs



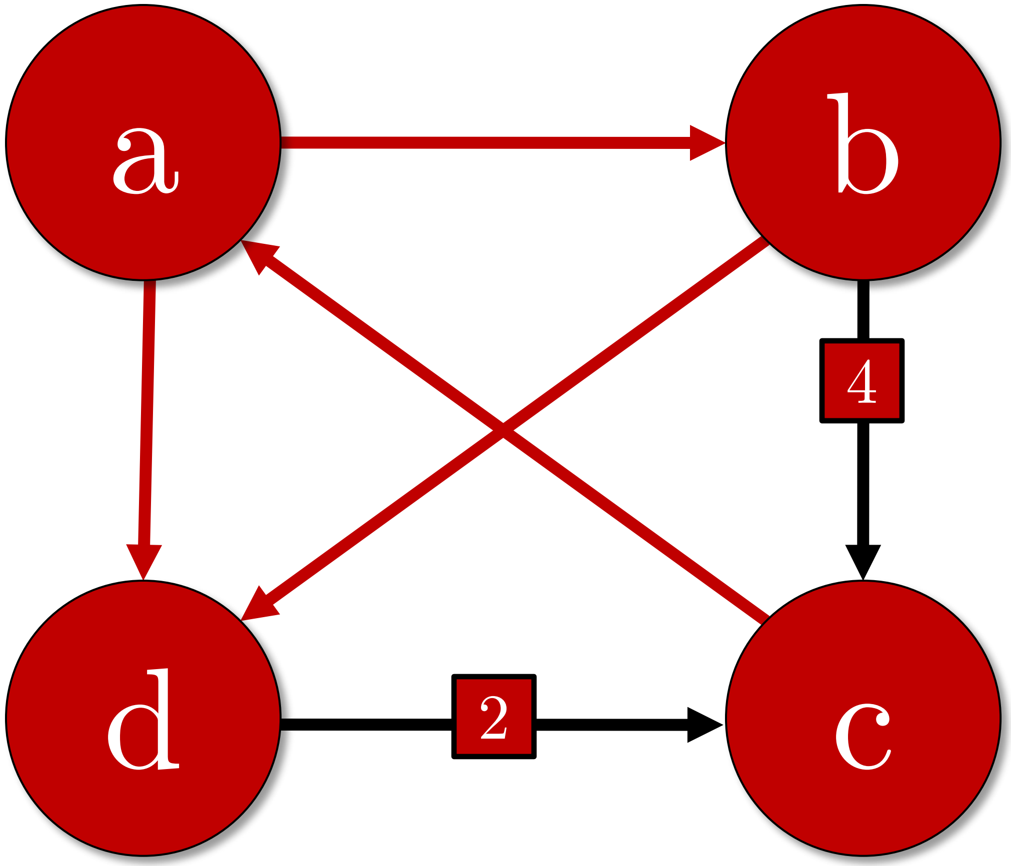
Example: ranked pairs



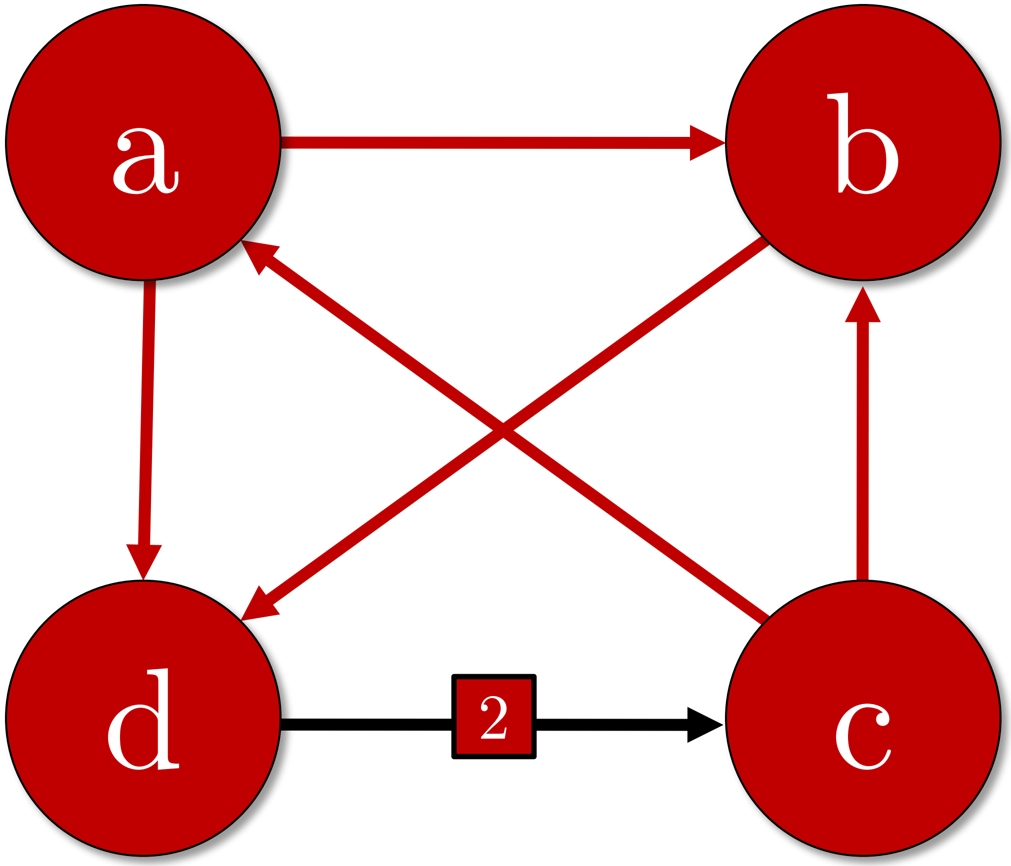
Example: ranked pairs



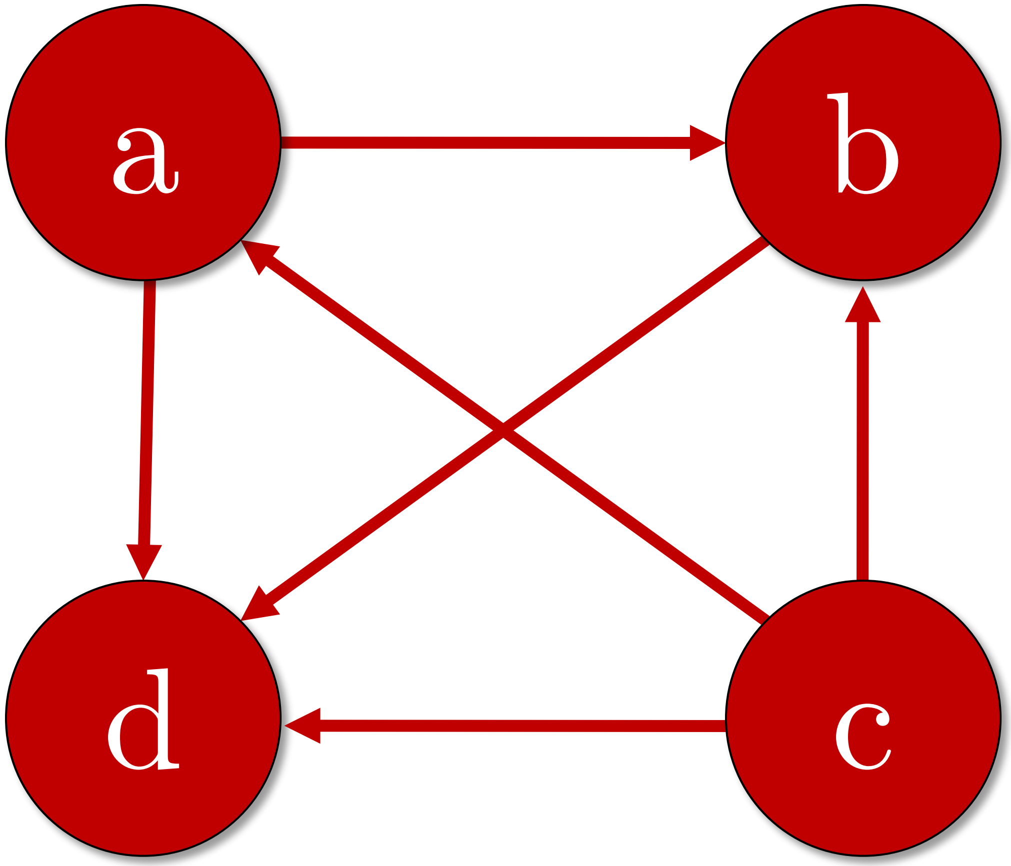
Example: ranked pairs



Example: ranked pairs

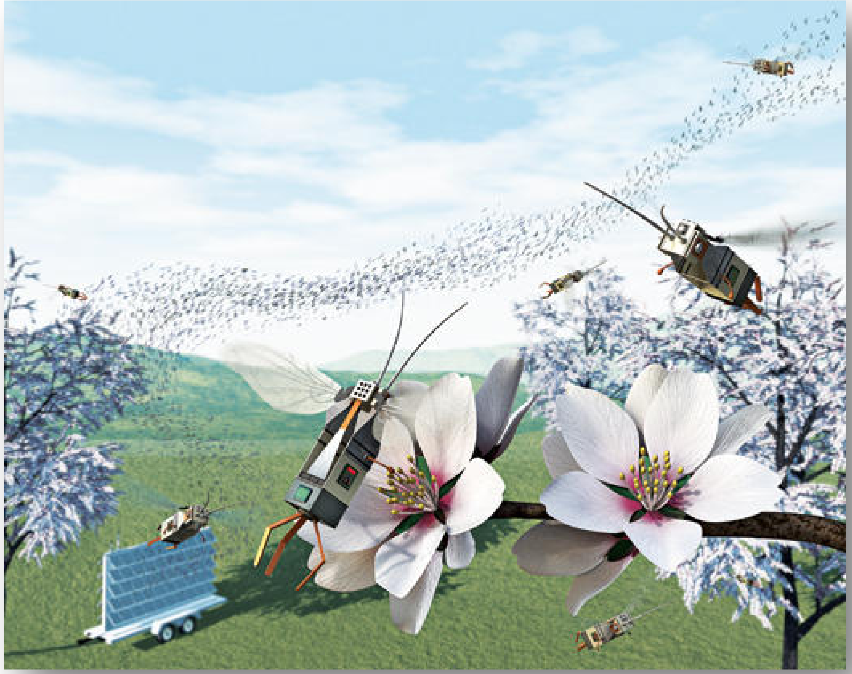


Example: ranked pairs



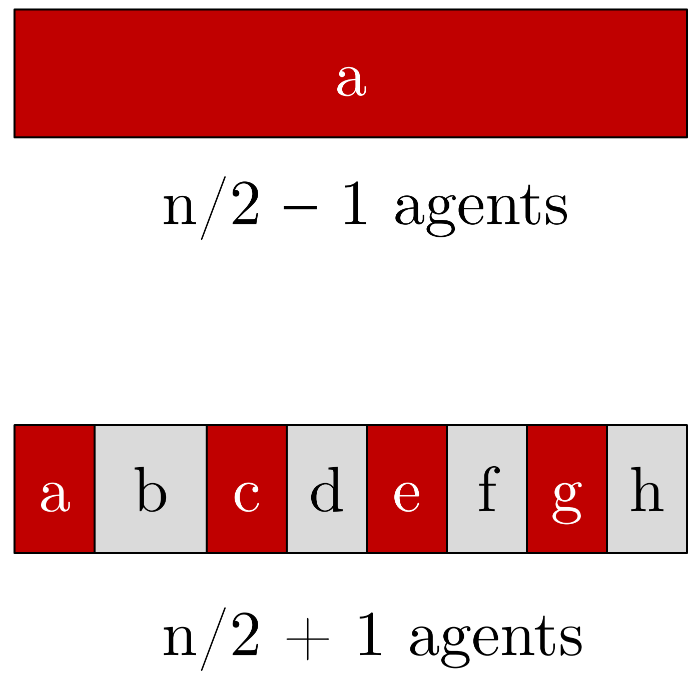
Maximizing social welfare

- Robobees need to decide on a joint plan (alternative)
- Many possible plans
- Each robobee (agent) has a numerical evluation (utility) for each alternative
- Want to maximize sum of utilities = *social welfare*
- Communication is restricted



Maximizing social welfare

- Approach 1: communicate utilities
    - May be infeasible
- Approach 2: each agent votes for favorite alternative (plurality)
    - logm bits per agent
    - May select a bad alternative



Maximizing social welfare

- Approach 3: each agent votes for an alternative with probability proportional to its utility
- **Theorem (informal):** if $n=\omega(m\log m)$ then this approach gives a $1+o(1)$ approximation for the optimal social welfare in expectation [Caragiannis+P, AIJ 2011]

Voting rules as MLEs

- Choose 8 RNA designs to synthesize
- Assume that each player provides a ranking
- Each pair of designs is ranked correctly with probability p>1/2

Voting rules as MLEs

- Goal: choose a set of 8 designs that maximizes the probability of containing the best design
- **Theorem:** if p is sufficiently close to $\frac{1}{2}$ then the set of 8 designs with highest Borda scores is such a set [P+Reddy+Shah]