Computational Issues in Group Decision Making

Jérôme Lang
IRIT, CNRS & Université Paul Sabatier
Toulouse, France

SOFSEM 07, Harrachov, January 21, 2006

1. **Introduction to social choice**
2. Computationally hard voting rules
3. Voting on combinatorial domains
4. Computational aspects of strategyproofness
5. Communication requirements
6. Fair division
7. Conclusion

Social choice theory

Designing and analysing methods for collective decision making

1. a *set of agents* $\mathcal{A} = \{1, ..., n\}$;
2. a *set of alternatives* $\mathcal{X}$;
3. each agent *i* has some *preferences* on the alternatives

$\Rightarrow$ *choosing a socially preferred alternative*

Two important subdomains of social choice:

- *Vote*: agents (*voters*) express their preferences on a set of alternatives (*candidates*) and must come up to choose a candidate (or a nonempty subset of candidates).
- *Resource allocation* (fair division, auctions...): agents express their preferences over combinations of resources they may receive and an allocation must be found.

Social choice theory

1. a *set of agents* $\mathcal{A} = \{1, \ldots, n\}$;
2. a *set of alternatives* $\mathcal{X}$;
3. **each agent** i **has some preferences on the alternatives**

- *cardinal preferences*:
  - *numerical preferences* $u_i : \mathcal{X} \rightarrow \mathbb{R}$ *utility function*
  - *qualitative preferences* $u_i : \mathcal{X} \rightarrow V$ qualitative ordered scale
- *ordinal preferences*: $\succeq_i$ preference relation (transitive + reflexive) on $\mathcal{X}$

Social choice theory

- designing and evaluating formal methods of collective decision making

  Typical results: *impossibility/possibility theorems*

  There exists / there does not exist a social choice procedure meeting requirements (R1),...,(Rp)

  Example: Arrow's theorem

  If the number of alternatives is at least 3, any aggregation function defined on all profiles and satisfying unanimity and independence from irrelevant alternatives is dictatorial.

- *computational issues are neglected*

  Knowing that a given procedure *can* be computed is generally enough.

AI and social choice theory: two research areas

From social choice theory to AI

importing concepts and procedures from social choice for solving problems arising in AI applications

- societies of artificial agents (voting, negotiating / bargaining, ...)
- aggregation procedures for web site ranking and information retrieval
- vote procedures for clustering and pattern recognition

From AI to social choice theory

using AI notions and algorithms for solving complex group decision making problems.

$$\Downarrow$$

computational social choice

(the subject of this talk)

1. Introduction to social choice
2. **Computationally hard voting rules**
3. Voting on combinatorial domains
4. Computational aspects of strategyproofness
5. Communication requirements
6. Fair division
7. Conclusion

Voting rules and correspondences

1. a finite *set of voters* $\mathcal{A} = \{1, ..., n\}$;

2. a [finite] *set of candidates (alternatives)* $\mathcal{X}$;

3. a *profile* = a preference relation (= linear order) on $\mathcal{X}$ for each agent

4. $\mathcal{P}^n$ set of all profiles.

Voting rule $F : \mathcal{P}^n \to \mathcal{X}$

$F(P_1, \ldots, P_n)$ = socially preferred (elected) candidate

Voting correspondence $C : \mathcal{P}^n \to 2^{\mathcal{X}} \setminus \{\emptyset\}$

$C(P_1, \ldots, P_n)$ = set of socially preferred candidates.

Rules are obtained from correspondences by tie-breaking.

A family of voting rules: *positional scoring rules*

- N voters, p candidates
- fixed list of p integers $s_1 \geq \ldots \geq s_p$
- voter i ranks candidate x in position $j \Rightarrow score_i(x) = s_j$
- choose the candidate maximizing $s(x) = \sum_{i=1}^{n} score_i(x)$

Examples:

- $s_1 = 1, s_2 = \ldots = s_p = 0 \Rightarrow$ *plurality* rule;
- $s_1 = s_2 = \ldots = s_{p-1} = 1, s_p = 0 \Rightarrow$ *veto* rule;
- $s_1 = p-1, s_2 = p-2, \ldots s_p = 0 \Rightarrow$ *Borda* rule;

Another family of voting rules: *Condorcet-consistent rules*

Let $N(x,y) = \#\{i, x \succ_i y\}$ be the number of voters who prefer x to y.

Condorcet winner:
a candidate x such that $\forall y \neq x$, $N(x,y) > \frac{n}{2}$.

- the existence of a Condorcet winner is not guaranteed;
- when a Condorcet winner exists, it is unique

A *Condorcet-consistent rule* elects the Condorcet winner when there is one.

Another family of voting rules: *Condorcet-consistent rules*

Examples:

- *Simpson rule* (or *maximin*):
  $N(x,y)$ number of voters who prefer x to y.
  Simpson score: $S(x) = \min_{y \neq x} N(x,y)$
  Simpson winners = candidates maximizing S.

- *Copeland rule*:
  $x >_{maj} y$: a strict majority of voters prefers x to y.
  $C(x) = \#\{y | x >_{maj} y\} - \#\{y | y >_{maj} x\}$
  Copeland winners = candidates maximizing C.

Computing voting rules

Most voting rules are computed in polynomial time

Examples:

- positional scoring rules: $O(np)$
- Copeland, Simpson: $O(np^2)$

Computing voting rules

But some voting rules are NP-hard.

Dodgson for any $x \in \mathcal{X}$, $D(x)$ = smallest number of elementary changes needed to make x a Condorcet winner.

elementary change = exchange of adjacent candidates in a voter's ranking

Deciding whether x is a Dodgson winner requires a logarithmic number of calls to NP oracles: $\Delta_2^{\mathsf{P}}(O(\log n))$-complete [Hemaspaandra, Hemaspaandra & Rothe, 97]

Practical computation of Dodgson winners (and approximation schemes): (McCabe-Dansted, Pricthard and Slinko, 06), (Homan and Hemaspaandra, 06).

Computing voting rules

Young for any $x \in \mathcal{X}$, $Y(x)$ = smallest number of elementary changes needed to make x a Condorcet winner.

elementary change = removal of a voter

Deciding whether x is a Young winner is $\Delta_2^{\mathrm{P}}(O(\log n))$-complete as well [Rothe, Spakowski & Vogel, 03]

Computing voting rules

Kemeny

$d_K(P,P')$ = number of $(x,y) \in \mathcal{X}^2$ on which P and P' disagree;

$d_K(P,\langle P_1,\ldots,P_n\rangle) = \sum_{i=1,\ldots,n} d_K(P,P_i)$

P^* Kemeny consensus $\Rightarrow d_K(P^*,\langle P_1,\ldots,P_n\rangle)$ minimum

Kemeny winner = candidate ranked first in a Kemeny consensus

Deciding whether x is a Kemeny winner is $\Delta_2^{\mathsf{P}}(O(\log n))$-complete
[Hemaspaandra, Spakowski & Vogel, 03]

Practical computation of Kemeny winners (Davenport and Kalagnanam, 04);
(Conitzer, Davenport and Kalagnanam, 06), (Ailon, Charikar and Newman, 05).

Computing voting rules

Slater

$P = (P_1, \ldots, P_n)$ profile

M_P majority graph induced by P: contains the edge $x \rightarrow y$ iff a strict majority of voters prefers x to y.

Slater ranking = linear order on $\mathcal{X}$ minimising the distance to M_P.

Slater's rule is NP-hard, even under the restriction that pairwise ties cannot occur (Ailon, , Charikar and Newman, 05), (Alon, 06), (Conitzer, 06).

Computation of Slater rankings: (Charon and Hudry 00, 06; Conitzer 06).

1. Introduction to social choice
2. Computationally hard voting rules
3. **Voting on combinatorial domains**
4. Computational aspects of strategyproofness
5. Communication requirements
6. Fair division
7. Conclusion

Key question: *structure* of the set $\mathcal{X}$ of candidates?

Example 1 choosing a president:

$$\mathcal{X} = \{\texttt{John Kerry, George Bush, Ralph Nader}\}$$

Example 2 choosing a common menu:

$$\begin{aligned} \mathcal{X} = & \quad \{\texttt{asparagus risotto, foie gras}\} \\ & \times \{\texttt{roasted chicken, vegetable curry}\} \\ & \times \{\texttt{white wine, red wine}\} \end{aligned}$$

Example 3 recruiting committee (3 positions, 6 candidates):

$$\mathcal{X} = \{A \mid A \subseteq \{a,b,c,d,e,f\},\ |A| \leq 3\}.$$

Key question: *structure* of the set of candidates?

In Examples 2-3: *combinatorial domain*

$\mathcal{VAR} = \{X_1, \ldots, X_n\}$ set of variables

$\mathcal{X} = D_1 \times \ldots \times D_n$ (where D_i is a finite value domain for variable X_i)

Voting on combinatorial sets of alternatives

$\mathcal{VAR} = \{X_1, \ldots, X_n\}$ set of variables

$\mathcal{X} = D_1 \times ... \times D_n$ set of alternatives (D_i value domain for variable X_i)

Naive formulation: given a profile ($\succ_1$, ..., $\succ_n$) and a voting rule F, compute $F(\succ_1, \ldots, \succ_N)$.

First problem: the explicit representation of each $\succ_i$ is exponentially large (in the number of variables)
$\Rightarrow$ need for *compact preference representation languages*.
Such languages are meant to express preference relations (or utility functions) in as little space as possible, and to elicit preference from agents as quickly as possible.

Second problem: the direct computation of a voting rule can be very hard: when the input is expressed in a compact representation language, computing voting rules is both NP-hard and coNP-hard, and often much above.

A first idea: voting separately on each variable

⇒ *multiple election paradoxes* (Brams, Kilgour & Zwicker 98)

S: build a new swimming pool; *T*; build a new tennis court.
Suppose the true preferences are

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: How can voters 1-4 report their projected preference on $\{S, \bar{S}\}$ and $\{T, \bar{T}\}$?

Voting separately on each variable

$\Rightarrow$ *multiple election paradoxes* (Brams, Kilgour & Zwicker 98)

S: build a new swimming pool; T; build a new tennis court.
Suppose the true preferences are

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 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.

Brams, Kilgour & Zwicker: *“The only way of avoiding the paradox would consist in voting for combinations [of values] (...). If there are more than eight or so combinations to rank, the voter’s task could become burdensome. How to package combinations (...) so as not to swamp the voter with inordinately many choices – some perhaps inconsistent – is a practical problem that will not be easy to solve.”*

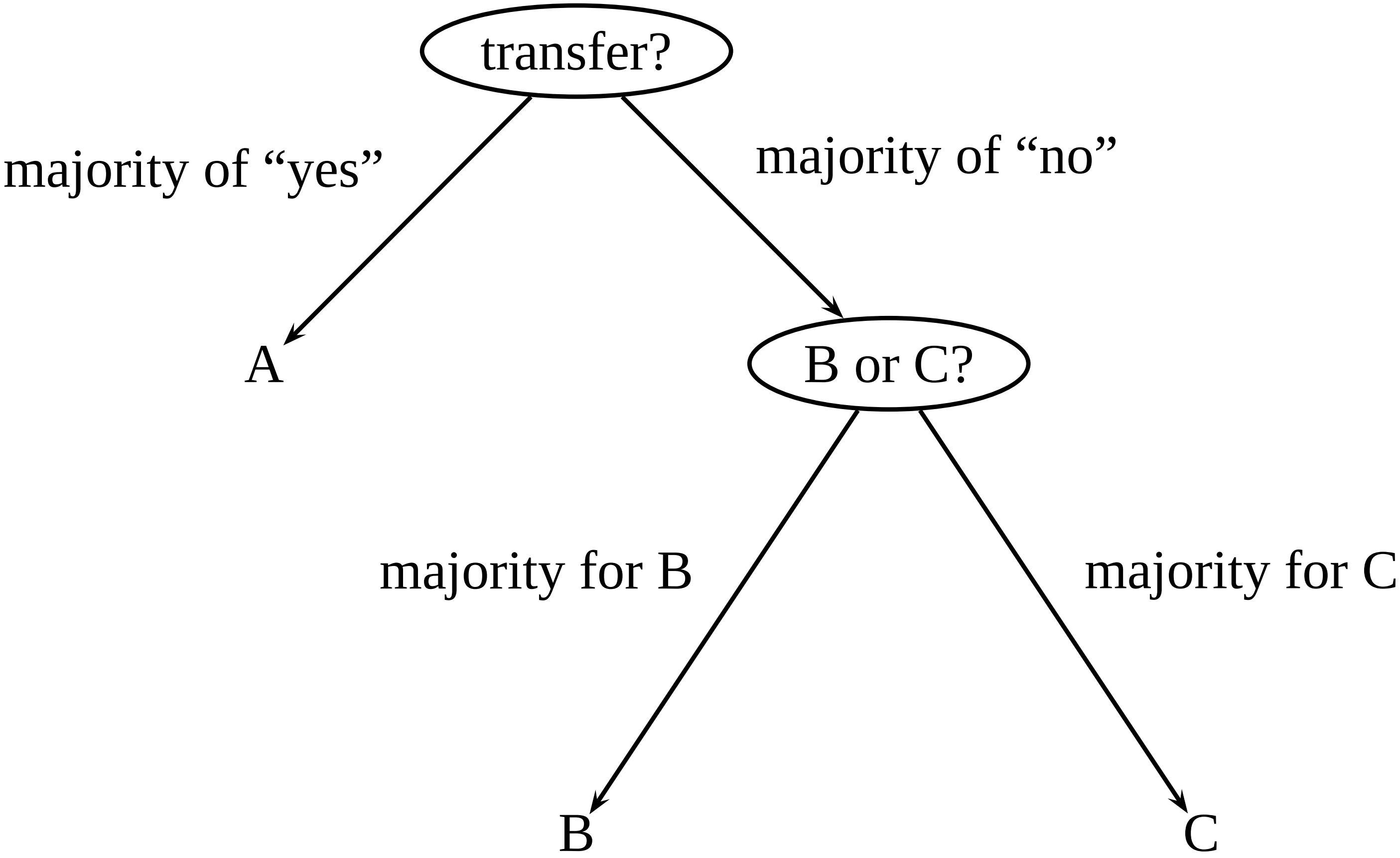
⇒ **compact preference representation languages**.

A university has a position to fill. Three candidates: A, B, C.

A already has a position in another university.

B and C do not have any position.

The law requires the recruiting committee to consider transfers first.



Computing voting rules when the input is described in a compact representation language is very hard (most rules are above NP and coNP, even when comparing two alternatives is in P).

Two ways of escaping this:

1. *approximate* the output;
2. impose *structural restrictions* on the preference profiles.

In this paper we address 2.

Assumption: voters' preferences enjoy similar preferential independence relations between variables

CP-nets [Boutilier, Brafman, Hoos and Poole, 99]

Idea: exploit *conditional preferential independence* between variables

$\{U,V,W\}$ partition of $\mathcal{VAR}$.

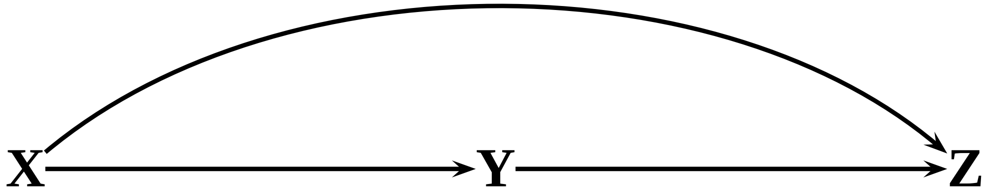
$D_U = \times_{X_i \in U} D_i$ etc.

U is preferentially independent of V (given W) iff

$$\text{for all } u,u' \in Dom(U),\ v,v' \in Dom(V),\ w \in Dom(W),$$
$$(u,v,w) \succeq (u',v,w) \text{ if and only if } (u,v',w) \succeq (u',v',w)$$

given a fixed value w of W, the preference over the possibles values of U is independent from the value of V

CP-nets



$$x \succ \bar{x}$$

$$x : y \succ \bar{y}$$
$$\bar{x} : \bar{y} \succ y$$

$$x \vee y : z \succ \bar{z}$$
$$\neg(x \vee y) : \bar{z} \succ z$$

$x : y \succ \bar{y}$

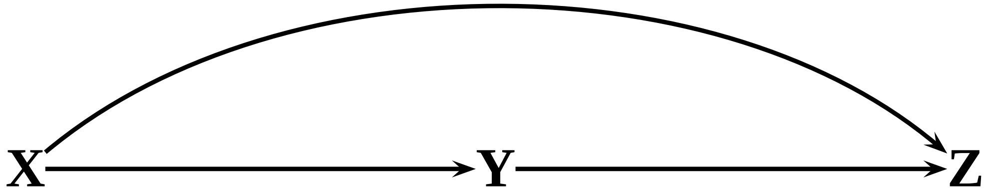
if $X = x$

then $Y = y$ preferred to $Y = \bar{y}$

everything else (z) being equal (*ceteris paribus*)

$$xyz \succ x\bar{y}z; \quad xy\bar{z} \succ x\overline{yz};$$
$$\overline{xy}z \succ \bar{x}yz; \quad \overline{xyz} \succ \bar{x}y\bar{z}$$

CP-nets



$x \succ \bar{x}$

$x : y \succ \bar{y}$
$\bar{x} : \bar{y} \succ y$

$x \vee y : z \succ \bar{z}$
$\neg(x \vee y) : \bar{z} \succ z$

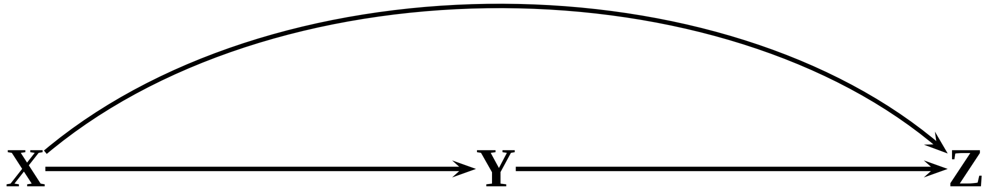
$\succ^X$: $xyz \succ \bar{x}yz$, $xy\bar{z} \succ \bar{x}y\bar{z}$, $x\bar{y}z \succ \overline{xy}z$, $x\overline{yz} \succ \overline{xyz}$

$\succ^Y$: $xyz \succ x\bar{y}z$, $xy\bar{z} \succ x\overline{yz}$, $\overline{xy}z \succ \bar{x}yz$, $\overline{xyz} \succ \bar{x}y\bar{z}$

$\succ^Z$: $xyz \succ xy\bar{z}$, $x\bar{y}z \succ x\overline{yz}$, $\bar{x}yz \succ \overline{xy}z$, $\overline{xyz} \succ \overline{xy}z$

$\succ_C$ = transitive closure of $\succ^X \cup \succ^Y \cup \succ^Z$

CP-nets



$$x \succ \bar{x}$$

$$x : y \succ \bar{y}$$
$$\bar{x} : \bar{y} \succ y$$

$$x \vee y : z \succ \bar{z}$$
$$\neg(x \vee y) : \bar{z} \succ z$$

$$\succ: \quad xyz \nearrow x\bar{y}z \searrow,\ xyz \searrow xy\bar{z} \nearrow \quad x\bar{y}\bar{z} \longrightarrow \bar{x}\bar{y}\bar{z} \longrightarrow \bar{x}\bar{y}z \longrightarrow \bar{x}yz \longrightarrow \bar{x}y\bar{z}$$

$\mathcal{G}$ *acyclic graph* on $\mathcal{VAR}$

A preference relation R is compatible with $\mathcal{G}$ iff for each $X \in \mathcal{VAR}$, X is preferentially independent of $Par(X)$ given $\overline{\{X\} \cup Par(X)}$.

$$\mathcal{G}: \quad \begin{array}{ccccc} & & & & \mathbf{Z} \\ & & & & \downarrow \\ \mathbf{X} & \rightarrow & \mathbf{Y} & \rightarrow & \mathbf{T} \end{array}$$

X is independent of $\{Y, Z, T\}$

Z is independent of $\{X, Y, T\}$

Y is independent of $\{Z, T\}$ given X

T is independent of $\{X\}$ given $\{Y, Z\}$

Observation: R compatible with $\mathcal{G}$ if and only if R extends some R' expressible by a CP-net with associated graph $\mathcal{G}$.

$\mathcal{G}$ *acyclic graph* on $\mathcal{VAR}$

$\langle R_1, \ldots, R_N \rangle$ is compatible with $\mathcal{G}$ iff each R_i is compatible with $\mathcal{G}$.

Observation: the set of $\mathcal{G}$-admissible profiles is Arrow-consistent

Sequential decomposability: apply local voting procecedures (on single variables), one after the other, in an order compatible with $\mathcal{G}$.

Observation: we don't need to know the whole preference relations $R_1, \ldots, R_N$ but only the CP-nets

$$\mathcal{C}_1 = (\mathcal{G}, \mathcal{T}_1), \ldots, \mathcal{C}_N = (\mathcal{G}, \mathcal{T}_N)$$

underlying $R_1, \ldots, R_N$.

$\mathcal{G}$ *acyclic graph* on $\mathcal{VAR}$; $\succ$ compatible with $\mathcal{G}$

$O = X_1 > ... > X_p$ *follows* $\mathcal{G}$ iff $(X_i, X_j) \in \mathcal{G}$ implies $X_i > X_j$

projection of $\succ$ on X_i given $(x_1, \ldots, x_{i-1})$:

$x_i \succ^{X_i | X_1 = x_1, \ldots, X_{i-1} = x_{i-1}} x'_i$ iff for all $(x_{i+1}, \ldots, x_p) \in D_{i+1} \times \ldots \times D_p$,
$(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_p) \succ (x_1, \ldots, x_{i-1}, x'_i, x_{i+1}, \ldots, x_p)$

Example:

$$xyz \succ x\bar{y}z \succ xy\bar{z} \succ x\overline{yz} \succ \overline{xyz} \succ \overline{xy}z \succ \bar{x}yz \succ \bar{x}y\bar{z}$$

$$x \succ^{X} \bar{x} \qquad y \succ^{Y|X=x} \bar{y} \qquad z \succ^{Z|X=x,Y=y} \bar{z}$$

$y \succ^{Y|X=x} \bar{y}$: given $X = x$, the agent prefers y to $\bar{y}$ (whatever the fixed value of Z)

$\mathcal{G}$ *acyclic graph* on $\mathcal{VAR}$; $R = (\succ_1, ..., \succ_N)$ compatible with $\mathcal{G}$;
$\mathcal{O} = X_1 > ... > X_p$ linear order on $\mathcal{VAR}$ *following* $\mathcal{G}$.

r_i collection of voting rules (one for each X_i).

Sequential voting rule $Seq(r_1, \ldots, r_p)$:

- $x_1^* = r_1(\succ_1^{X_1}, \ldots, \succ_N^{X_1})$;
- $x_2^* = r_2(\succ_1^{X_2|X_1=x_1^*}, \ldots, \succ_N^{X_2|X_1=x_1^*})$;
- $\ldots$
- $x_p^* = r_p(\succ_1^{X_p|X_1=x_1^*,..,X_{p-1}=x_{p-1}^*}, .., \succ_N^{X_p|X_1=x_1^*,..,X_{p-1}=x_{p-1}^*})$

$Seq(r_1, \ldots, r_p)(R) = (x_1^*, \ldots, x_p^*)$

Example: $r_X = r_Y =$ majority rule

3 voters	2 voters	2 voters
$\bar{x}y \succ \overline{xy} \succ x\bar{y} \succ xy$	$xy \succ x\bar{y} \succ \overline{xy} \succ \bar{x}y$	$x\bar{y} \succ xy \succ \bar{x}y \succ \overline{xy}$

For all voters, X is preferentially independent of Y: $\mathcal{G} = \{(X,Y)\}$

$\succ^X$:

3 voters	2 voters	2 voters
$\bar{x} \succ x$	$x \succ \bar{x}$	$x \succ \bar{x}$

4 voters unconditionally prefer x over $\bar{x} \Rightarrow x^* = r_X(\succ_1, \ldots, \succ_7) = x$

Example: $r_X = r_Y$ = majority rule

3 voters	2 voters	2 voters
$\bar{x}y \succ \overline{xy} \succ x\bar{y} \succ xy$	$xy \succ x\bar{y} \succ \overline{xy} \succ \bar{x}y$	$x\bar{y} \succ xy \succ \bar{x}y \succ \overline{xy}$

$$x^* = r_X(\succ_1, \ldots, \succ_7) = x$$

$\succ^{Y|X=x}$:

3 voters	2 voters	2 voters
$\bar{y} \succ y$	$y \succ \bar{y}$	$\bar{y} \succ y$

given $X = x$, 5 voters out of 7 prefer $\bar{y}$ to $y \Rightarrow y^* = r^{Y|X=x}(\succ_1, \ldots, \succ_7) = \bar{y}$

$$Seq(r_X, r_Y)(\succ_1, \ldots, \succ_7) = (x, \bar{y})$$

A voting rule r on $\mathcal{X} = D_1 \times \ldots \times D_p$ is **decomposable**
iff there exist n voting rules $r_1, \ldots, r_p$ on $D_1, \ldots, D_p$ such that:

for any linear order $\mathcal{O} = X_1 > ... > X_p$ on $\mathcal{VAR}$
and for any preference profile $R = (R_1, ..., R_N)$ following $\mathcal{O}$,
we have $Seq(r_1, \ldots, r_p)(R) = r(R)$.

- no positional scoring rule is decomposable;
- most other well-known voting rules fail to be decomposable

Obviously:

- any dictatorial rule is decomposable
- any constant rule is decomposable

Question: *are there any "reasonable" decomposable rules?*

Conjecture: if C is a decomposable, neutral and anonymous correspondence, then $C(R) = \mathcal{X}$ for all R.

Condorcet winner

$$x \text{ such that } \forall y \neq x, \#\{i, x \succ_i y\} > \frac{N}{2}$$

Sequential Condorcet winner:

$\mathcal{G}$ *acyclic graph* on $\mathcal{VAR}$; $(\succ_1, ..., \succ_N)$ compatible with $\mathcal{G}$;
$\mathcal{O} = X_1 > ... > X_p$ *following* $\mathcal{G}$.

$(x_1^*, \ldots, x_p^*)$ *sequential Condorcet winner* for P and $\mathcal{O}$ iff

- $\forall x_1' \in D_1, \#\{i, x_1^* \succ_i^{X_1} x_1'\} > \frac{N}{2}$;
- $\ldots$
- $\forall x_p' \in D_p\ \#\{i, x_p^* \succ_i^{X_p | X_1 = x_1^*, \ldots, X_{p-1} = x_{p-1}^*} x_p'\} > \frac{N}{2}$

Sequential Condorcet winner:

2 voters	1 voter	2 voters
$x\bar{y} \succ \overline{xy} \succ xy \succ \bar{x}y$	$xy \succ x\bar{y} \succ \bar{x}y \succ \overline{xy}$	$\bar{x}y \succ \overline{xy} \succ xy \succ x\bar{y}$

X and Y are preferentially independent $\Rightarrow$ take any order

- 3 voters unconditionally prefer x to $\bar{x} \Rightarrow x$ local Condorcet winner
- 3 voters unconditionally prefer y to $\bar{y} \Rightarrow y$ local Condorcet winner

$\Rightarrow xy$ sequential Condorcet winner

- a Condorcet winner is a sequential Condorcet winner – but the converse does not hold (4 voters prefer $\overline{xy}$ to xy). Cf. *paradox of the three referenda* (Laslier, 2002)
- equivalence obtained if preferences are conditionally lexicographic.

1. Introduction to social choice
2. Computationally hard voting rules
3. Voting on combinatorial domains
4. **Computational aspects of strategyproofness**
5. Communication requirements
6. Fair division
7. Conclusion

Manipulation and strategyproofness

Manipulation: a coalition of voters expressing an insincere preference profile so as to give more chance to a preferred candidate to be elected.

Example: r = plurality rule

3 voters	2 voters	2 voters
$\bar{x}y$	xy	$x\bar{y}$
$\bar{x}\bar{y}$	$x\bar{y}$	xy
$x\bar{y}$	$\bar{x}\bar{y}$	$\bar{x}y$
xy	$\bar{x}y$	$\bar{x}\bar{y}$

3 voters	**2 voters**	2 voters
$\bar{x}y$	$x\bar{y}$	$x\bar{y}$
$\bar{x}\bar{y}$	xy	xy
$x\bar{y}$	$\bar{x}\bar{y}$	$\bar{x}y$
xy	$\bar{x}y$	$\bar{x}\bar{y}$

Outcome: $\bar{x}y$

Outcome: $x\bar{y}$

Manipulation and strategyproofness

Another example: *plurality with runoff*

8	4	5
a	*c*	*b*
b	*b*	*a*
c	*a*	*c*

1st round: *c* eliminated

2nd round: *b* elected

Manipulation and strategyproofness

Manipulation: a coalition of voters expressing an insincere preference profile so as to give more chance to a preferred candidate to be elected.

Example: *plurality with runoff*

2+6	4	5
a	*c*	*b*
b	*b*	*a*
c	*a*	*c*

2	6	4	5
c	*a*	*c*	*b*
a	*b*	*b*	*a*
b	*c*	*a*	*c*

1st round: *c* eliminated

2nd round: *b* elected

Manipulation and strategyproofness

Manipulation: a coalition of voters expressing an insincere preference profile so as to give more chance to a preferred candidate to be elected.

Example: *plurality with runoff*

2+6	4	5
a	*c*	*b*
b	*b*	*a*
c	*a*	*c*

1st round: *c* eliminated

2nd round: *b* elected

2	6	4	5
c	*a*	*c*	*b*
a	*b*	*b*	*a*
b	*c*	*a*	*c*

1st round: *b* eliminated

2nd round: *a* elected.

Manipulation and strategyproofness

Gibbard (73) and Satterthwaite (75) 's theorem: if the number of candidates is at least 3, then any nondictatorial voting procedure is manipulable for some profiles.

Barriers to manipulation:

- making manipulation *less efficient*: make as little as possible of the others' votes known to the would-be manipulating coalition
- make manipulation *hard to compute*
  [Bartholdi, Tovey & Trick, 89]; [Bartholdi & Orlin, 91];
  [Conitzer & Sandholm, 02, 03]; [Conitzer, Lang & Sandholm, 03]

Making manipulation computationally hard

$\{(1,\alpha_1),\ldots,(n,\alpha_n)\}$ set of *weighted* voters ($\alpha_i \in \mathbb{N}^*$ for all i)

CONSTRUCTIVE MANIPULATION EXISTENCE: given a voting rule r, a set of p candidates $\mathcal{X}$, a candidate $x \in \mathcal{X}$, and the preferences rankings of voters $1,\ldots,k < n$, is there a way for voters $K+1,\ldots,n$ to cast their votes such that x is elected?

- plurality: in P;
- all other scoring rules (including Borda and veto): in P for $p = 2$, NP-complete for $p \geq 3$;
- Copeland and Simpson: in P for $p \leq 3$, NP-complete for $p \geq 4$;

[Conitzer & Sandholm, 02]; [Conitzer, Lang & Sandholm, 03]

1. Introduction to social choice
2. Computationally hard voting rules
3. Voting on combinatorial domains
4. Computational aspects of strategyproofness
5. **Communication requirements**
6. Fair division
7. Conclusion

Incomplete knowledge and communication complexity

Given some *incomplete* description of the voters' preferences,

- is the outcome of the voting rule determined?
- if not, whose information about which candidates is needed?

4 voters: $c \succ d \succ a \succ b$

2 voters: $a \succ b \succ d \succ c$

2 voters: $b \succ a \succ c \succ d$

1 voter: $? \succ ? \succ ? \succ ?$

plurality winner already known (c)

Borda

partial scores (for 8 voters): a: 14 ; b: 10 ; c: 14; d: 10

$\Rightarrow$ only need to know the last voters's preference between a and c

general study in [Conitzer & Sandholm, 02]

Incomplete knowledge and communication complexity

Communication complexity [Yao 79]: measure the minimum amount of information to be communicated so that the outcome of the voting procedure is determined.

⇒ design protocols for gathering the information as economically as possible

Incomplete knowledge and communication complexity

Example: plurality with runoff, n voters, p candidates.

Optimal protocol:

step 1 voters send the name of their most preferred candidate to the central authority C

$\hookrightarrow$ $\mathbf{n} \log \mathbf{p}$ bits

step 2 C sends the names of the two finalists to the voters

$\hookrightarrow$ $\mathbf{2n} \log \mathbf{p}$ bits

step 3 voters send the name of their preferred finalist to C

$\hookrightarrow$ $\mathbf{n}$ bits

total $\mathbf{n}(\mathbf{3} \log \mathbf{p} + \mathbf{1})$ bits (in the worst case)

[Conitzer & Sandholm, 05]

1. Introduction to social choice
2. Computationally hard voting rules
3. Voting on combinatorial domains
4. Computational aspects of strategyproofness
5. Communication requirements
6. **Fair division**
7. Conclusion

Resource allocation / fair division

$\mathcal{A} = \{1, \ldots, n\}$ agents

$\mathcal{R} = \{r_1, \ldots, r_p\}$ *indivisible* resources (objects)

$\pi : \mathcal{A} \to 2^{\mathcal{R}}$ *allocation*

Possible requirements for allocations:

- $\pi(i) \cap \pi(j) = \emptyset$ for $i \neq j$: *preemptive allocations*;
- $\cup_i \pi(i) = \mathcal{R}$: *complete allocations*;
- $\pi(i) = \pi(j)$ for all i, j: *shared allocations*

Finding an allocation
= group decision making with a combinatorial set of alternatives

Resource allocation $\neq$ fair division

Combinatorial auctions

$V_i : 2^{\mathcal{R}} \to \mathbb{N}$ for each agent i

$V_i(X)$ maximal value (price) that i is ready to pay for the combination of resources X

V_i additive for all $i \Rightarrow$ elicitation and optimal allocation are easy

V_i generally *not additive*

{left shoe}	5 \$	{beer}	4 \$
{right shoe}	5 \$	{lemonade}	3 \$
{left shoe, right shoe}	40 \$	{beer, lemonade}	5 \$
complementarity (superadditivity)		supplementarity (subadditivity)	

Resource allocation $\neq$ fair division

Combinatorial auctions: given $V_i : 2^{\mathcal{R}} \rightarrow \mathbb{N}$ for each agent i,

find the allocation maximizing the seller's revenue:

$$\pi^* \; maximizing \; \sum_{i=1}^{n} V(\pi(i))$$

purely utilitarianistic criterion (no equity/fairness involved)

Computational issues:

- representation / elicitation of the value functions$\Rightarrow$ bidding languages [Sandholm 99; Nisan 00; Boutilier & Hoos 01]
- computation of the optimal allocation (NP-hard): a huge literature

Fair division: three families of criteria

Numerical criteria

Need *numerical preferences* (sums of utilities are meaningful)

- utilitarianism + monetary compensation

agents	1	2
$\{a,b,c\}$	10	10
$\{a,b\}$	8	9
$\{a,c\}$	8	6
$\{b,c\}$	5	5
$\{a\}$	5	4
$\{b\}$	5	3
$\{c\}$	2	4
$\emptyset$	0	0

Fair division: three families of criteria

Numerical criteria

Need *numerical preferences* (sums of utilities are meaningful)

- utilitarianism + monetary compensation

agents	1	2
$\{a,b,c\}$	10	10
$\{a,b\}$	$\boxed{8}$	9
$\{a,c\}$	8	6
$\{b,c\}$	5	5
$\{a\}$	5	4
$\{b\}$	5	3
$\{c\}$	2	$\boxed{4}$
$\emptyset$	0	0

optimal allocation: $\pi = \langle \{a,b\}, \{c\} \rangle$

+ monetary compensation from 1 to 2: $\frac{8-4}{2} = 2$

Fair division: three families of criteria

Qualitative criteria

Need *(at least) qualitative preferences* $u_i : 2^{\mathcal{R}} \rightarrow L$ totally ordered scale common to all agents $\Rightarrow$ interpersonal comparison of preference allowed.

Fair division: three families of criteria

Qualitative criteria

Need *(at least) qualitative preferences* $u_i : 2^{\mathcal{R}} \rightarrow L$

- **equity** (or egalitarianism): the *leximin* ordering

agents	1	2
$\{a,b,c\}$	10	10
$\{a,b\}$	8	9
$\{a,c\}$	8	6
$\{b,c\}$	5	5
$\{a\}$	5	4
$\{b\}$	5	3
$\{c\}$	2	4
$\emptyset$	0	0

Fair division: three families of criteria

Qualitative criteria

Need *(at least) qualitative preferences* $u_i : 2^{\mathcal{R}} \to L$ totally ordered scale

- **equity** (or egalitarianism): the *leximin* ordering

agents	1	2
$\{a,b,c\}$	10	10
$\{a,b\}$	8	9
$\{a,c\}$	8	$\boxed{6}$
$\{b,c\}$	5	5
$\{a\}$	5	4
$\{b\}$	$\boxed{5}$	3
$\{c\}$	2	4
$\emptyset$	0	0

optimal allocation:

$\pi = \langle \{b\}, \{a,c\} \rangle$

Fair division: three families of criteria

Ordinal criteria need *(at least) ordinal preferences*

$\geq_i$: $2^{\mathcal{R}} \to L$ complete preference relation on $2^{\mathcal{R}}$

- **Pareto efficiency**: π is *efficient* iff there is no π' such that $\pi'(i) \geq_i \pi(i)$ for all i and $\pi'(i) >_i \pi(i)$ for at least one i.
- **envy-freeness**: π is *envy-free* iff for all $i, j \neq i$, $\pi(i) \geq_i \pi(j)$

- **Pareto efficiency**: π is *efficient* iff there is no π' such that $\pi'(i) \geq_i \pi(i)$ for all i and $\pi'(i) >_i \pi(i)$ for at least one i.

- **envy-freeness**: π is *envy-free* iff for all $i, j \neq i$, $\pi(i) \geq_i \pi(j)$

agents	1	2
$\{a,b,c\}$	10	10
$\{a,b\}$	8	9
$\{a,c\}$	8	$\boxed{6}$
$\{b,c\}$	5	5
$\{a\}$	5	4
$\{b\}$	$\boxed{5}$	3
$\{c\}$	2	4
$\emptyset$	0	0

$\pi = \langle\{b\},\{a,c\}\rangle$ Pareto-efficient

but not envy-free: 1 envies 2

- **Pareto efficiency**: π is *efficient* iff there is no π' such that $\pi'(i) \geq_i \pi(i)$ for all i and $\pi'(i) >_i \pi(i)$ for at least one i.
- **envy-freeness**: π is *envy-free* iff for all $i, j \neq i$, $\pi(i) \geq_i \pi(j)$

agents	1	2
$\{a,b,c\}$	10	10
$\{a,b\}$	8	9
$\{a,c\}$	8	6
$\{b,c\}$	5	$\boxed{5}$
$\{a\}$	$\boxed{5}$	4
$\{b\}$	5	3
$\{c\}$	2	4
$\emptyset$	0	0

$\pi' = \langle\{a\},\{b,c\}\rangle$ envy-free but not Pareto-efficient

For this example there is no allocation being both efficient and envy-free

Fair division: three families of criteria

preferences	numerical	qualitative	ordinal
	$u_i : 2^{\mathcal{R}} \rightarrow \mathbb{N}$	$u_i : 2^{\mathcal{R}} \rightarrow L$ L ordered scale	$\geq_i$ on 2^R
monetary compensations	+	-	-
interpersonal comparisons	+	+	-
intrapersonal comparisons	+	+	+
	utilitarianism	equity	Pareto efficiency envy-freeness

Resource allocation / fair division

- social choice theory: *axiomatic study of criteria*
- AI & OR: computational and representation issues, mainly for combinatorial auctions

⇒ Representation and computational issues for fair division?

– approximate envy-freeness: [Lipton-Markakis-Mossel-Saberi 04]

– logical representation + complexity results for

  - ordinal fair division: [Bouveret Lang 05]
  - cardinal fair division [Bouveret Fargier Lang Lemaître 05]

– complexity issues in *distributed* allocation: [Dunne, Wooldridge Laurence 05; Chevaleyre, Endriss, Estivié Maudet 04]

Fair division under dichotomous preferences [Bouveret Lang, 05]

dichotomous preference relations R is dichotomous if and only if there is a set of "good" bundles $Good$ such that for each subsets A, B of $\mathcal{R}$, $A \succeq_R B$ if and only if $A \in Good$ or $B \notin Good$.

Example:

$$X = \{a,b,c\} \Rightarrow 2^X = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,b\}, \{a,c\}, \{b,c\}, \{a,b,c\}\}$$

$$\frac{Good \longrightarrow \{\{a,b\}, \{b,c\}\}}{\overline{Good} \longrightarrow \{\emptyset, \{a\}, \{b\}, \{c\}, \{a,c\}, \{a,b,c\}\}}$$

Fair division under dichotomous preferences

A dichotomous preference is fully defined by its set of good bundles $\Rightarrow$ *propositional logic representation*

Example:

	Paul (agent 1)	Mary (agent 2)
$Good_i$	$\{\{a,b\},\{b,c\},\{a,b,c\}\}$	$\{\{b\}\{b,c\}\}$
φ_i	$b \wedge (a \vee c)$	$b \wedge \neg a$

$\succeq_{R_i}$ monotonous $\Leftrightarrow$ $Good_i$ upward closed $\Leftrightarrow$ φ_i positive formula

Fair division under dichotomous preferences

Simple propositional representation of the problem

$$\mathcal{P} = \langle \varphi_1, \ldots, \varphi_N \rangle$$

agent i, good $x \mapsto$ propositional variable x_i (x allocated to i)

rewrite φ_i, replacing each x by $x_i \Rightarrow \varphi_i^*$.

Example (continued):

	Paul (agent 1)	Mary (agent 2)
$Good_i$	$\{\{a,b\},\{b,c\},\{a,b,c\}\}$	$\{\{b\}\{b,c\}\}$
φ_i	$b \wedge (a \vee c)$	$b \wedge \neg a$
φ_i^*	$b_1 \wedge (a_1 \vee c_1)$	$b_2 \wedge \neg a_2$

Fair division under dichotomous preferences

allocation $\approx$ truth assignment of the x_i, satisfying:

$$\Gamma_{\mathcal{P}} = \bigwedge_{x \in X} \bigwedge_{i \neq j} \neg(x_i \wedge x_j)$$

Example (continued):

$$\Gamma_{\mathcal{P}} = \neg(a_1 \wedge a_2) \wedge \neg(b_1 \wedge b_2) \wedge \neg(c_1 \wedge c_2)$$

$\pi : [1 \mapsto \{a, c\}, 2 \mapsto \{b\}] \Rightarrow F(\pi) = (a_1, \neg a_2, \neg b_1, b_2, c_1, \neg c_2)$

Fair division under dichotomous preferences

Simple characterization of envy-freeness :

$$\Lambda_{\mathcal{P}} = \bigwedge_{i=1,\dots,N} \left[\varphi_i^* \vee \left(\bigwedge_{j \neq i} \neg \varphi_{j|i}^* \right) \right]$$

where $\varphi_{j|i}^* = \varphi_i^*(x_i \leftarrow x_j)$

Proposition: π is envy-free if and only if $F(\pi) \models \Lambda_{\mathcal{P}}$.

Example (continued):

$$\Lambda_{\mathcal{P}} = \left(\overbrace{(b_1 \wedge (a_1 \vee c_1))}^{1 \text{ is satisfied with her share}} \vee \overbrace{\neg(b_2 \wedge (a_2 \wedge c_2))}^{1 \text{ wouldn't be satisfied with } 2\text{'s share}} \right) \wedge \left(\underbrace{(b_2 \wedge \neg a_2)}_{2 \text{ is satisfied with her share}} \vee \underbrace{\neg(b_1 \wedge \neg a_1)}_{2 \text{ wouldn't be satisfied with } 1\text{'s share}} \right)$$

Fair division under dichotomous preferences

Pareto-efficiency requires that allocations satisfy a *maximal* set of agents.

Proposition: π is efficient if and only if $\{\varphi_i^* | F(\pi) \models \varphi_i^*\}$ is a maximal $\Gamma_{\mathcal{P}}$-consistent subset of $\{\varphi_1^*, \ldots, \varphi_N^*\}$.

Example (continued):

	agent 1	agent 2
$Good_i$	$\{\{a,b\},\{b,c\},\{a,b,c\}\}$	$\{\{b\}\{b,c\}\}$
φ_i	$(b \wedge (a \vee c))$	$b \wedge \neg a$
φ_i^*	$(b_1 \wedge (a_1 \vee c_1))$	$b_2 \wedge \neg a_2$

$$\Gamma_{\mathcal{P}} = \neg(a_1 \wedge a_2) \wedge \neg(b_1 \wedge b_2) \wedge \neg(c_1 \wedge c_2)$$

The 2 maximal $\Gamma_{\mathcal{P}}$-consistent subsets of $\{\varphi_1^*, \varphi_2^*\}$ are $\{\varphi_1^*\}$ and $\{\varphi_2^*\}$

Fair division under dichotomous preferences

Putting things together:

There exists an efficient and envy-free allocation

if and only if

$\exists S$ maximal $\Gamma_{\mathcal{P}}$-consistent subset of $\{\varphi_1^*, \ldots, \varphi_N^*\}$
such that $\bigwedge S \wedge \Gamma_{\mathcal{P}} \wedge \Lambda_{\mathcal{P}}$ is consistent.

$\Rightarrow$ SKEPTICAL INFERENCE IN DEFAULT LOGIC! (Reiter 1980)

Fair division under dichotomous preferences

Definition: Δ a set of formulae, β and ψ formulae.
ψ is a *skeptical consequence* of $\langle \beta, \Delta \rangle$ (denoted $\langle \beta, \Delta \rangle \mid\!\sim^{\forall} \psi$)
iff $\forall S \in MaxCons(\Delta, \beta)$, $\bigwedge S \wedge \beta \models \psi$.

Proposition:

there exists an EEF allocation *iff*

$$\langle \Gamma_{\mathcal{P}}, \{\varphi_1^*, \ldots, \varphi_N^*\} \rangle \not\mid\!\sim^{\forall} \neg \Lambda_{\mathcal{P}}$$

$\Rightarrow$ using default logic algorithms for finding EEF allocations.

Proposition: deciding whether there exists an EEF allocation is Σ_2^p-complete.

Fair division: bipartite fair matching

Two types of agents: $A = \{a_1, \ldots, a_n\}$; $B = \{b_1, \ldots, b_n\}$.

Find a fair matching given preferences of A-agents over B and preferences of B-agents over A

Example: $A = \{a(lice), b(etty), c(harles)\}$; $B = \{Barcelona, London, Prague\}$

a	:	London > Prague > Barcelona	Barcelona	:	a > c > b
b	:	Barcelona > London > Prague	London	:	b > c > a
c	:	London > Barcelona > Prague	Prague	:	c > b > a

Fair division: bipartite fair matching

Example: $A = \{a(lice), b(etty), c(harles)\}$; $B = \{Barcelona, London, Prague\}$

a	:	London > Prague > Barcelona	Barcelona	:	a > c > b
b	:	Barcelona > London > Prague	London	:	b > c > a
c	:	London > Barcelona > Prague	Prague	:	c > b > a

Stable allocation: if candidate x is matched with university u then any university u' such that $u' >_x u$ is matched with a candidate x' such that $x' >_{u'} x$, and similarly for universities.

π_1: a $\mapsto$ Prague, b $\mapsto$ Barcelona, c $\mapsto$ London
π_2: a $\mapsto$ Barcelona, b $\mapsto$ London, c $\mapsto$ Prague

π_1, π_2 stable allocations

Fair division: bipartite fair matching

Example: $A = \{a(lice), b(etty), c(harles)\}$; $B = \{Barcelona, London, Prague\}$

a	:	London $>$ Prague $>$ Barcelona	Barcelona	:	a $>$ c $>$ b
b	:	Barcelona $>$ London $>$ Prague	London	:	b $>$ c $>$ a
c	:	London $>$ Barcelona $>$ Prague	Prague	:	c $>$ b $>$ a

Stable allocation: if candidate x is matched with university u then any university u' such that $u' >_x u$ is matched with a candidate x' such that $x' >_{u'} x$, and similarly for universities.

π_1: a $\mapsto$ Prague, b $\mapsto$ Barcelona, c $\mapsto$ London
π_2: a $\mapsto$ Barcelona, b $\mapsto$ London, c $\mapsto$ Prague

π_1, π_2 stable allocations
π_1 Pareto-efficient for candidates but not for universities
π_2 Pareto-efficient for universities but not for candidates

Computational social choice: other issues

- social software;
- sequential group decision making;
- fairness and uncertainty;
- automated mechanism design;
- negotiation;
- communication languages;
- ...