A quick tour of computational social choice

Where Artificial Intelligence meets collective decision making

Sylvain Bouveret

LIG – Grenoble INP

Journées inaugurales du Pré-GDR IA
Montpellier, 13 et 14 juin 2016

Outline

Outline

Social choice

Social choice theory focuses on the analysis of collective decision making methods.

Social choice

Social choice theory focuses on the analysis of collective decision making methods.

- A set of alternatives $\mathcal{O}$

Social choice

Social choice theory focuses on the analysis of collective decision making methods.

- A set of alternatives $\mathcal{O}$
- A set of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$...

Social choice

Social choice theory focuses on the analysis of collective decision making methods.

- A set of alternatives $\mathcal{O}$
- A set of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$...
- ...Expressing opinions over the alternatives.

Social choice

Social choice theory focuses on the analysis of collective decision making methods.

- A set of alternatives $\mathcal{O}$
- A set of agents $\mathcal{A} = \{a_1, \ldots, a_n\}$...
- ...Expressing opinions over the alternatives.

$$\Downarrow$$

Collective opinion, choice of an alternative...

Voting

We have to elect a representative from a set of m candidates on which the n voters have diverse preferences.

Voting

We have to elect a representative from a set of m candidates on which the n voters have diverse preferences.

- Alternatives: candidates
- Agents: voters
- Preferences: ballots (usually linear orders)

Voting rules

- $X = \{a, b, c, \ldots\}$ set of candidates
- $N = \{1, \ldots, n\}$ set of voters
- each voter reports a ranking $\succ_i$ over candidates;
- voting profile: $P = \langle \succ_1, \ldots, \succ_n \rangle$

$$\begin{array}{ll} \text{voters } 1, 2, 3, 4: & c \succ b \succ d \succ a \\ \text{voters } 5, 6, 7, 8: & a \succ b \succ d \succ c \\ \text{voter } 9: & c \succ a \succ b \succ d \end{array}$$

Voting rules

- $X = \{a, b, c, \ldots\}$ set of candidates
- $N = \{1, \ldots, n\}$ set of voters
- each voter reports a ranking $\succ_i$ over candidates;
- voting profile: $P = \langle \succ_1, \ldots, \succ_n \rangle$

$$
\begin{array}{ll}
\text{voters } 1, 2, 3, 4: & c \succ b \succ d \succ a \\
\text{voters } 5, 6, 7, 8: & a \succ b \succ d \succ c \\
\text{voter } 9: & c \succ a \succ b \succ d
\end{array}
$$

plurality rule: the winner is the candidate ranked first by the largest number of voters

$$plurality(P) = c$$

Voting rules

- $X = \{a, b, c, \ldots\}$ set of candidates
- $N = \{1, \ldots, n\}$ set of voters
- each voter reports a ranking $\succ_i$ over candidates;
- voting profile: $P = \langle \succ_1, \ldots, \succ_n \rangle$

$$\begin{array}{ll} \text{voters } 1, 2, 3, 4: & c \succ b \succ d \succ a \\ \text{voters } 5, 6, 7, 8: & a \succ b \succ d \succ c \\ \text{voter } 9: & c \succ a \succ b \succ d \end{array}$$

Borda rule: a candidate ranked 1st / 2nd / 3rd / last in a vote gets 3 / 2 / 1 / 0 points. The candidate with maximum total number of points wins.

$$a \mapsto (4 \times 3) + 2 = 14 \quad b \mapsto 17 \quad c \mapsto 15 \quad d \mapsto 8$$

$$Borda(P) = b$$

Voting rules

- $X = \{a, b, c, \ldots\}$ set of candidates
- $N = \{1, \ldots, n\}$ set of voters
- each voter reports a ranking $\succ_i$ over candidates;
- voting profile: $P = \langle \succ_1, \ldots, \succ_n \rangle$

$$\begin{array}{ll} \text{voters } 1, 2, 3, 4: & c \succ b \succ d \succ a \\ \text{voters } 5, 6, 7, 8: & a \succ b \succ d \succ c \\ \text{voter } 9: & c \succ a \succ b \succ d \end{array}$$

many other rules!

Fair Division – Cake-Cutting

We have to divide a rectangular heterogeneous cake among n *agents having different valuations about parts of the cake.*

Fair Division – Cake-Cutting

We have to divide a rectangular heterogeneous cake among n *agents having different valuations about parts of the cake.*

Fair Division – Cake-Cutting

We have to divide a rectangular heterogeneous cake among n *agents having different valuations about parts of the cake.*

- Alternatives: allocations of the cake
- Agents: cake eaters
- Preferences: valuation functions (generally additive)

Protocols

Usually, we care about:

- Proportionality: each agent feels that her share is worth at least $\frac{1}{n}$ of the cake.
- Envy-freeness: each agent feels that her share is better than the share of any other agent.

Protocols

Usually, we care about:

- Proportionality: each agent feels that her share is worth at least $\frac{1}{n}$ of the cake.
- Envy-freeness: each agent feels that her share is better than the share of any other agent.

2 agents: I cut, you choose.

- Agent 1 cuts the cake into two pieces of equal value to her.
- Agent 2 chooses.

Guarantees envy-freeness and proportionality.

More than two agents

The Banach-Knaster Last Diminisher procedure:

1. Agent 1 cuts off a piece (she estimates to be worth $\frac{1}{n}$)

More than two agents

The Banach-Knaster Last Diminisher procedure:

1. Agent 1 cuts off a piece (she estimates to be worth $\frac{1}{n}$)
2. Each agent from 2 to n have the choice either to pass, or to trim the piece further down.

More than two agents

The Banach-Knaster Last Diminisher procedure:

1. Agent 1 cuts off a piece (she estimates to be worth $\frac{1}{n}$)
2. Each agent from 2 to n have the choice either to pass, or to trim the piece further down.
3. The last agent having cut the piece takes it and leaves the game.

More than two agents

The Banach-Knaster Last Diminisher procedure:

1. Agent 1 cuts off a piece (she estimates to be worth $\frac{1}{n}$)
2. Each agent from 2 to n have the choice either to pass, or to trim the piece further down.
3. The last agent having cut the piece takes it and leaves the game.
4. The game starts again with the remaining agents and the rest of the cake (including trimmings).

More than two agents

The Banach-Knaster Last Diminisher procedure:

1. Agent 1 cuts off a piece (she estimates to be worth $\frac{1}{n}$)
2. Each agent from 2 to n have the choice either to pass, or to trim the piece further down.
3. The last agent having cut the piece takes it and leaves the game.
4. The game starts again with the remaining agents and the rest of the cake (including trimmings).

More than two agents

The Banach-Knaster Last Diminisher procedure:

1. Agent 1 cuts off a piece (she estimates to be worth $\frac{1}{n}$)
2. Each agent from 2 to n have the choice either to pass, or to trim the piece further down.
3. The last agent having cut the piece takes it and leaves the game.
4. The game starts again with the remaining agents and the rest of the cake (including trimmings).

Guarantees proportionality (of course not envy-freeness).

Fair Division – Indivisible Goods

We have to allocate a set of indivisible objects to n agents having different valuations about them.

Fair Division – Indivisible Goods

We have to allocate a set of indivisible objects to n agents having different valuations about them.

- Alternatives: allocations of the objects
- Agents: object consumers
- Preferences: valuation functions / orders,...

Fair Division – Indivisible Goods

We have to allocate a set of indivisible objects to n agents having different valuations about them.

- Alternatives: allocations of the objects
- Agents: object consumers
- Preferences: valuation functions / orders,...

We will come back to that in more details later.

Matching

We have to match agents from a group S_1 *to agents from a group* S_2*. Agents from* S_1 *have preferences over agents from* S_2*, and vice-versa.*

Matching

We have to match agents from a group S_1 to agents from a group S_2. Agents from S_1 have preferences over agents from S_2, and vice-versa.

Examples:

- Matching students to schools (one-to-many matching)
- Matching students to projects (many-to-many matching)
- Matching men to women – stable marriage (one-to-one matching)

The Stable Marriage Problem

- n men and n women
- each man has a linear preference order over women, and vice versa.
- We look for a stable marriage.

The Stable Marriage Problem

- n men and n women
- each man has a linear preference order over women, and vice versa.
- We look for a stable marriage.

The Gale-Shapley algorithm (1962):

- Each man who is not yet engaged proposes to his favourite women he has not yet proposed to.
- Each woman picks her favourite among all the proposal she has and the man she is currently engaged with.
- Loop until everyone is engaged.

Coalition Formation

n agents have to form groups. Each agent has preferences over the other agents.

Coalition Formation

n agents have to form groups. Each agent has preferences over the other agents.

- Alternatives: valid partitions of the participants.
- Agents: participants.
- Preferences: usually numerical (additive) preferences on the other participants.

Coalition Formation

n agents have to form groups. Each agent has preferences over the other agents.

- Alternatives: valid partitions of the participants.
- Agents: participants.
- Preferences: usually numerical (additive) preferences on the other participants.

Generalization of the matching problem. Usually we look for stable coalitions (hedonic games), or collectively optimal ones.

Judgment Aggregation

We have to make a judgment over a set of logically interdependent issues. Each agent n is an independent judge who has (consistent) opinions about these issues.

- Alternatives: logically interdependent issues
- Agents: judges
- Preferences: usually approval (yes / no) opinions.

Paradox of Judgment Aggregation

- Instructions from IJCAI-ECAI-2018 PC chair: accept a paper if and only if it is original and technically valid
- Accept $\leftrightarrow$ Original $\wedge$ Valid

	Original?	Valid?	Accept?
Reviewer 1	Yes	Yes	Yes
Reviewer 2	Yes	No	No
Reviewer 3	No	Yes	No
majority	Yes	Yes	No

Paradox of Judgment Aggregation

- Instructions from IJCAI-ECAI-2018 PC chair: accept a paper if and only if it is original and technically valid
- Accept $\leftrightarrow$ Original $\wedge$ Valid

	Original?	Valid?	Accept?
Reviewer 1	Yes	Yes	Yes
Reviewer 2	Yes	No	No
Reviewer 3	No	Yes	No
majority	Yes	Yes	No

- (Metareview). Your paper was judged to be original and technically valid. However, we decided to reject it.
- Judgment aggregation: aggregate opinions about logically interrelated issues...

Paradox of Judgment Aggregation

- Instructions from IJCAI-ECAI-2018 PC chair: accept a paper if and only if it is original and technically valid
- Accept $\leftrightarrow$ Original $\wedge$ Valid

	Original?	Valid?	Accept?
Reviewer 1	Yes	Yes	Yes
Reviewer 2	Yes	No	No
Reviewer 3	No	Yes	No
majority	Yes	Yes	No

- (Metareview). Your paper was judged to be original and technically valid. However, we decided to reject it.
- Judgment aggregation: aggregate opinions about logically interrelated issues… in a logically consistent way.
- Strong links to nonmonotonic reasoning, belief merging, inconsistency handling.

Social Choice Everywhere

- Assigning courses to students
- Electing a political representative (e.g. the head of the Pré-GDR…)
- Choosing a collective meeting date
- Choosing the future name for a region
- Electing the winner of the Eurovision song contest
- Scheduling the workload of a team of workers
- Matching patients with hospitals
- Diving a piece of land
- Forming teams
- Choosing the place for a common facility
- …

Outline

Early ages

- From Ancient Greece and India: Aristotle, Chânakya...
- ...To the late XVIIIth century:
  - Condorcet
  - Borda
- And the British philosophical roots of utilitarianism: Bentham, Stuart Mill...

Birth of Modern Social Choice

- Arrow's theorem (1951):

With at least 3 alternatives, an aggregation function satisfies unanimity and independence of irrelevant alternatives if and only if it is a dictatorship.

Birth of Modern Social Choice

- Arrow's theorem (1951):

  With at least 3 alternatives, an aggregation function satisfies unanimity and independence of irrelevant alternatives if and only if it is a dictatorship.

- Results are mainly axiomatic (economics/mathematics)
- Impossibility theorems: incompatibility of a small set of seemingly innocuous conditions, like Arrow's theorem.
- Computational issues are neglected so far.

Where Computation Comes into Play

- Around the 50's: protocols for fair division (*e.g.* Banach-Knaster) $\rightsquigarrow$ algorithms?
- Early 80's: combinatorial auctions
- Early 90's: computer scientists start studying computational issues in social choice (complexity of voting...)
- 2006: First COMSOC Workshop
- As of 2016: a very active community, well represented in AAMAS, IJCAI, AAAI, ECAI...

Computational social choice

COMSOC $\approx$ Social Choice $\cap$ Computer Science

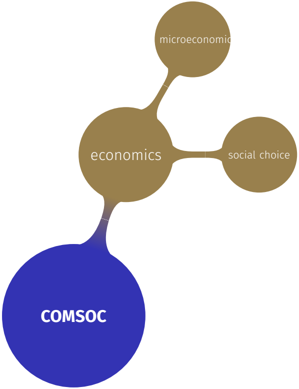
Computational social choice

COMSOC $\approx$ Social Choice $\cap$ Computer Science

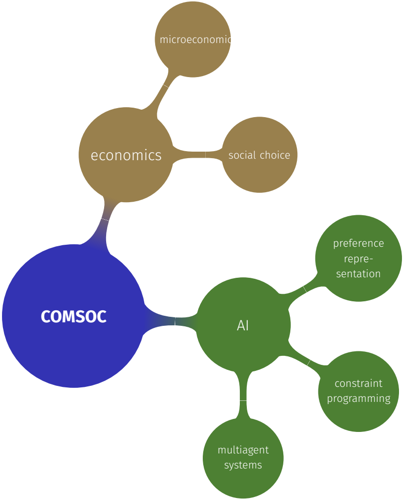
1. Use techniques from economics to solve problems in IT (network sharing, job allocation...)
2. Use techniques from CS to analyze and solve economical problems (complexity of voting procedures, compact preference representation...)

An Interdisciplinary Domain

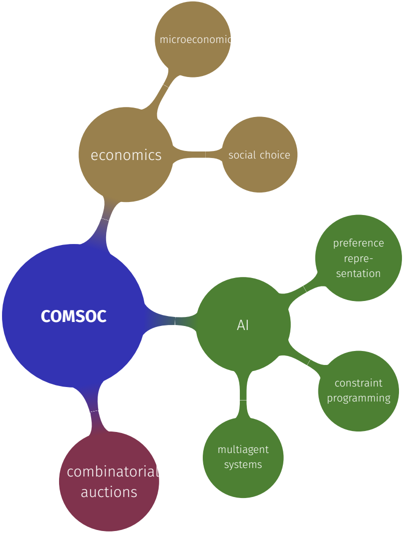
An Interdisciplinary Domain



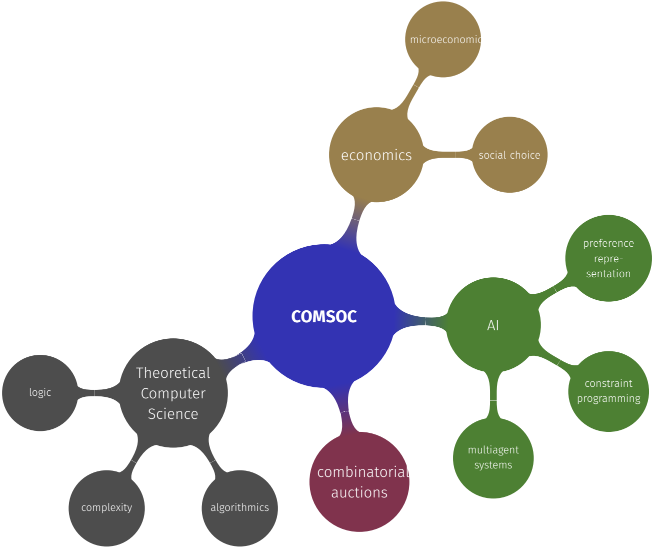
An Interdisciplinary Domain



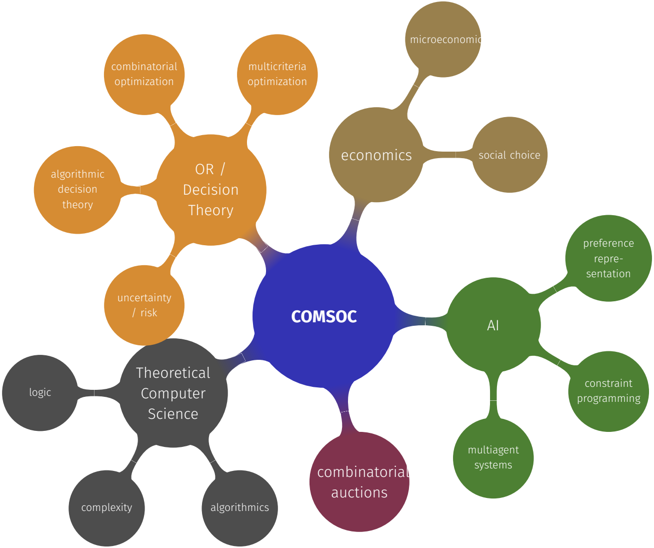
An Interdisciplinary Domain



An Interdisciplinary Domain



An Interdisciplinary Domain



Outline

The Condorcet Principle

3 voters:

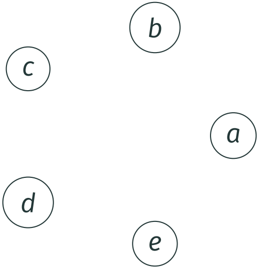
$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$

The Condorcet Principle

3 voters:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$

Run a tournament between the candidates (pairwise comparisons)

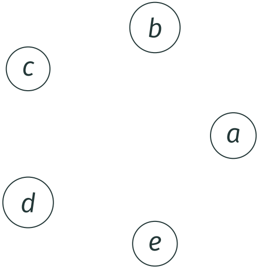


The Condorcet Principle

3 voters:

$$\mathbf{a} \succ b \succ d \succ c \succ e$$
$$b \succ \mathbf{a} \succ e \succ d \succ c$$
$$c \succ e \succ \mathbf{a} \succ b \succ d$$

Run a tournament between the candidates (pairwise comparisons)

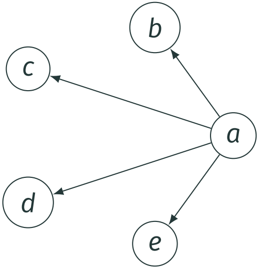


The Condorcet Principle

3 voters:

$$\mathbf{a} \succ b \succ d \succ c \succ e$$
$$b \succ \mathbf{a} \succ e \succ d \succ c$$
$$c \succ e \succ \mathbf{a} \succ b \succ d$$

Run a tournament between the candidates (pairwise comparisons)

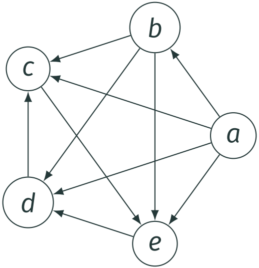


The Condorcet Principle

3 voters:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$

Run a tournament between the candidates (pairwise comparisons)

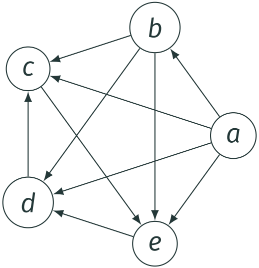


The Condorcet Principle

3 voters:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$

Run a tournament between the candidates (pairwise comparisons)



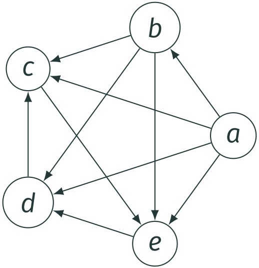
The Condorcet Winner is the candidate that wins against all the other candidates

The Condorcet Principle

3 voters:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$

Run a tournament between the candidates (pairwise comparisons)



The Condorcet Winner is the candidate that wins against all the other candidates $\rightarrow a$

Dodgson rule

However, preferences may cycle and the Condorcet Winner may not exist.
Condorcet-consistent rules elect the Condorcet Winner when it exists.

Dodgson rule

However, preferences may cycle and the Condorcet Winner may not exist.
Condorcet-consistent rules elect the Condorcet Winner when it exists.
The Dodgson rule:

> *If a Condorcet Winner exists, elect it. Otherwise compute for each candidate c the number of adjacent swaps in the individual preferences required to make c a Condorcet Winner. Elect the candidate that minimizes that number.*

Dodgson rule

However, preferences may cycle and the Condorcet Winner may not exist.
Condorcet-consistent rules elect the Condorcet Winner when it exists.
The Dodgson rule:

> *If a Condorcet Winner exists, elect it. Otherwise compute for each candidate* c *the number of adjacent swaps in the individual preferences required to make* c *a Condorcet Winner. Elect the candidate that minimizes that number.*

Theorem (Hemaspaandra *et al.*, 1997)

Winner determination for Dodgson rule in complete for parallel access to NP.

Manipulating Borda

- Borda rule
- a single voter hasn't voted yet
  - 4 voters so far:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$
$$d \succ c \succ b \succ a \succ e$$

  - Current Borda scores

$$a \mapsto 10 \quad b \mapsto 10 \quad c \mapsto 8 \quad d \mapsto 7 \quad e \mapsto 5$$

Can the last voter find a vote so that the winner is … a?

Manipulating Borda

- Borda rule
- a single voter hasn't voted yet
  - 4 voters so far:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$
$$d \succ c \succ b \succ a \succ e$$

  - Current Borda scores

$$a \mapsto 10 \quad b \mapsto 10 \quad c \mapsto 8 \quad d \mapsto 7 \quad e \mapsto 5$$

Can the last voter find a vote so that the winner is … a?

- $a \succ \dots$
- yes

Manipulating Borda

- Borda rule
- a single voter hasn't voted yet
  - 4 voters so far:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$
$$d \succ c \succ b \succ a \succ e$$

  - Current Borda scores

$$a \mapsto 10 \quad b \mapsto 10 \quad c \mapsto 8 \quad d \mapsto 7 \quad e \mapsto 5$$

Can the last voter find a vote so that the winner is … c?

Manipulating Borda

- Borda rule
- a single voter hasn't voted yet
  - 4 voters so far:

$$a \succ b \succ d \succ c \succ e$$
$$b \succ a \succ e \succ d \succ c$$
$$c \succ e \succ a \succ b \succ d$$
$$d \succ c \succ b \succ a \succ e$$

  - Current Borda scores

$$a \mapsto 10 \quad b \mapsto 10 \quad c \mapsto 8 \quad d \mapsto 7 \quad e \mapsto 5$$

Can the last voter find a vote so that the winner is ... c?

- $c \succ e \succ d \succ b \succ a$
- scores: $c \mapsto 12$, $a \mapsto 10$, $b \mapsto 11$, $d \mapsto 9$, $e \mapsto 8$
- yes

Manipulating Borda: two voters

- Two voters haven't voted yet
- Borda rule
- Tie-breaking priority $a > b > c > d > e > f$.
- Current Borda scores:

$$a \mapsto 12 \quad b \mapsto 10 \quad c \mapsto 9 \quad d \mapsto 9 \quad e \mapsto 4 \quad f \mapsto 1$$

- Do the last two voters have a constructive manipulation for e?
- A simple greedy algorithm like before does not work.

Manipulation of the Borda rule

Existence of a manipulation for the Borda rule:

- for a single voter : in P
  - Bartholdi, Tovey & Trick, *Social Choice and Welfare*, 89
- for a coalition of at least two voters : NP-complete
  - Betzler, Niedermeyer & Woeginger, IJCAI-11
  - Davies, Katsirelos, Narodytska & Walsh, AAAI-11
- Lots of results of this kind

Complexity and Manipulation

Is there a better rule than Borda that prevents manipulation?

Complexity and Manipulation

Is there a better rule than Borda that prevents manipulation?

Theorem (Gibbard-Satterthwaite, 1973/75)

All resolute and surjective voting rules over more than 3 candidates are either dictatorial or manipulable

Complexity and Manipulation

Is there a better rule than Borda that prevents manipulation?

Theorem (Gibbard-Satterthwaite, 1973/75)

All resolute and surjective voting rules over more than 3 candidates are either dictatorial or manipulable

But computational complexity can be seen as a barrier to manipulation.

Complexity and Manipulation

Is there a better rule than Borda that prevents manipulation?

Theorem (Gibbard-Satterthwaite, 1973/75)

All resolute and surjective voting rules over more than 3 candidates are either dictatorial or manipulable

But computational complexity can be seen as a barrier to manipulation.
Observation: worst-case complexity, under complete knowledge ($\rightarrow$ in practice?)

Voting in Combinatorial Domains

Combinatorial domains in voting: multiple referendums, multi-winner (*e.g.* committee) election...

Voting in Combinatorial Domains

Combinatorial domains in voting: multiple referendums, multi-winner (*e.g.* committee) election...

Example

2 binary variables:

- S (build a new swimming pool)
- T (build a new tennis court)

voters 1 and 2	$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$
voters 3 and 4	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$
voter 5	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

Voting in Combinatorial Domains

Combinatorial domains in voting: multiple referendums, multi-winner (*e.g.* committee) election...

Example

2 binary variables:

- S (build a new swimming pool)
- T (build a new tennis court)

voters 1 and 2	$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$
voters 3 and 4	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$
voter 5	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

A naive solution: don't bother and vote separately on each variable.

$\Rightarrow$ multiple election paradoxes

Multiple Election Paradoxes

voters 1 and 2	$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$
voters 3 and 4	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$
voter 5	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

Multiple Election Paradoxes

voters 1 and 2	$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$
voters 3 and 4	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$
voter 5	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

Problem 1: voters 1-4 feel ill at ease reporting a preference on $\{S, \bar{S}\}$ and $\{T, \bar{T}\}$

Multiple Election Paradoxes

voters 1 and 2	$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$
voters 3 and 4	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$
voter 5	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

Problem 1: voters 1-4 feel ill at ease reporting a preference on $\{S, \bar{S}\}$ and $\{T, \bar{T}\}$

Problem 2: suppose they do so by an "optimistic" projection

- voters 1, 2 and 5: S; voters 3 and 4: $\bar{S} \Rightarrow$ decision = S;
- voters 3,4 and 5: T; voters 1 and 2: $\bar{T} \Rightarrow$ decision = T.

Alternative ST is chosen although it is the worst alternative for all but one voter.

Voting and CP-nets: aggregating CP-nets

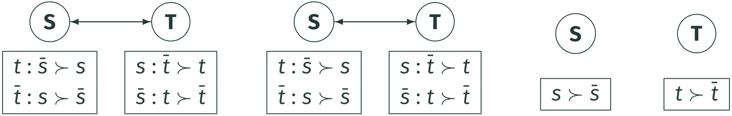
First solution: use a compact preference representation language and aggregate the formulas

Voting and CP-nets: aggregating CP-nets

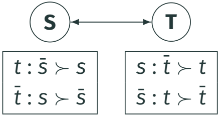
First solution: use a compact preference representation language and aggregate the formulas

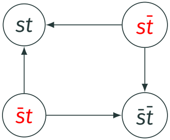
Example (Swimming pool and tennis)

2 voters	2 voters	1 voter
$s\bar{t} \succ \bar{s}t \succ \bar{s}\bar{t} \succ st$	$\bar{s}t \succ s\bar{t} \succ \bar{s}\bar{t} \succ st$	$st \succ \bar{s}t \succ s\bar{t} \succ \bar{s}\bar{t}$



Aggregate locally (by majority) for each pair of adjacent outcomes:





Voting and CP-nets: aggregating CP-nets

+ always applicable, because any preference relation is compatible with some CP-net (possibly with cyclic dependencies).
– elicitation cost: in the worst case, exponential number of queries to each voter
– computation cost: dominance in CP-nets with cyclic dependencies is PSPACE-complete
– there might be no winner; there might be several winners

[Xia et al., 2008, Conitzer et al., 2011, Li et al., 2011]

Voting and CP-nets: sequential voting

Assumption: there exists an order on variables, say $x_1 > \ldots > x_p$, such that for every voter and for every i, x_i is preferentially independent of $x_{i+1}, \ldots, x_p$ given $x_1, \ldots, x_{i-1}$.

Sequential voting: apply local voting rules, one variable after the other, in an order compatible with G.

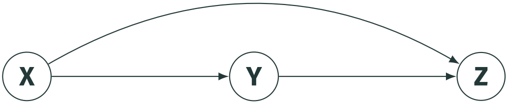
At every step:

- we elicit the voters' preferences about a single variable;
- a local rule is used to compute the value chosen for this variable;
- this value is communicated to the voters.

We don't need to know the whole preference relations of the voters but only a part of their CP-nets.

[Lang and Xia, 2009]

Voting and CP-nets: sequential voting



1. elicit voters' preferences on **X** (possible because their preferences on **X** are unconditional);
2. apply local voting rule r_X and determine the "local" winner x^*;
3. elicit voters' preferences on **Y** given **X** = x^* (possible because their preferences on **Y** depend only on **X**);
4. apply local voting rule r_Y and determine y^*;
5. elicit voters' preferences on **Z** given **X** = x^* and **Y** = y^*.
6. apply local voting rule r_Z and determine z^*.
7. winner: (x^*, y^*, z^*)

Incomplete Preferences

- New votes are coming (online vote, Doodle poll…)
- New candidates are coming (Doodle poll, recruiting committee…)
- Incomplete lists
- Truncated ballots

Incomplete Preferences

- New votes are coming (online vote, Doodle poll...)
- New candidates are coming (Doodle poll, recruiting committee...)
- Incomplete lists
- Truncated ballots

Winning candidate becomes a modal notion:

- x is a necessary winner if she wins under all possible completions of the profile.
- x is a possible winner if she wins under at least one completion of the profile.

Konczak & L (05); Walsh (07); Xia & Conitzer (08) ...

Incomplete Profiles and Manipulation...

- Borda rule
- a single voter hasn't voted yet
  - 4 voters so far:

$$
\begin{aligned}
a \succ b \succ d \succ c \succ e \\
b \succ a \succ e \succ d \succ c \\
c \succ e \succ a \succ b \succ d \\
d \succ c \succ b \succ a \succ e
\end{aligned}
$$

  - Current Borda scores

$$a \mapsto 10 \quad b \mapsto 10 \quad c \mapsto 8 \quad d \mapsto 7 \quad e \mapsto 5$$

Can the last voter find a vote so that the winner is a?

Incomplete Profiles and Manipulation...

- Borda rule
- a single voter hasn't voted yet
  - 4 voters so far:
$$\begin{array}{c} a \succ b \succ d \succ c \succ e \\ b \succ a \succ e \succ d \succ c \\ c \succ e \succ a \succ b \succ d \\ d \succ c \succ b \succ a \succ e \end{array}$$
  - Current Borda scores
$$a \mapsto 10 \quad b \mapsto 10 \quad c \mapsto 8 \quad d \mapsto 7 \quad e \mapsto 5$$

Can the last voter find a vote so that the winner is a?
$\rightarrow$ Is a a possible winner?

Automated Proofs

Recent advances in automated solving have been applied to social choice theorems.
Idea: Cast classical problems in a suitable logic and use automated theorem provers (*e.g.* SAT solvers, SMT solvers...)
No "new" theorems so far but:

- Automated verification of known proofs (*e.g.* the Gibbard-Sattherthwaith theorem [Nipkow, 2009])
- Simpler proofs or shorter counterexamples found (*e.g* the no-show paradox [Brandt et al., 2016]).

Outline

Fair Division of Indivisible Goods...

You have:

- a finite set of objects $\mathcal{O} = \{1, \ldots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ having some preferences on the set of objects they may receive

Fair Division of Indivisible Goods...

You have:

- a finite set of objects $\mathcal{O} = \{1, \ldots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \ldots, n\}$ having some preferences on the set of objects they may receive

How would you allocate the objects to the agents so as to be as fair as possible?

Fair Division of Indivisible Goods…

You have:

- a finite set of objects $\mathcal{O} = \{1, \dots, m\}$
- a finite set of agents $\mathcal{A} = \{1, \dots, n\}$ having some preferences on the set of objects they may receive

How would you allocate the objects to the agents so as to be as fair as possible?

More precisely, you want:

- an allocation $\vec{\pi} : \mathcal{A} \to 2^{\mathcal{O}}$
- such that $\pi_i \cap \pi_j = \emptyset$ if $i \neq j$ (preemption),
- $\bigcup_{i \in \mathcal{A}} \pi_i = \mathcal{O}$ (no free-disposal),
- and which takes into account the agents' preferences

Preferences for Fair Division

An intuitive way of expressing preferences…

Preferences for Fair Division

An intuitive way of expressing preferences...

- We assume that the preferences are ordinal.
- Each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Preferences for Fair Division

An intuitive way of expressing preferences...

- We assume that the preferences are ordinal.
- Each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Problem: How to compare subsets of objects ?

$\rightsquigarrow$ *e.g* $abc \overset{?}{\prec\succ} ab$; $ab \overset{?}{\prec\succ} ac$?

Preferences for Fair Division

An intuitive way of expressing preferences...

- We assume that the preferences are ordinal.
- Each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Problem: How to compare subsets of objects ?

$\rightsquigarrow$ *e.g* $abc \overset{?}{\prec\succ} ab$; $ab \overset{?}{\prec\succ} ac$?

$\rightarrow$ We need to be able to express preferences over $2^{\mathcal{O}}$.

Combinatorial Spaces…

The combinatorial trap…

Two variables…

$o_1 \succ o_2 \succ o_1 o_2 \succ \emptyset \rightarrow 3$ comparaisons (linear order).

Combinatorial Spaces…

The combinatorial trap…

Four variables…

$o_1o_2 \succ o_2o_3o_4 \succ o_1 \succ \emptyset \succ o_2 \succ o_1o_2o_3o_4 \succ o_1o_3 \succ o_2o_4 \succ o_3o_4 \succ o_1o_4 \succ o_1o_3o_4 \succ o_2o_3 \succ o_4 \succ o_3 \succ o_1o_2o_4 \succ o_1o_2o_3 \rightarrow 15$ comparisons (linear order).

Combinatorial Spaces...

The combinatorial trap...

Twenty variables...

$o_8o_5 \succ o_5o_3o_9 \succ o_8 \succ \emptyset \succ o_5 \succ o_8o_5o_3o_9 \succ o_8o_3 \succ o_5o_9 \succ o_3o_9 \succ$
$o_8o_9 \succ o_8o_3o_9 \succ o_5o_3 \succ o_9 \succ o_3 \succ o_8o_5o_9 \succ o_8o_5o_3o_1o_2o_5o_8o_9 \succ$
$o_1o_5o_6 \succ o_7 \succ o_2o_3o_4o_5o_6o_7o_8 \succ o_1o_2o_3o_4o_5 \succ o_1o_3 \succ o_2 \succ$
$o_1o_3o_7o_9 \succ o_1o_5 \succ o_1o_7o_8o_9 \succ o_2 \succ o_4 \succ o_6 \succ o_1o_7 \succ o_1o_2o_3 \succ$
$o_1o_2 \succ o_2o_5o_4 \succ o_1 \succ o_2 \succ o_1o_2o_5o_4 \succ o_1o_5 \succ o_2o_4 \succ o_5o_4 \succ$
$o_1o_4 \succ o_1o_5o_4 \succ o_2o_5 \succ o_4 \succ o_5 \succ o_1o_2o_4 \succ o_1o_2o_5 \succ o_1o_5 \succ$
$o_5o_3o_9 \succ o_1 \succ \emptyset \succ o_5 \succ o_1o_5o_3o_9 \succ o_1o_3 \succ o_5o_9 \succ o_3o_9 \succ$
$o_1o_9 \succ o_1o_3o_9 \succ o_5o_3 \succ o_9 \succ o_3 \succ o_1o_5o_9 \succ o_1o_5o_3o_9o_6o_5o_1o_9 \succ$
$o_9o_5o_6 \succ o_7 \succ o_6o_3o_4o_5o_6o_7o_1 \succ o_9o_6o_3o_4o_5 \succ o_9o_3 \succ o_6 \succ$
$o_9o_3o_7o_9 \succ o_9o_5 \succ o_9o_7o_1o_9 \succ o_6 \succ o_4 \succ o_6 \succ o_9o_7 \succ o_9o_6o_3 \succ$
$o_9o_6 \succ o_6o_5o_4 \succ o_9 \succ o_6 \succ o_9o_6o_5o_4 \succ o_9o_5 \succ o_6o_4 \succ o_5o_4 \succ$
$o_9o_4 \succ$

$\rightarrow$ 1048575 comparisons $\rightarrow$ elicitation needs more than 12 days!

The dilemma

- Expressing preferential dependencies is necessary in many cases.
- however... explicit representation and elicitation of $\succeq$ or u are unfeasible in practice.

The dilemma

- Expressing preferential dependencies is necessary in many cases.
- however...explicit representation and elicitation of $\succeq$ or u are unfeasible in practice.

$\Rightarrow$ Compact preference representation languages

The dilemma

- Expressing preferential dependencies is necessary in many cases.
- however…explicit representation and elicitation of $\succeq$ or u are unfeasible in practice.

$\Rightarrow$ Compact preference representation languages

- Cardinal utilities: Weighted propositional logic, bidding languages, GAI-nets, k-additive functions…
- Ordinal utilities: Prioritized goal bases, CI-nets…

CI-nets: the language

A language inspired from CP-nets...

CI-nets: the language

A language inspired from CP-nets...

Conditional importance statement

Conditional importance statement: $\mathcal{S}^+, \mathcal{S}^- : \mathcal{S}_1 \rhd \mathcal{S}_2$ (with $\mathcal{S}^+$, $\mathcal{S}^-$, $\mathcal{S}_1$ and $\mathcal{S}_2$ pairwise-disjoint).

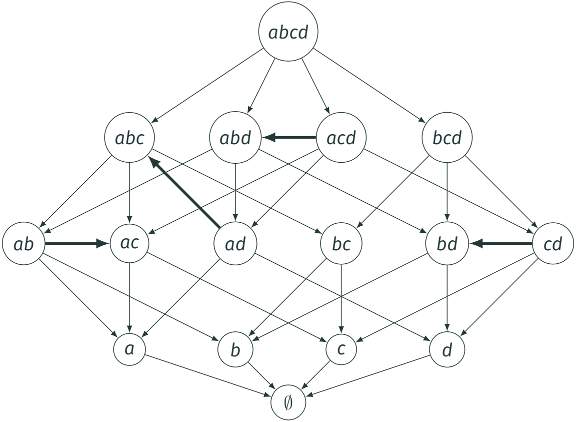
Example: $a\overline{d} : b \rhd ce$ implies for example $ab \succ ace$, $abfg \succ acefg$, $\ldots$

CI-net

A CI-net on $\mathcal{V}$ is a set $\mathcal{N}$ of conditional importance statements on $\mathcal{V}$.

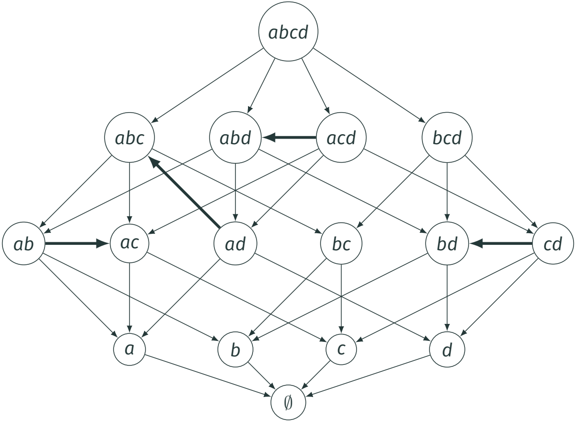
Semantics

A CI-net of 4 objects $\{a, b, c, d\}$: $\{a : d \triangleright bc, a\overline{d} : b \triangleright c, d : c \triangleright b\}$



Semantics

A CI-net of 4 objects $\{a, b, c, d\}$: $\{a : d \rhd bc, a\overline{d} : b \rhd c, d : c \rhd b\}$



Induced preference relation $\succ_{\mathcal{N}}$: the smallest monotonic preference relation compatible with all CI-statements.

CI-nets: Features

- Expressivity:
  - CI-nets can express all strict monotonic preference relations on $2^{\mathcal{V}}$.
  - Full expressivity is lost as soon as we only allow positive (resp. negative) preconditions or the cardinality of compared sets is bounded.
- Complexity:
  - [SATISFIABILITY] (consistency) is PSPACE-complete.
  - [DOMINANCE] is PSPACE-complete.

CI-nets: Features

- Expressivity:
  - CI-nets can express all strict monotonic preference relations on $2^{\mathcal{V}}$.
  - Full expressivity is lost as soon as we only allow positive (resp. negative) preconditions or the cardinality of compared sets is bounded.
- Complexity:
  - [SATISFIABILITY] (consistency) is PSPACE-complete.
  - [DOMINANCE] is PSPACE-complete.

Conclusion: a very expressive and compact language, at the price of a high computational complexity.
Is it really useful in practice?

Responsive ordinal preferences

A restricted setting...

Responsive ordinal preferences

A restricted setting...

- We assume that the preferences are ordinal.
- Restriction: each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Responsive ordinal preferences

A restricted setting...

- We assume that the preferences are ordinal.
- Restriction: each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Problem: How to compare subsets of objects ?

$\rightsquigarrow$ *e.g* $abc \overset{?}{\prec\succ} ab$; $ab \overset{?}{\prec\succ} ac$?

Responsive ordinal preferences

A restricted setting...

- We assume that the preferences are ordinal.
- Restriction: each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Problem: How to compare subsets of objects ?

$\rightsquigarrow$ *e.g* $abc \overset{?}{\prec\succ} ab$; $ab \overset{?}{\prec\succ} ac$?

1. Assume monotonicity $\rightsquigarrow$ *e.g* $abc \succ ab$.
2. Assume responsiveness: if $(X \cup Y) \cap Z = \emptyset$ then $X \succ Y$ iff $X \cup Z \succ Y \cup Z$.

$\rightsquigarrow$ *e.g* $ab \succ ac$.

Responsive ordinal preferences

A restricted setting...

- We assume that the preferences are ordinal.
- Restriction: each agent specifies a linear order $\rhd$ on O (single objects)

$$\mathcal{A} : a \rhd b \rhd c \rhd d$$

Problem: How to compare subsets of objects ?

$\rightsquigarrow$ *e.g* $abc \overset{?}{\prec\succ} ab$; $ab \overset{?}{\prec\succ} ac$?

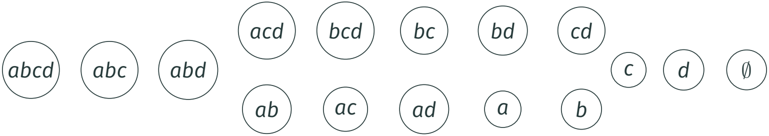
1. Assume monotonicity $\rightsquigarrow$ *e.g* $abc \succ ab$.
2. Assume responsiveness: if $(X \cup Y) \cap Z = \emptyset$ then $X \succ Y$ iff $X \cup Z \succ Y \cup Z$.

   $\rightsquigarrow$ *e.g* $ab \succ ac$.

Actually this is a restricted version of CI-nets.

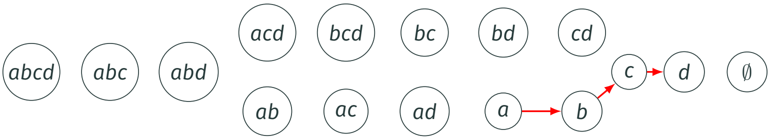
Example

- $\mathcal{A} : a \triangleright b \triangleright c \triangleright d$
- Responsiveness
- Monotonicity



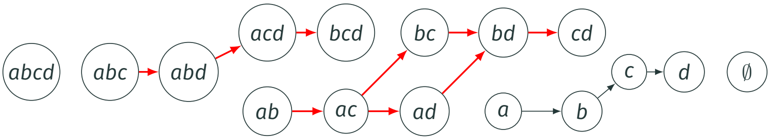
Example

- $\mathcal{A}: a \rhd b \rhd c \rhd d$
- Responsiveness
- Monotonicity



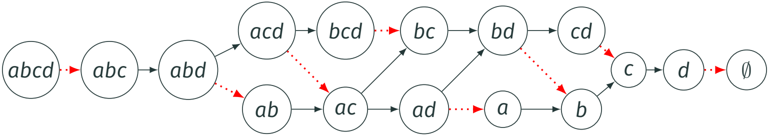
Example

- $\mathcal{A} : a \rhd b \rhd c \rhd d$
- Responsiveness
- Monotonicity



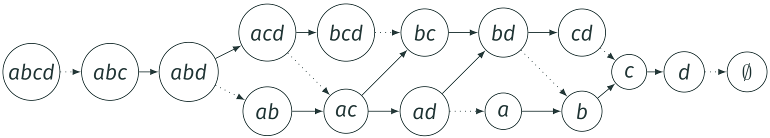
Example

- $\mathcal{A} : a \rhd b \rhd c \rhd d$
- Responsiveness
- Monotonicity



Example

- $\mathcal{A} : a \triangleright b \triangleright c \triangleright d$
- Responsiveness
- Monotonicity



Dominance

Proposition

$X \succ_{\mathcal{A}} Y \Leftrightarrow \exists$ an injective mapping of improvements $Y \mapsto X$.

Dominance

Proposition

$X \succ_{\mathcal{A}} Y \Leftrightarrow \exists$ an injective mapping of improvements $Y \mapsto X$.

Example: $\mathcal{A} = a \rhd b \rhd c \rhd d \rhd e \rhd f$

- $\{ a , c , d \} \succ_{\mathcal{A}} \{ b , c , e \}$
- $\{ a , d , e \}$ and $\{ b , c , f \}$ are incomparable.
- $\{a, c, d\}$ and $\{b, c, e, f\}$ are incomparable.

Dominance

Proposition

$X \succ_{\mathcal{A}} Y \Leftrightarrow \exists$ an injective mapping of improvements $Y \mapsto X$.

Example: $\mathcal{A} = a \triangleright b \triangleright c \triangleright d \triangleright e \triangleright f$

- $\{a, c, d\} \succ_{\mathcal{A}} \{b, c, e\}$
- $\{a, d, e\}$ and $\{b, c, f\}$ are incomparable.
- $\{a, c, d\}$ and $\{b, c, e, f\}$ are incomparable.

Dominance

Proposition

$X \succ_{\mathcal{A}} Y \Leftrightarrow \exists$ an injective mapping of improvements $Y \mapsto X$.

Example: $\mathcal{A} = a \triangleright b \triangleright c \triangleright d \triangleright e \triangleright f$

- $\{\ a\ ,\ c\ ,\ d\ \} \succ_{\mathcal{A}} \{\ b\ ,\ \underset{?}{c}\ ,\ e\ \}$
- $\{\ a\ ,\ d\ ,\ e\ \}$ and $\{\ b\ ,\ c\ ,\ f\ \}$ are incomparable.
- $\{a, c, d\}$ and $\{b, c, e, f\}$ are incomparable.

Dominance

Proposition

$X \succ_{\mathcal{A}} Y \Leftrightarrow \exists$ an injective mapping of improvements $Y \mapsto X$.

Example: $\mathcal{A} = a \rhd b \rhd c \rhd d \rhd e \rhd f$

- $\{a, c, d\} \succ_{\mathcal{A}} \{b, c, e\}$
- $\{a, d, e\}$ and $\{b, c, f\}$ are incomparable.
- $\{a, c, d\}$ and $\{b, c, e, f\}$ are incomparable.

[Brams et al., 2004, Brams and King, 2005]

Envy-freeness

Fairness…

Envy-freeness

Fairness...

Envy-freeness: $\langle \succ_1, \ldots, \succ_n \rangle$ total strict orders, allocation π.

$$\pi \text{ envy-free } \Leftrightarrow \forall i, j, \pi(i) \succ_i \pi(j)$$

Envy-freeness

Fairness…

Envy-freeness: $\langle \succ_1, \ldots, \succ_n \rangle$ total strict orders, allocation π.

$$\pi \text{ envy-free } \Leftrightarrow \forall i, j, \pi(i) \succ_i \pi(j)$$

When $\langle \succ_1, \ldots, \succ_n \rangle$ are partial orders?

Envy-freeness

Fairness...

Envy-freeness: $\langle \succ_1, \ldots, \succ_n \rangle$ total strict orders, allocation π.

$$\pi \text{ envy-free } \Leftrightarrow \forall i, j, \pi(i) \succ_i \pi(j)$$

When $\langle \succ_1, \ldots, \succ_n \rangle$ are partial orders?

$\rightsquigarrow$ Envy-freeness becomes a modal notion

Possible and necessary Envy-freeness

- π is Possibly Envy-Free *iff* for all i, j, we have $\pi(j) \not\succ_i \pi(i)$;
- π is Necessary Envy-Free *iff* for all i, j, we have $\pi(i) \succ_i \pi(j)$.

Pareto-efficiency

Efficiency...

Pareto-efficiency

Efficiency...

- Complete allocation.
- Pareto-efficiency

Pareto-efficiency

Efficiency...

Classical Pareto dominance

π' dominates π if for all i, $\pi'(i) \succeq_i \pi(i)$ and for some j, $\pi'(j) \succ_j \pi(j)$

Extended to possible and necessary Pareto dominance.

- π is *possibly Pareto-efficient* (PPE) if there exists no allocation π' such that π' necessarily dominates π.
- π' is *necessarily Pareto-efficient* (NPE) if there exists no allocation π' such that π' possibly dominates π.

Envy-freeness and efficiency

complete PPE NPE ← $\boxed{\text{Efficiency}}$

Envy-freeness and efficiency

complete PPE NPE ← Efficiency

PEF

NEF

↑

Fairness

Envy-freeness and efficiency

	complete	PPE	NPE
PEF	X	X	X
NEF	X	X	X

Efficiency

Fairness

Envy-freeness and efficiency

	complete	PPE	NPE
PEF	X	X	X
NEF	X	X	X

Efficiency

Fairness

Envy-freeness and efficiency cannot always be satisfied simultaneously

Envy-freeness and efficiency

	complete	PPE	NPE
PEF	X	X	X
NEF	X	X	X

Efficiency

Fairness

Envy-freeness and efficiency cannot always be satisfied simultaneously

Questions:

- under which conditions is it guaranteed that there exists a allocation that satisfies Fairness and Efficiency ?
- how hard it is to determine whether such an allocation exists?

Results

	complete	PPE	NPE
PEF	P (algorithm)	P (algorithm)	?
NEF	NP-complete	NP-complete (P for 2 agents)	NP-hard (Σ_2^p-completeness conjectured)

Results

	complete	PPE	NPE
PEF	P (algorithm)	P (algorithm)	?
NEF	NP-complete	NP-complete (P for 2 agents)	NP-hard (Σ_2^p-completeness conjectured)

- Results refined and extended by [Aziz et al., 2015], to the case of preferences with indifferences
- Notion of stochastic dominance

Distributed allocation

When many agents are involved, a centralized allocation may not be the most adapted solution (elicitation, computation time…).
Idea of distributed allocation:

- Start from an initial allocation
- Let the agents negotiate by swapping (bundles of) resources. Different kinds of deals:
  - with / without money
  - bounded in the number of resources involved
  - rational
  - …

Convergence properties

- Good news: for any separable collective criterion (utilitarian SW, leximin-egalitarian SW…), any sequence of locally improving deals eventually results in a socially optimal allocation

Convergence properties

- Good news: for any separable collective criterion (utilitarian SW, leximin-egalitarian SW...), any sequence of locally improving deals eventually results in a socially optimal allocation
- Bad news:
  - Any kind of restriction on the types of deals ruins this convergence property
  - The sequence of deals can be exponentially long

[Sandholm, 1998, Endriss et al., 2006, Chevaleyre et al., 2010]

Sequential allocation

Between fully centralized allocation and fully distributed allocation, a very simple procedure…

Sequential allocation

Between fully centralized allocation and fully distributed allocation, a very simple procedure...

Ask the agents to pick in turn their most preferred object among the remaining ones, according to some predefined sequence.

Example

3 agents A, B, C, 6 objects, sequence $ABCCBA \rightarrow A$ chooses first (and takes her preferred object), then B, then C, then C again...

Problems in Sequential Allocation

- Best sequence: We "feel" that *ABCCBA* is fairer than *AABBCC*...

  $\rightarrow$ *What is the fairest sequence ?*

Problems in Sequential Allocation

- Best sequence: We “feel” that *ABCCBA* is fairer than *AABBCC*…

  $\rightarrow$ *What is the fairest sequence ?*

Under some independence assumptions, classical utilitarianism, alternating sequences are optimal for two agents [Kalinowski et al., 2013a].

Problems in Sequential Allocation

- Best sequence: We “feel” that *ABCCBA* is fairer than *AABBCC*…

$\rightarrow$ *What is the fairest sequence ?*

Under some independence assumptions, classical utilitarianism, alternating sequences are optimal for two agents [Kalinowski et al., 2013a].
Egalitarianism:

p	$n = 2$	$n = 3$
4	ABBA	ABCC
5		
6		
8		
10		

Problems in Sequential Allocation

- Best sequence: We “feel” that *ABCCBA* is fairer than *AABBCC*…

  $\rightarrow$ *What is the fairest sequence ?*

Under some independence assumptions, classical utilitarianism, alternating sequences are optimal for two agents [Kalinowski et al., 2013a].
Egalitarianism:

p	$n = 2$	$n = 3$
4	ABBA	ABCC
5	AABBB	ABCCB
6		
8		
10		

Problems in Sequential Allocation

- Best sequence: We “feel” that *ABCCBA* is fairer than *AABBCC*…

$\rightarrow$ *What is the fairest sequence ?*

Under some independence assumptions, classical utilitarianism, alternating sequences are optimal for two agents [Kalinowski et al., 2013a].
Egalitarianism:

p	$n = 2$	$n = 3$
4	ABBA	ABCC
5	AABBB	ABCCB
6	ABABBA	ABCCBA
8	ABBABAAB	AACCBBCB
10	ABBAABABBA	ABCABBCACC

Strategical Issues

- Manipulation: I know all the other agents' preferences. At my turn, can I choose not to pick my preferred item to get a better share?

Strategical Issues

- Manipulation: I know all the other agents' preferences. At my turn, can I choose not to pick my preferred item to get a better share?
- First result: if I want S, I can find a manipulation to get it if it is possible.
  - Idea: greedy algorithm (pick the items in the same order of the others preferences)
  - Observation: reminds the algorithm for Borda manipulation?

Strategical Issues

- Manipulation: I know all the other agents' preferences. At my turn, can I choose not to pick my preferred item to get a better share?
- First result: if I want S, I can find a manipulation to get it if it is possible.
  - Idea: greedy algorithm (pick the items in the same order of the others preferences)
  - Observation: reminds the algorithm for Borda manipulation?
- Optimal manipulation: P for two agents [Bouveret and Lang, 2014], NP-complete for more [Aziz et al., 2016].

Strategical Issues

- Manipulation: I know all the other agents' preferences. At my turn, can I choose not to pick my preferred item to get a better share?
- First result: if I want S, I can find a manipulation to get it if it is possible.
  - Idea: greedy algorithm (pick the items in the same order of the others preferences)
  - Observation: reminds the algorithm for Borda manipulation?
- Optimal manipulation: P for two agents [Bouveret and Lang, 2014], NP-complete for more [Aziz et al., 2016].
- Game-theoretic issues: Subgame-Perfect Nash Equilibrium, Simple Nash Equilibrium...

[Kalinowski et al., 2013b, Kohler and Chandrasekaran, 1971]

Outline

Take-away message

- COMSOC: Social Choice meets Computer Science
- A lot of space for problems related to IA and CS in general: algorithmics, complexity, preference / uncertainty representation and reasoning, learning...
- A young ($\approx$ 15-20 years) but active field.

Future Trends?

Computational Social choice becomes more and more practical...

Future Trends?

Computational Social choice becomes more and more practical...

spliddit

WHALE³ WHich ALternative is Elected

http://www.spliddit.org/ http://whale3.noiraudes.net/

Future Trends?

Computational Social choice becomes more and more practical...

http://www.spliddit.org/ http://whale3.noiraudes.net/

Not only theroretically good solutions, but efficient solution that work in practice (running time, preference elicitation...)

Further readings

- Ulle Endriss's web page. A lot of resources:
  - `http://www.illc.uva.nl/COMSOC/`
  - `https://staff.fnwi.uva.nl/u.endriss/teaching/comsoc/`
- Some tutorials by Jérôme Lang (on which this presentation is based)
- *Handbook of Computational Social Choice* (2016). Brandt, Felix, Conitzer, Vincent, Endriss, Ulle, Lang, Jérôme et Procaccia, Ariel D., éditeurs. Cambridge University Press.
- *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice and Fair Division* (2016). Rothe, Jörg, éditeur. Springer.
- *Panorama de l'IA* (2014), volume 1, chapitre 15 (Systèmes Multiagents : Décision Collective). Marquis, Pierre, Papini, Odile et Prade, Henri, éditeurs. Cepaduès.

Aziz, H., Bouveret, S., Lang, J., and MacKenzie, S. (2016).
Manipulating picking sequences.
Working paper.

Aziz, H., Gaspers, S., Mackenzie, S., and Walsh, T. (2015).
Fair assignment of indivisible objects under ordinal preferences.
Artificial Intelligence, 227:71–92.

Bouveret, S. and Lang, J. (2014).
Manipulating picking sequences.
In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI'14)*, Prague, Czech Republic. IOS Press.

Brams, S. J., Edelman, P. H., and Fishburn, P. C. (2004).
Fair division of indivisible items.
Theory and Decision, 5(2).

Brams, S. J. and King, D. (2005).
Efficient fair division – help the worst off or avoid envy?
Rationality and Society, 17(4).

Brandt, F., Geist, C., and Peters, D. (2016).
Optimal bounds for the no-show paradox via sat solving.
In *Proceedings of AAMAS'2016*.

Chevaleyre, Y., Endriss, U., and Maudet, N. (2010).
Simple negotiation schemes for agents with simple preferences: Sufficiency, necessity and maximality.
Journal of Autonomous Agents and Multiagent Systems, 20(2):234–259.

Conitzer, V., Lang, J., and Xia, L. (2011).
Hypercubewise preference aggregation in multi-issue domains.
In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*, Barcelona, Spain.

Endriss, U., Maudet, N., Sadri, F., and Toni, F. (2006).
Negotiating socially optimal allocations of resources.
Journal of Artificial Intelligence Research, 25:315–348.

Kalinowski, T., Narodytska, N., and Walsh, T. (2013a).

A social welfare optimal sequential allocation procedure.
In *Proceedings of IJCAI 2013*.

Kalinowski, T., Narodytska, N., Walsh, T., and Xia, L. (2013b).
Strategic behavior when allocating indivisible goods sequentially.
In *Proceedings of AAAI'13*.

Kohler, D. A. and Chandrasekaran, R. (1971).
A class of sequential games.
Operations Research, 19(2):270–277.

Lang, J. and Xia, L. (2009).
Sequential composition of voting rules in multi-issue domains.
Mathematical Social Sciences, 57(3):304–324.

Li, M., Vo, Q. B., and Kowalczyk, R. (2011).
Majority-rule-based preference aggregation on multi-attribute domains with cp-nets.
In *Proceedings of AAMAS'11*.

Nipkow, T. (2009).

Social choice theory in hol.
Journal of Automated Reasoning, 43(3):289–304.

Sandholm, T. W. (1998).
Contract types for satisficing task allocation: I. theoretical results.
In Sen, S., editor, *Proceedings of the AAAI Spring Symposium: Satisficing Models*, pages 68–75, Menlo Park, California. AAAI Press.

Xia, L., Conitzer, V., and Lang, J. (2008).
Voting on multiattribute domains with cyclic preferential dependencies.
In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI-08)*, pages 202–207.