Introduction to [computational] social choice

Jérôme Lang
LAMSADE, CNRS & Université Paris-Dauphine

KR 2012, 10 June 2012

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

Social choice theory

Social choice: *designing and analysing methods for collective decision making*

- an important subfield of economics
- goes back to the 18th century (Condorcet, Borda)

Some examples of social choice problems:

- elections
- deciding where to have dinner altogether tonight (and what time)
- `Doodle` polls (find a date for a meeting)
- in a divorce settlement: deciding who will have the chidren's custody, who keeps the stereo and who keeps the cat.
- in a jury: agreeing on a verdict
- aggregate ranked lists of web pages given by different search engines

Social choice theory

More formally:

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

$\Rightarrow$ *choosing a socially preferred alternative*

Two important subdomains of social choice:

- *Voting*: 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).
- *Fair division*: agents express their preferences over combinations of resources they may receive and an allocation must be found.
- *Judgment aggregation*: agents express their opinion on the truth of each of a number of propositions; one must come up with a consistent collective judgment.

Social choice theory

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

preferences?

Most usual models:

- *cardinal preferences*: each agent has a *utility function* $u_i : \mathcal{X} \to \mathbb{R}$ or $u_i : \mathcal{X} \to V$ qualitative ordered scale
- *ordinal preferences*: each agent has a *preference relation* $\succeq_i$ = weak order (or sometimes, linear order) on $\mathcal{X}$
- *dichotomous preferences*: each agent has a partition $\{G_i, B_i\}$ of $\mathcal{X}$ (good / bad alternatives)

[More sophisticated models: semi-orders, intervals orders, fuzzy preferences, etc.]

A very rough story of social choice:

1. end of 18th century: Condorcet and Borda (cf. *Proceedings of KR-1789*)
2. 1951: birth of modern social choice
   - results are mainly *axiomatic* (mathematics / economics)
     – impossibility theorems: *there exists no way of aggregating preferences satisfying a small set of seemingly innocuous conditions*
     – characterization theorems: *every way of aggregating preferences satisfying set of conditions S must be of form F*
   - computational issues are neglected
3. from the early 90's (and even more since the early 2000's): computer scientists come into play

⇒ **Computational social choice**: using computational notions and techniques (mainly from Artificial Intelligence, Operations Research, Theoretical Computer Science) for solving complex collective decision making problems.

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Easy voting
4. Hard voting
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

Aggregation functions, rules, correspondences

1. a finite *set of voters/agents* $\mathcal{A} = \{1, ..., n\}$;
2. a finite *set of candidates/alternatives* $\mathcal{X}$;
3. a *profile* = a preference relation (= linear order) on $\mathcal{X}$ for each agent

$$P = (V_1, \ldots, V_n) = (\succ_1, \ldots, \succ_n)$$

   V_i (or $\succ_i$) = *vote* expressed by voter i.

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

Voting rule / Social choice rule $r : \mathcal{P}^n \to \mathcal{X}$

$r(V_1, \ldots, V_n)$ = socially preferred candidate

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

$C(V_1, \ldots, V_n)$ = set of socially preferred candidates.

Aggregation function $H : \mathcal{P}^n \to \mathcal{P}$

$H(\succ_1, \ldots, \succ_n)$ = collective preference relation over X

Rules vs. correspondences

Why do we need correspondences?

- $m = 2$ candidates a, $b \Rightarrow$ obvious choice = majority
- P consists of $n = 2k$ votes: $k : a \succ b$ and $k : b \succ a$ (perfect tie)
- with voting correspondences: this is not a problem

$$C(P) = \{a, b\}$$

- with voting rules: we need a *tie-breaking mechanism*
  - give up *neutrality*: use a predefined priority relation on candidates (e.g. preference for status quo, for the eldest candidate etc.)
  - give up *anonymity*: use a predefined strict relation on voters or sets of voters (e.g. priority given to the chair's vote)

Remark: unless for some specific cases of (m, n), *no voting rule satisfies both neutrality and anonymity*.

Rules vs. correspondences

The usual way of defining voting voting rules:

- we first define a voting correspondence F
- a voting rule is implicitly defined from F by using a tie-breaking priority
- usual assumption: break neutrality
- F + tie-breaking priority $>$ over $\mathcal{X} \mapsto F_>$ voting rule
- $F_>(P) = \max(>, F(P))$

Example:

- $P = \langle a \succ b, b \succ a \rangle$
- Maj voting correspondence: $Maj(P) = \{a, b\}$
- $Maj_{a>b}$ and $Maj_{b>a}$ voting rules
- $Maj_{a>b}(P) = a$

In the rest of the talk, we usually define correspondences, and rules are defined implicitly by the tie-breaking priority.

Majority

When there are only two candidates a and b, the only “reasonable” correspondence is **majority**:

$$Maj(V_1,\ldots,V_n) = \begin{cases} \{a\} & \text{if a strict majority of voters prefer } a \text{ to } b \\ \{b\} & \text{if a strict majority of voters prefer } b \text{ to } a \\ \{a,b\} & \text{otherwise (tie)} \end{cases}$$

Exact characterization of majority in (May, 1952).

- *Anonymity*: voters should be treated symmetrically
- *Neutrality*: candidates should be treated symmetrically
- *Positive Responsiveness*: if a (sole or tied) winner receives increased support, then she should become the sole winner.

Theorem (May, 1952) A voting correspondence for two candidates satisfies anonymity, neutrality and positive responsiveness if and only if it is Maj.

FAQ

Q: Why don't we allow voters to express indifferences?

A: most voting rules can be easily and naturally generalized to profiles consisting of weak orders (instead of linear orders). We just don't do that today because we don't have enough time.

Q: Why don't we allow voters to express incomparibilities?

A: this is less easy. Later in the talk we'll talk about voting with profiles consisting of partial orders, even if our interpretation then won't be that voters are indifferent but that we have an incomplete knowledge of their preferences.

Q: Wouldn't be simpler to ask voters to give numbers?

A: in some cases, yes; in most cases, no. Numbers raise the issue of interpersonal comparison (is my 7 really better than your 6?), and it is sometimes difficult for voters to report numbers.

Voting with more than three candidates

$X = \{a, b, c, d, e\}$

- 33 votes: $a \succ b \succ c \succ d \succ e$
- 16 votes: $b \succ d \succ c \succ e \succ a$
- 3 votes: $c \succ d \succ b \succ a \succ e$
- 8 votes: $c \succ e \succ b \succ d \succ a$
- 18 votes: $d \succ e \succ c \succ b \succ a$
- 22 votes: $e \succ c \succ b \succ d \succ a$

Who should be elected?

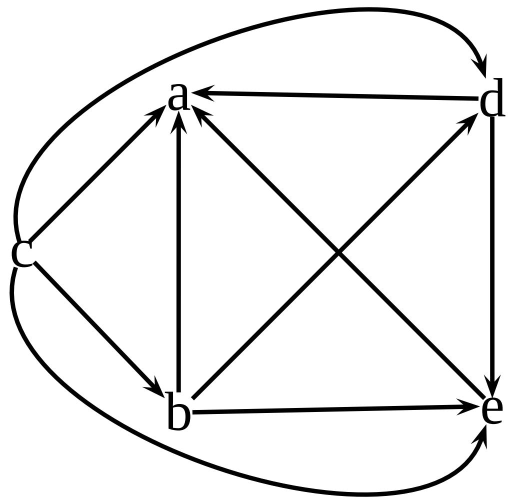
Generalizing simple majority:

pairwise majority given any two alternatives $x, y \in X$, use simple majority to determine whether the group prefers x to y or vice versa.

Does this work?

- 33 votes: $a \succ b \succ c \succ d \succ e$
- 16 votes: $b \succ d \succ c \succ e \succ a$
- 3 votes: $c \succ d \succ b \succ a \succ e$
- 8 votes: $c \succ e \succ b \succ d \succ a$
- 18 votes: $d \succ e \succ c \succ b \succ a$
- 22 votes: $e \succ c \succ b \succ d \succ a$

Majority graph associated with the profile ($x \longrightarrow y$ means that a majority of voters prefer x to y):



Collective preference relation: $c \succ b \succ d \succ e \succ a$

Winner: c

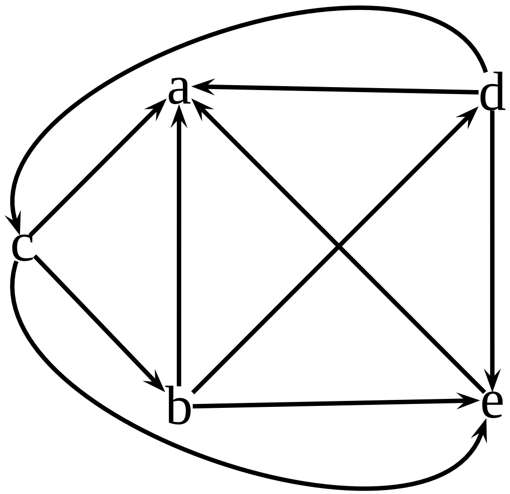
Generalizing simple majority:

pairwise majority given any two alternatives $x, y \in X$, use simple majority to determine whether the group prefers x to y or vice versa.

Does this *always* work?

- 33 votes: $a \succ b \succ \mathbf{d} \succ \mathbf{c} \succ e$
- 16 votes: $b \succ d \succ c \succ e \succ a$
- 3 votes: $c \succ d \succ b \succ a \succ e$
- 8 votes: $c \succ e \succ b \succ d \succ a$
- 18 votes: $d \succ e \succ c \succ b \succ a$
- 22 votes: $e \succ c \succ b \succ d \succ a$

Majority graph associated with the profile ($x \longrightarrow y$ means that a majority of voters prefer x to y):



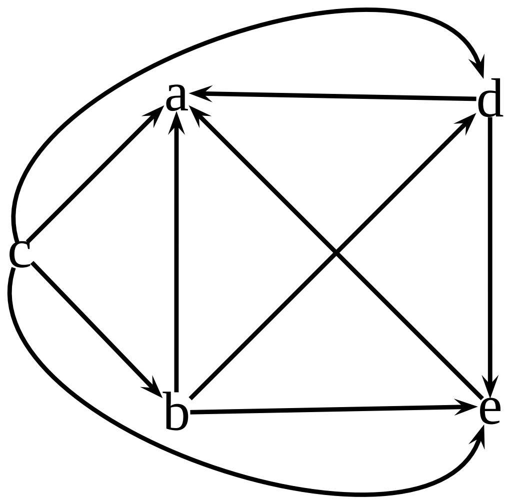
Collective preference relation: $\{b \succ c \succ d \succ b \succ \ldots\} \succ e \succ a$;

Winner: ?

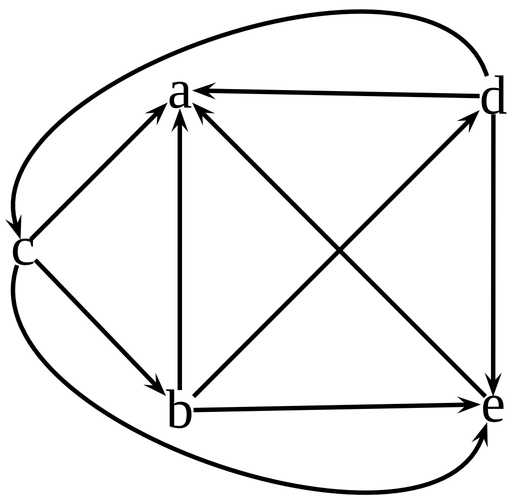
Condorcet winner

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

Condorcet winner: a candidate x such that $\forall y \neq x$, $N(x,y) > \frac{n}{2}$
(= a candidate who beats any other candidate by a majority of votes).



c Condorcet winner



no Condorcet winner

- sometimes there is no Condorcet winner
- when there is a Condorcet winner, it is unique
- a rule is *Condorcet-consistent* if it outputs the Condorcet winner whenever there is one.

Single-peakedness

Let $O : x_1 > x_2 > \ldots > x_n$ be a voter-independent axis on which alternatives are located.

- next KR chair: Mélenchon $>$ Joly $>$ Hollande $>$ Bayrou $>$ Sarkozy $>$ Le Pen
- number of breaks: $0 > 1 > 2 > 3$

Let $peak(\succ) \in X$ the preferred alternative according to $\succ$.

$\succ$ is *single-peaked* with respect to O if for any pair of alternatives x, y :

- if $x < y < peak(\succ)$ then $y \succ x$
- if $peak(\succ) < x < y$ then $x \succ y$

$\langle \succ_1, \ldots, \succ_n \rangle$ is single-peaked with respect to O if every $\succ_i$ is.

- $P = \langle 1 \succ 2 \succ 0 \succ 3, 2 \succ 3 \succ 0 \succ 1, 0 \succ 1 \succ 2 \succ 3 \rangle$
- $Q = \langle 1 \succ 2 \succ 0 \succ 3, 2 \succ 3 \succ 1 \succ 0, 0 \succ 1 \succ 2 \succ 3 \rangle$

Are P and Q single-peaked?

Single-peakedness

Theorem: if P is single-peaked then

- the pairwise majority relation associated with P is transitive
- if n is odd, then P has is a Condorcet winner. This Condorcet winner is the median of $\{peak(\succ_1), \ldots, peak(\succ_n)\}$
- (n even: there exists a *weak* Condorcet winner).

Example:

- $Q = \langle \mathbf{1} \succ 2 \succ 0 \succ 3,\ \ \mathbf{2} \succ 3 \succ 1 \succ 0,\ \ \mathbf{0} \succ 1 \succ 2 \succ 3 \rangle$
- collective preference: $1 \succ 2 \succ 0 \succ 3$
- Condorcet winner: 1

So far: everything is ok when

- we have only two alternatives
- (more generally) P is single-peaked

What can we do when $|X| \geq 3$ and P is not single-peaked?

What can we do when $|X| \geq 3$ and P is not single-peaked?

We would like to have a way of aggregating preferences (and/or to select a set of cowinners) satisfying some desirable conditions.

Condition 1: unrestricted domain (UR)

F is a function mapping every collection of linear orders $\langle \succ_1, \ldots, \succ_n \rangle$ into a collective linear order $\succ_c$

- no domain restriction such as single-peakedness
- no randomization

Condition 2: Pareto efficiency

for any $x, y \in X$, if for every i we have $x \succ_i y$ then $x \succ_c y$

- also called *unanimity*: if everyone prefers x to y, so does the group

Condition 3: independence of irrelevant alternatives (IIA)

for any $x, y \in X$, the collective preference between x and y should depend only on the individual preferences between x and y.

Formally: let $P = \langle \succ_1, \ldots, \succ_n \rangle$, $Q = \langle \succ'_1, \ldots, \succ'_n \rangle$, $\succ_c = F(P)$, $\succ'_c = F(Q)$. If for every i we have ($x \succ_i y$ if and only if $x \succ'_i y$) then ($x \succ_c y$ if and only if $x \succ'_c y$).

Condition 4: nondictatorship

It is not the case that there exists a voter i such that for every profile $P = \succ_1, \ldots, \succ_n \rangle$ we have $F(P) = \succ_i$

Arrow's theorem (1951)

If $|X| \geq 3$, there is no aggregation function satisfying conditions 1, 2, 3 and 4.

Remarks: there exists several stronger versions and variants, especially

- where the individual and collective preference relations are weak orders
- where Pareto efficiency is replaced by weaker conditions

Arrow's theorem, reformulation for voting correspondences (Taylor, 2005)

A voting correspondence C satisfies

- *weak Pareto* if for all $P = \langle \succ_1, \ldots, \succ_n \rangle$ and $x, y \in X$:
  if for every i we have $x \succ_i y$ then $y \notin C(P)$;
- *IIA* if for all $P = \langle \succ_1, \ldots, \succ_n \rangle$, $P' = \langle \succ'_1, \ldots, \succ'_n \rangle$, and $x, y \in X$:
  (a) for all i, $x \succ_i y \Leftrightarrow x \succ'_i y$, (b) $x \in C(P)$ and (c) $y \notin C(P)$ imply (d) $y \notin C(P')$.
  Informally: if x wins in P, y loses in P, and the relative order of x and y is the same in every vote of P and P', then y must also lose in P'.
- *nondictatorial* if it is not the case that there exists a voter i such that for every profile P, $C(P) = \{top(\succ_i)\}$.

Arrow's theorem (voting version)

If $|X| \geq 3$, no voting correspondence satisfies weak Pareto, IIA and nondictatorship.

Escaping Arrow's theorem

Relaxing nondictatorship is not considered an option.

Relaxing the unrestricted domain property

- **(1)** *domain restriction* such as single-peakedness
- **(2)** output a *collective preference relation with cycles or incomparabilities*
- **(3)** different input (*numerical or dichotomous preferences*)

Relaxing Pareto-efficiency

- Exercise: define a voting correspondence satisfying all properties except Pareto.
- not really interesting.

(4) Relaxing IIA

- lots of interesting voting correspondences satisfying all properties except IIA.
- Exercise: define a few such voting correspondences.

Escaping Arrow's theorem

Relaxing nondictatorship is not considered an option.

Relaxing the unrestricted domain property

- **(1)** *domain restriction* such as single-peakedness
- **(2)** output a *collective preference relation with cycles or incomparabilities*
- **(3)** different input (*such as numerical preferences*)

Relaxing Pareto-efficiency

- Exercise: define a voting correspondence satisfying all properties except Pareto.
- not really interesting.

(4) Relaxing IIA

- lots of interesting voting correspondences satisfying all properties except IIA.
- Exercise: define a few such voting correspondences.

Numerical preferences

- *profile*: $P = \langle u_1, \ldots, u_n \rangle$, $u_i : \mathcal{X} \to L$ *utility function*, L linearly ordered scale.
  - $L = \{1, \ldots, 10\}$ (trip advisor)
  - $L = \{$strong reject, weak reject, marginal, weak accept, strong accept$\}$
- $\star$ aggregation function on L
- winner(s): maximize $\star(u_1(x), \ldots, u_n(x))$

Three key choices for $\star$:

- $\star = +$ (utilitarianism): possible only if the scale L allows it
- $\star = \min$ — or better, leximin (egalitarianism)
- $\star = \mathtt{median}$ ($\sim$ "majority judgment", Balinski & Laraki 2010)

Arrow's theorem does not apply.

Dichotomous preferences: approval voting

- *profile* = a *subset of candidates* $A_i \subseteq \mathcal{X}$ for each voter:

$$P = \langle A_1, \ldots, A_n \rangle$$

$S_P(x)$ = number of voters i approved by i (= such that $x \in A_i$)

Winner(s): candidate(s) maximizing S_P.

- $X = \{a, b, c, d, e\}$
- $n = 5$
- $P = \langle \{a, c\}, \{b, c, d\}, \emptyset, \{a, b, c, d, e\}, \{d\} \rangle$
- $S_P(c) = S_P(d) = 3$; $S_P(a) = S_P(b) = 2$; $S_P(e) = 1$.
- cowinners: $\{c, d\}$

Arrow's theorem does not apply.

Subsets, numbers, or rankings?

- dichotomous preferences: weak expressivity (cannot express intensities of preference)
- numerical preferences: very rich *but* two main problems: interpersonal comparison of preference (does a 7 given by me mean the same thing as a 7 given by you?) + difficulty of elicitation.
- ordinal preferences: good trade-off; but Arrow's theorem.
  - most natural way to espace Arrow's theorem: relax IIA
  - define specific voting/aggregation rules and see how good they are.

1. Social choice and computational social choice
2. Preference aggregation, Arrow’s theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

Positional scoring rules

- n voters, m candidates
- fixed list of m integers $s_1 \geq \ldots \geq s_m$
- voter i ranks candidate x in position $j \Rightarrow score_i(x) = s_j$
- winner: candidate maximizing $s(x) = \sum_{i=1}^{n} score_i(x)$ (+ tie-breaking if necessary)

Examples:

plurality $s_1 = 1, s_2 = \ldots = s_m = 0$

veto $s_1 = s_2 = \ldots = s_{m-1} = 1, s_m = 0$

(more generally) ***k*-approval** $s_1 = \ldots = s_k = 1, s_{k+1} = \ldots = s_m = 0.$

- 1-approval = plurality
- $m-1$-approval = veto

Borda $s_1 = m-1, s_2 = m-2, \ldots s_m = 0$

33	$a \succ b \succ c \succ d \succ e$
16	$b \succ d \succ c \succ e \succ a$
3	$c \succ d \succ b \succ a \succ e$
8	$c \succ e \succ b \succ d \succ a$
18	$d \succ e \succ c \succ b \succ a$
22	$e \succ c \succ b \succ d \succ a$

- plurality: $a \mapsto 33$, $b \mapsto 16$, $c \mapsto 11$, $d \mapsto 18$, $e \mapsto 22$
  winner: a
- Borda: $a \mapsto (33 \times 4) + (3 \times 1) = 135$, $b \mapsto 247$, $c \mapsto 244$, $d \mapsto 192$, $e \mapsto 182$
  winner: b
- veto: $a \mapsto 36$, $b \mapsto 100$, $c \mapsto 100$, $d \mapsto 100$, $e \mapsto 64$
  cowinners: b, c, d
- 3-approval: $a \mapsto 33$, $b \mapsto 82$, $c \mapsto 100$, $d \mapsto 37$, $e \mapsto 48$
  winner: c

Voting rules: some important properties

Condorcet-consistency Condorcet winner elected whenever there is one.

Pareto-efficiency if every voter prefers x to y then y cannot be the winner.

Monotonicity if the winner for profile P is x and P' is obtained from P by raising x in a vote without changing anything else, then the winner for P' is still x.

Participation if the winner for profile P is x and $P' = P \cup \{\succ_{n+1}\}$, then the winner for P' is either x, or a candidate y such that $y \succ_{n+1} x$.

Reinforcement/consistency if P and Q are two profiles (on disjoint electorates) and x is the winner for P and the winner for Q, then it is also the winner for $P \cup Q$.

Question: which properties do positional scoring rules satisfy?

- Pareto-efficiency: yes
- monotonicity: yes
- participation: yes
- reinforcement: yes

Positional scoring rules

- positional scoring rules are Pareto-efficient.
- recall that Pareto-efficient, nondictatorial rules cannot satisfy IIA.
- therefore, positional scoring rules do not satisfy IIA.

Example for plurality:

P

voters $1,2,3,4:$	$a \succ b \succ c$
voters $5,6,7:$	$b \succ a \succ c$

Q

voters $1,2:$	$a \succ b \succ c$
voters $3,4:$	$c \succ a \succ b$
voters $5,6,7:$	$b \succ a \succ c$

- a winner in P
- every voter prefers a to b in P if and only if he prefers a to b in Q
- b winner in Q
- therefore: plurality violates IIA.

A characterization result for positional scoring rules

Continuity if electorate N_1 elects x and electorate N_2 does not, adding sufficiently many replicates of N_1 to N_2 leads to elect x

Theorem (Young, 75) A voting correspondence is a positional scoring correspondence if and only if it satisfies anonymity, neutrality, reinforcement, and continuity

Positional scoring rules and Condorcet-consistency

Theorem (Fishburn, 73) No positional scoring rule is Condorcet-consistent

6	$a \succ b \succ c$
3	$c \succ a \succ b$
4	$b \succ a \succ c$
4	$b \succ c \succ a$

Without loss of generality, let $s_1 = 0$.

- $S(a) = 6s_2 + 7s_1$
- $S(b) = 8s_2 + 6s_1$
- $S(b) - S(a) = 2s_2 - s_1 = s_2 + (s_2 - s_1) > 0$
- $S(b) > S(a)$ whatever the value of s_1 and s_2
- but a Condorcet winner!

Not a positional scoring rule, but close in spirit:

Bucklin

- $S_k(P,x)$ = number of voters who rank x in the first k positions
- $k^* = \min\{k,\ \text{there exists a x such that } S_k(P,x) > \frac{n}{2}\}$
- Bucklin winner(s) = k^*-approval winner(s)

33	$a \succ b \succ c \succ d \succ e$
16	$b \succ d \succ c \succ e \succ a$
3	$c \succ d \succ b \succ a \succ e$
8	$c \succ e \succ b \succ d \succ a$
18	$d \succ e \succ c \succ b \succ a$
22	$e \succ c \succ b \succ d \succ a$

Bucklin winner(s)?

- $k^* = 3$
- winner: c

Condorcet-consistent rules

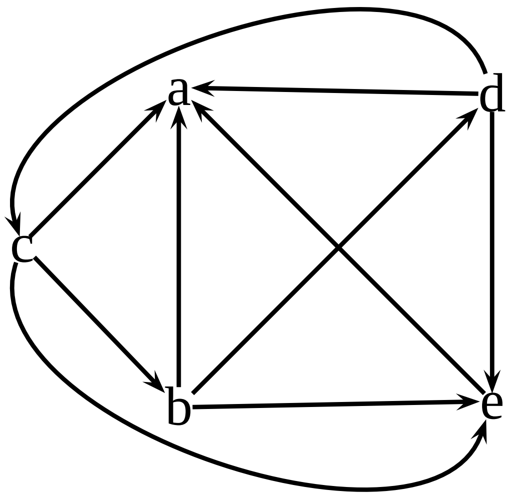
P profile $\mapsto M(P)$ directed graph associated with P A voting rule r is *based on the majority graph* if $r(P) = f(M(P))$ for some function f.

An example:

Copeland (for an odd number of voters)

$C(x)$ = number of candidates y such that $M(P)$ contains $x \longrightarrow y$.

Copeland winner = candidate maximizing C.



$C(a) = 0$

$C(b) = 3$

$C(c) = 3$

$C(d) = 3$

$C(e) = 1$

winners: b, c

FAQ

When there is an even number of voters, what do we do with ties?

Then we define Copeland$_\alpha$: if a candidate x beats p candidates and is tied with $q (= m-1-p)$ candidates then $C_\alpha(x) = p + \alpha q$. Usual choice: $\alpha = \frac{1}{2}$.

For the sake of simplicity, in the rest of the talk, when defining voting rules based on the majority graph, we assume an odd number of voters; in this case the majority graph is a complete asymmetric graph: a tournament.

Condorcet-consistent rules

P profile $\mapsto N_P(x,y) = \#\{i, x \succ_i y\}$ number of voters in P who prefer x to y.

A voting rule r is *based on the weighted majority graph* if $r(P) = g(N_P)$ for some function g.

An example:

maximin

Winner(s): maximize $S_m(x) = \min_{y \neq x} N_P(x,y)$

N_P	a	b	c	d	e
a	–	33	33	33	36
b	67	–	49	79	52
c	67	51	–	33	60
d	67	21	67	–	70
e	66	48	40	30	–

$S_m(a) = 33$
$S_m(b) = 49$
$S_m(c) = 33$
$S_m(d) = 21$
$S_m(e) = 30$
winner: b

Condorcet-consistent rules

P profile $\mapsto N_P(x,y) = \#\{i, x \succ_i y\}$ number of voters in P who prefer x to y.
A voting rule r is *based on the weighted majority graph* if $r(P) = g(N_P)$ for some function g.

Another example:

ranked pairs

1. $G :=$ graph with X as vertices and no edge.
2. order the pairs (x,y) by non-increasing order of $N_P(x,y)$, using some tie-breaking priority when necessary
3. take the first pair (x,y) in the list
4. if adding $x \longrightarrow y$ to G does not produce any cycle then add it to G
5. remove (x,y) from the list
6. if there is a unique vertex x in G with no incoming edge then return x else go back to 3.

Who is the winner for the previous profile?

Condorcet-consistent rules

Participation if the winner for profile P is x and $P' = P \cup \{\succ_{n+1}\}$ then the winner for P' is either x, or a candidate y such that $y \succ_{n+1} x$.

Reinforcement if P and Q are two profiles (on disjoint electorates) and x is the winner for P and the winner for Q, then it is also the winner for $P \cup Q$.

1. if $m \geq 3$ then no Condorcet-consistent rule satisfies reinforcement (Young, 75)
2. if $m \geq 4$ then no Condorcet-consistent rule satisfies participation (Moulin, 86)

Proof of 2. for maximin:

3	$a \succ d \succ c \succ b$
3	$a \succ d \succ b \succ c$
5	$d \succ c \succ c \succ a$
4	$b \succ c \succ a \succ d$

maximin winner: a

3	$a \succ d \succ c \succ b$
3	$a \succ d \succ b \succ c$
5	$d \succ c \succ c \succ a$
4	$b \succ c \succ a \succ d$
4	$\mathbf{c} \succ \mathbf{a} \succ \mathbf{b} \succ \mathbf{d}$

maximin winner: b

The four new voters had rather stayed home (*no-show paradox*)

Plurality with runoff

- let x, y the two candidates with the highest plurality score (use the tie-breaking rule if necessary)
- winner: majority winner between x and y

Used (today!) for elections in France.

33	$a \succ b \succ c \succ d \succ e$
16	$b \succ d \succ c \succ e \succ a$
3	$c \succ d \succ b \succ a \succ e$
8	$c \succ e \succ b \succ d \succ a$
18	$d \succ e \succ c \succ b \succ a$
22	$e \succ c \succ b \succ d \succ a$

- first step: keep a and e
- winner: e

Single transferable vote (STV)

Repeat

$d :=$ candidate ranked first by the fewest voters;

eliminate d from all ballots

$\{$votes for d transferred to the next best remaining candidate$\}$;

Until there exists a candidate x ranked first by more than 50% of the votes;

Winner: x

When there are only 3 candidates, STV coincides with plurality with runoff.

STV is used for political elections in several countries (at least Australia and Ireland)

***Single transferable vote* (STV)**

33	a ≻ b ≻ c ≻ d ≻ e
16	b ≻ d ≻ c ≻ e ≻ a
3	c ≻ d ≻ b ≻ a ≻ e
8	c ≻ e ≻ b ≻ d ≻ a
18	d ≻ e ≻ c ≻ b ≻ a
22	e ≻ c ≻ b ≻ d ≻ a

eliminate *c*

→

33	a ≻ b ≻ c ≻ d ≻ e
16	b ≻ d ≻ c ≻ e ≻ a
3	c ≻ d ≻ b ≻ a ≻ e
8	c ≻ e ≻ b ≻ d ≻ a
18	d ≻ e ≻ c ≻ b ≻ a
22	e ≻ c ≻ b ≻ d ≻ a

eliminate *b*

→

33	a ≻ b ≻ c ≻ d ≻ e
16	b ≻ d ≻ c ≻ e ≻ a
3	c ≻ d ≻ b ≻ a ≻ e
8	c ≻ e ≻ b ≻ d ≻ a
18	d ≻ e ≻ c ≻ b ≻ a
22	e ≻ c ≻ b ≻ d ≻ a

eliminate *e*

→

33	a ≻ b ≻ c ≻ d ≻ e
16	b ≻ d ≻ c ≻ e ≻ a
3	c ≻ d ≻ b ≻ a ≻ e
8	c ≻ e ≻ b ≻ d ≻ a
18	d ≻ e ≻ c ≻ b ≻ a
22	e ≻ c ≻ b ≻ d ≻ a

→ winner: *d*

***Single transferable vote* (STV)**

- (previous example) winner: d
- recall that c is the Condorcet winner
- therefore: STV is not Condorcet-consistent [same thing for majority with runoff]

Does STV and plurality with runoff satisfy other properties?

- plurality with runoff and STV do not satisfy monotonicity

6	$a \succ b \succ c$
4	$c \succ b \succ a$
5	$b \succ c \succ c$
2	$c \succ a \succ b$

b eliminated

winner: a

6	$a \succ b \succ c$
4	$c \succ b \succ a$
5	$b \succ c \succ c$
2	$\mathbf{a} \succ c \succ b$

c eliminated

winner: b

- they also fail to satisfy participation and reinforcement

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

Computing voting rules

The voting rules you have seen so far can be computed in polynomial time:

- positional scoring rules, Bucklin, plurality with runoff, approval: $O(nm)$
- Copeland, maximin, ranked pairs*, STV*: $O(nm^2)$.

But some voting rules are NP*-hard.*

Computing voting rules

The voting rules you have seen so far can be computed in polynomial time:

- positional scoring rules, Bucklin, plurality with runoff, approval: $O(nm)$
- Copeland, maximin, ranked pairs*, STV*: $O(nm^2)$.

But some voting rules are NP-*hard.*

Question What does $*$ mean in ranked pairs* and STV*?

Parallel universes

Ranked pairs and STV are polynomial-time computable in their usual version, where a tie occurring at some step is broken immediately. The *parallel universe* versions (Conitzer, Rognlie and Xia, 09) consists in exploring all possibilities and possible use tie-breaking at the very last moment.

4	$a \succ d \succ b \succ c$
3	$b \succ c \succ d \succ a$
2	$c \succ d \succ a \succ b$
2	$d \succ b \succ c \succ a$

Tie-breaking : $a > b > d > c$

- break ties immediately: c eliminated, then b, winner: d
- parallel universes:
  - branch 1 (above): winner: d
  - branch 2: d eliminated, then c, winner: a
  - cowinners $\{a, d\}$, winner: a.

- Conitzer, Rognlie and Xia (09): winner determination for parallel universe STV is NP-complete.
- Brill and Fischer (12): winner determination for parallel universe ranked pairs is NP-complete.

Hard rules: Kemeny

Looks for rankings that are as close as possible to the preference profile and chooses the top-ranked candidates in these rankings.

- *Kemeny distance*:

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

$$d_K(V, \langle V_1, \ldots, V_n \rangle) = \sum_{i=1,\ldots,n} d_K(V, V_i)$$

- *Kemeny consensus* = linear order $\succ^*$ such that $d_K(\succ^*, \langle V_1, \ldots, V_n \rangle)$ minimum
- *Kemeny winner* = candidate ranked first in a Kemeny consensus

The Kemeny rule if often use in database and in information retrieval, for aggregating "rankings" given by different databases or search engines.

Hard rules: Kemeny

4	$a \succ b \succ c$
3	$b \succ c \succ a$
2	$c \succ a \succ b$

N	a	b	c
a	–	6	4
b	**3**	–	7
c	**5**	**2**	–

Computing $d(a \succ b \succ c, \langle V_1, \ldots, V_9 \rangle)$:

- 3 voters disagree with $a \succ b$
- 5 voters disagree with $a \succ c$
- 2 voters disagree with $b \succ c$
- hence $d(a \succ b \succ c, \langle V_1, \ldots, V_9 \rangle) = 10$.

Kemeny scores:

abc	*acb*	*bac*	*bca*	*cab*	*cba*
10	15	13	12	14	17

Kemeny consensus: *abc*; Kemeny winner: *a*

Hard rules: Kemeny

- early results: Kemeny is NP-hard (Orlin, 81; Bartholdi *et al.*, 89; Hudry, 89)
- deciding whether a candidate is a Kemeny winner is $\Delta_2^{\mathsf{P}}(O(\log n))$-complete (Hemaspaandra, Spakowski & Vogel, 04): needs logarithmically many oracles.

Hard rules: Kemeny

- Kemeny rule as a maximum likelihood estimator (Young, 95):
  - there is an objective, correct ranking of candidates (the idea comes back to Condorcet)
  - there is a fixed $p \in (\frac{1}{2}, 1]$ such that for any voter i and candidates x, y, the probability that i says $x \succ y$ given that x is objectively above y is p
  - a ranking $\succ$ maximizes $p(\langle V_1, \ldots, V_n | \succ)$ iff it is a Kemeny consensus.
- can be easily applied to with incomplete rankings
- frequently used for web page ranking

Hard rules: Dodgson

For any $x \in X$, $D(x)$ = smallest number of elementary changes needed to make x a Condorcet winner.

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

Dodgson winner(s): candidate(s) minimizing $D(x)$

Hard rules: 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

Dodgson winner(s): candidate(s) minimizing $D(x)$

An example (Nurmi, 04):

10	$d \succ a \succ b \succ c$
8	$b \succ c \succ a \succ d$
7	$c \succ a \succ b \succ d$
4	$d \succ c \succ a \succ b$

Dodgson winner: d, although d is the Condorcet loser.

Who is the winner if all votes are reversed?

Hard rules: Dodgson

Another example (Brandt, 09): Dodgson does not satisfy *homogeneity*

Replace every voter by three voters:

$2: \quad d \succ c \succ a \succ b$	$6: \quad d \succ c \succ a \succ b$
$2: \quad b \succ c \succ a \succ d$	$6: \quad b \succ c \succ a \succ d$
$2: \quad c \succ a \succ b \succ d$	$6: \quad c \succ a \succ b \succ d$
$2: \quad d \succ b \succ c \succ a$	$6: \quad d \succ b \succ c \succ a$
$2: \quad a \succ b \succ c \succ d$	$6: \quad a \succ b \succ c \succ d$
$1: \quad a \succ d \succ b \succ c$	$3: \quad a \succ d \succ b \succ c$
$1: \quad d \succ a \succ b \succ c$	$3: \quad d \succ a \succ b \succ c$
Dodgson winner: a	Dodgson winner: d

Hard rules: Dodgson

- Bartholdi, Tovey & Trick, 89: deciding whether *x* is a Dodgson winner is NP-hard.
- Hemaspaandra, Hemaspaandra & Rothe, 97: deciding whether *x* is a Dodgson winner is Θ_2^{P}-complete (= requires a logarithmic number of calls to NP oracles)

Caragiannis, Kaklamanis, Karanikolas & Procaccia (10): *socially desirable approximations of Dodgson*. Example: *monotonic approximations* = voting rules:

- satisfying monotonicity
- close enough to Dodgson
- (possibly) computable in polynomial time

The approximation of a voting rule is a new voting rule that may be interesting *per se*!

Hard 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

10	$c \succ b \succ a \succ d$
8	$d \succ a \succ b \succ c$
7	$d \succ b \succ a \succ c$
4	$b \succ a \succ c \succ d$

Find the Young winner(s).

Deciding whether x is a Young winner is Θ_2^P-complete (Rothe, Spakowski & Vogel, 03)

Hard rules: Slater

$P = (V_1, \ldots, V_n)$ 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 winner: best candidate in some Slater ranking

Slater's rule is NP-hard (but maybe not in NP), 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).

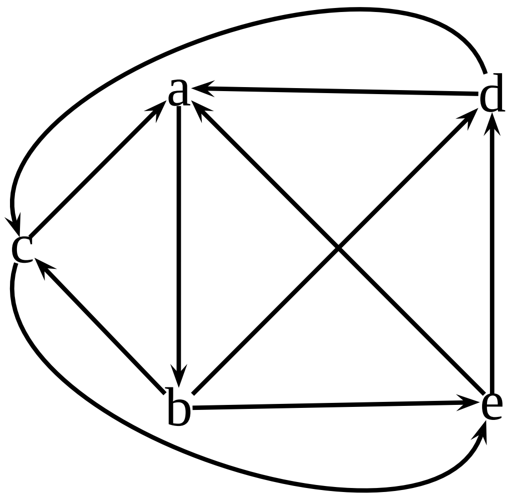
Hard rules: Banks

- M_P majority graph induced by P:
- maximal subtournament of M_G: maximal subset of $\mathcal{X}$ such that the restriction of M_G to $\mathcal{X}$ is transitive.
- x is a Banks winner if x is undominated in some maximal subtournament of M_G.
- deciding whether x is a Banks winner is NP-complete (Woeginger, 2003)
- however, it is possible to find an arbitrary Banks winner in polynomial time (Hudry, 2004)

Finding a Banks winner in polynomial time by a greedy algorithm:

> $A := \{x\}$ where x is an arbitrary candidate;
> **repeat**
>   find y such that the subgraph of M_P restricted to $A \cup \{y\}$ is cycle-free;
>   add y to A
> **until** it is no longer possible to do so;
> **return** the maximal element in A

Hard rules: Slater and Banks



Find the Slater and Banks winner(s).

Hard rules: other tournament solutions

- minimal covering set: non-trivially polynomial (Brandt & Fischer, 08);
- minimal extending set: NP-hard.
- tournament equilibrium set: NP-hard.

Hard rules

Discussion

- r is in P: easy to compute

  positional scoring rules, Bucklin, Copeland, maximin, ranked pairs*, plurality with runoff, STV*

- r is NP-complete: not easy to compute but easy to verify a solution using a succinct certificate

  Banks, STV**, ranked pairs**

- r is beyond NP: not even easy to verify.

  Kemeny, Young, Dodgson (and Slater?)

Is there a life after NP-hardness?

- *efficient computation*: design algorithms that do as well as possible, possibly using heuristics, or translations into well-known frameworks (such as integer linear programming).
- *fixed-parameter complexity*: isolate the components of the problem and find the main cause(s) of hardness
- *approximation*: design algorithms that produce a (generally suboptimal) result, with some performance guarantee.

KR crowd, please help So many rules at the second level of the polynomial hierarchy: can winner determination be encoded in ASP?

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

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

Example 1 choosing a common menu:

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

Example 2 multiple referendum: a local community has to decide on several interrelated issues (should we build a swimming pool or not? should we build a tennis court or not?)

Example 3 choosing a joint plan: the group travel problem (Klamler & Pfirschy). A set of cities; a set of agents; each of whom has preferences over edges between cities. The group will travel together and has to reach every city once.

Example 4 recruiting committee (3 positions, 6 candidates):

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

Combinatorial domains: $\mathcal{V} = \{X_1,\ldots,X_p\}$ set of *variables*, or *issues*;
$\mathcal{X} = D_1 \times \ldots \times D_p$ (where D_i is a finite value domain for variable X_i)

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	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

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	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

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!

How should such a vote be conducted?

Problem: *preferential dependencies* between variables ("I want to build the swimming pool only if the tennis court is not built") make it impossible to decompose in to a vote on every variable.

A few possible solutions:

1. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
2. ask voters to report only a small part of their preference relation and appply a voting rule that needs this information only, such as plurality.
3. ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.
4. sequential voting : decide on every variable one after the other, and broadcast the outcome for every variable before eliciting the votes on the next variable. (Example: $main - dish > wine$)
5. use a *compact preference representation language* in which the voters' preferences are represented in a concise way.

3 **ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.**

- every voter specifies one or several preferred alternatives $\vec{x}^*$;
- for all alternatives $\vec{x}, \vec{y} \in D$, $\vec{x} \succ_i \vec{y}$ if and only if $d(\vec{x}, \vec{x}^*) < d(\vec{y}, \vec{x}^*)$, where d is a predefined distance on D.

+ cheap in elicitation an computation.

– important domain restriction.

Two examples of such approaches:

- propositional merging (Konieczny & Pino-Perez 98, etc.)
- minimax approval voting

Minimax approval voting (Brams, Kilgour & Sanver, 2007)

- n voters, m candidates, $k \leq m$ positions to be filled
- each voter casts an approval ballot $V_i = (v_i^1, \ldots, v_i^m) \in \{0,1\}^m$
- for every subset Y of k candidates,
  - $d(Y, V_i)$ = Hamming distance between Y and V_i (number of disagreements)
  - $d(Y, (V_1, \ldots, V_n)) = \max_{i=1,\ldots,n} d(Y, V_i)$
  - find Y minimizing $d(Y, (V_1, \ldots, V_n))$

Example: $n = 4$, $m = 5$, $k = 2$.

	x_1	x_2	x_3	x_4	x_5
1	1	1	1	0	0
2	1	1	0	0	0
3	1	1	0	1	1
4	0	0	1	1	1

- $d(\{x_1, x_3\}, V) = \max(3, 2, 2, 3) = 3$; $\{x_1, x_3\}$ minimax-optimal committee.
- $\{x_1, x_2\}$ minisum-optimal committee; however, $d(\{x_1, x_2\}, V) = 5$.

Minimax approval voting

- finding an optimal committee is NP-hard (Frances & Litman, 97)
- (Le Grand, Markakis & Mehta, 07): approximation algorithms for minimax approval

Algorithm:

pick arbitrarily one of the ballots V_j
$k_j \leftarrow$ number of 1's in V_j
if $k_j > k$ **then** pick $k_j - k$ coordinates in V_j and set them to 0;
if $k_j < k$ **then** pick $k - k_j$ coordinates in V_j and set them to 1;
return the modified ballot V'_j

The above algorithm is a polynomial 3-approximation of minimax approval (Le Grand, Markakis & Mehta, 07)

Better approximation (ratio 2) in (Caragiannis, Kalaitzis & Markakis, 10)

A more general setting where minimax approval voting finds its place: **multiwinner elections** (Meir, Procaccia, Rosenschein & Zohar, 08; Betzler *et al.*, 11; Elkind *et al.*, 11; Lu and Boutilier, 11; etc.)

5 **use a *compact preference representation language***

Example: hypercubewise preference aggregation (Xia *et al.*, 08/11)

Example 1 (swimming pool): 5 voters, 2 binary issues S, T; each voter: a CP-net.

2 voters	2 voters	1 voter
$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$	$ST \succ \bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T}$
$\mathbf{S} \longleftrightarrow \mathbf{T}$	$\mathbf{S} \longleftrightarrow \mathbf{T}$	$\mathbf{S}$ $\quad$ $\mathbf{T}$
$\boxed{\begin{array}{l} T : \bar{S} \succ S \\ \bar{T} : S \succ \bar{S} \end{array}}$ $\boxed{\begin{array}{l} S : \bar{T} \succ T \\ \bar{S} : T \succ \bar{T} \end{array}}$	$\boxed{\begin{array}{l} T : \bar{S} \succ S \\ \bar{T} : S \succ \bar{S} \end{array}}$ $\boxed{\begin{array}{l} S : \bar{T} \succ T \\ \bar{S} : T \succ \bar{T} \end{array}}$	$\boxed{S \succ \bar{S}}$ $\boxed{T \succ \bar{T}}$

apply an aggregation function (here majority) on each entry of each table

$$\mathbf{S} \longleftrightarrow \mathbf{T}$$

$$\boxed{\begin{array}{l} T : \bar{S} \succ S \\ \bar{T} : S \succ \bar{S} \end{array}} \quad \boxed{\begin{array}{l} S : \bar{T} \succ T \\ \bar{S} : T \succ \bar{T} \end{array}}$$

$$\begin{tikzcd} ST & \color{red}{\bar{S}T} \arrow[l] \arrow[d] \\ \color{red}{S\bar{T}} \arrow[u] \arrow[r] & \bar{S}\bar{T} \end{tikzcd}$$

Example 2: 3 voters, 2 binary issues *A*, *B*

A $\longrightarrow$ **B**

$$\boxed{A \succ \bar{A}} \qquad \boxed{\begin{array}{l} A : B \succ \bar{B} \\ \bar{A} : \bar{B} \succ B \end{array}}$$

A $\longleftarrow$ **B**

$$\boxed{\begin{array}{l} B : A \succ \bar{A} \\ \bar{B} : \bar{A} \succ A \end{array}} \qquad \boxed{B \succ \bar{B}}$$

A **B**

$$\boxed{\bar{A} \succ A} \qquad \boxed{\bar{B} \succ B}$$

apply an aggregation function (here majority) on each entry of each table

A $\longleftrightarrow$ **B**

$$\boxed{\begin{array}{l} B : \bar{A} \succ A \\ \bar{B} : A \succ \bar{A} \end{array}} \qquad \boxed{\begin{array}{l} A : B \succ \bar{B} \\ \bar{A} : \bar{B} \succ B \end{array}}$$

$$\begin{tikzcd} AB \arrow[d] & \bar{A}B \arrow[l] \\ A\bar{B} \arrow[r] & \bar{A}\bar{B} \arrow[u] \end{tikzcd}$$

How should such a vote be conducted?

A few possible solutions:

1. ask voters to specify their preference relation by ranking all alternatives *explicitly*: inacceptable elicitation burden if more than 3 or 4 variables variables.
2. ask voters to report only a small part of their preference relation and appply a voting rule that needs this information only, such as plurality: catastrophical results as soon as the number the variables is not very small.
3. ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*: domain restriction + computational complexity.
4. sequential voting : decide on every variable one after the other, and broadcast the outcome for every variable before eliciting the votes on the next variable (example: $main - dish > wine$): domain restriction OR not so good results.
5. use a *compact preference representation language* in which the preferences are represented in a concise way: high elicitation + computational cost.

Conclusion: *if we want to avoid catastrophical results: impose a strong domain restriction and/or or pay a high communication and computational cost.*

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

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 with runoff*

$$\begin{array}{ll} 8 & a \succ b \succ c \\ 4 & c \succ b \succ a \\ 5 & b \succ a \succ c \end{array}$$

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: r = *plurality with runoff*

$\mathbf{2}+6$	$a \succ b \succ c$
4	$c \succ b \succ a$
5	$b \succ a \succ c$

1st round: c eliminated

2nd round: b elected

$\mathbf{2}$	$\mathbf{c \succ a \succ b}$
6	$a \succ b \succ c$
4	$c \succ b \succ a$
5	$b \succ a \succ c$

1st round: b eliminated

2nd round: a elected

Is this a specific flaw of plurality with runoff?

Unfortunately no.

Manipulation and strategyproofness

Gibbard (73) and Satterthwaite (75) 's theorem

If $|X| \geq 3$, any nondictatorial, surjective voting rule is manipulable for some profiles.

Escaping Gibbard and Satterthwaite

Relaxing nondictatorship is again not considered an option.

Assuming single-peakedness When voters have single-peaked preferences, there are strategyproof voting rules. Examples:

- $r(P)$ = median of peaks
- more generally: $r_k(P) =$ the kth leftmost peak

Randomization Technically a solution, but does not go further than that:

- select two random voters; let x and y be their top candidates; winner = $maj(x,y)$

Mechanism design (numerical preferences + money transfer)

- each voter gives the amount of money $v(x)$ she is ready to pay to see x elected;
- winner + payments determined by the Vickrey-Clarkes-Grove mechanism.

Full uncertainty Assume voters *do not know anything* about the others' preferences. Many voting rules are then strategyproof (Conitzer, Walsh and Xia, 11).

Nearly escaping Gibbard and Satterthwaite

One more solution:

Computational barrier

- make manipulation *hard to compute*.
- the harder it is to find a manipulation, the better the voting rule
- (similar approach in cryptography)

Given a voting rule r:

Input vote r, a set of m candidates $\mathcal{X}$, a candidate $x \in \mathcal{X}$, votes of voters $1, \ldots, k < n$

Question is it possible for voters $k+1, \ldots, n$ to cast their votes so that the winner is x?

First papers on the topic: Bartholdi, Tovey & Trick (89); and lots of papers since then.

Complexity of manipulation

Manipulating the Borda rule by a single voter

a	b	c	d
b	a	e	c
d	e	a	b
c	d	b	a
e	c	d	e

Current Borda scores:

a: 10

b: 10

c: 8

d: 7

e: 5

Can the last voter find a vote so that the winner is a? b? c? d? e?

Complexity of manipulation

Manipulating the Borda rule by two voters

Borda + tie-breaking priority $a > b > c > d > e$.

Current Borda scores:

a: 12

b: 10

c: 9

d: 9

e: 4

f: 1

Is there a constructive manipulation by *two* voters for e?

Complexity of manipulation

For the Borda rule:

Manipulation existence

- *for a single voter* : in P (Bartholdi, Tovey & Trick, 89).
- for a *coalition of at least two voters* : NP-complet (was an open result since long; proved independently in 2011 by (Betzler, Niedermeyer and Woeginger, IJCAI-11) and (Davies, Katsirelos, Narodytska and Walsh, AAAI-11)

Complexity of manipulation

Number of manipulators	1	at least 2
Copeland	P (1)	NP-complete (2)
STV	NP-complete (3)	NP-complete (3)
veto	P (4)	P (4)
cup	P (5)	P (5)
maximin	P (1)	NP-complete (6)
ranked pairs	NP-complete (6)	NP-complete (6)
Bucklin	P (6)	P (6)
Borda	P (1)	NP-complete (7,8)

(1) Bartholdi *et al.* (89); (2) Falisezwski *et al.* (08); (3) Bartholdi and Orlin (91); (4) Zuckerman *et al.* (08); (5) Conitzer *et al.* (07); (6) Xia *et al.* (09); (7) Betzler *et al.*, 11; (8) Davies *et al.*, 11.

Complexity of manipulation

An important concern:

- a worst-case NP-hardness results only says that *sometimes* (maybe rarely), computing a manipulation will be hard
  $\Rightarrow$ too weak
- *negative* results about the average hardness of manipulation (Conitzer and Sandholm, 06; Procaccia and Rosenschein, 07: Xia and Conitzer, 08).

Results about the *frequency* of manipulability (*e.g.*, Slinko *et al.*, 04; Xia and Conitzer, 08).

- k = size of the manipulating coalition
- if $k \ll \sqrt{n}$ then it is highly likely that there is no manipulation;
- if $k \gg \sqrt{n}$ then it is highly likely that there is a manipulation.

Other kinds of strategic behaviour: **procedural control**

Some voting procedures can be controlled by the authority conducting the election (i.e. the chair) to achieve strategic results.

Several kinds of control:

- adding / deleting / partitioning candidates
- adding / deleting / partitioning voters

For each type of control and each voting rule r, three possivilities

- r is *immune to control*: it is never possible for the chairman to change a dandidate c from a non-winner to a unique winner.
- r is *resistant to control*: r is not immune and it is computationally hard to recognize opportunities for control
- r is *vulnerable to control*: r is not immune and it is computationally easy to recognize opportunities for control

Other kinds of strategic behaviour: **procedural control**

Control by adding candidates The chairman to add "spoiler" candidates in hopes of diluting the support of those who might otherwise defeat his favourite candidate.

GIVEN: A set C of qualified candidates and a distinguished candidate $c \in C$, a set B of possible spoiler candidates, and a set V of voters with preferences over $C \cup B$.

QUESTION: Is there a choice of candidates from B whose entry into the election would assure victory for c?

- r is immune to control by adding candidates if it satisfies the following property: the winner among a set of candidates be the winner among every subset of candidates to which he belongs (Plott, 76).
- plurality voting is computationally resistant to control by adding candidates (Bartholdi, Tovey & Trick, 90).

Other kinds of strategic behaviour: **procedural control**

Agenda control in sequential voting In sequential voting on a combinatorial domain, the chair can sometimes influence the outcome by fixing the agenda.

40 %	$S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$
40 %	$\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$
20 %	$ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

Suppose that the voters' behaviour is majoritarily optimistic.

- vote on **S** and then on **T** $\Rightarrow S\bar{T}$
- vote sur **T** and then on **S** $\Rightarrow \bar{S}T$

The chair's strategy: **S** first if $S\bar{T} \succ_{arb} \bar{S}T$, **T** first $\bar{S}T \succ_{arb} S\bar{T}$.

(Under reasonable assumptions on the encoding of the input data) determine if there exists an order leading to a given outcome is NP-complete (Conitzer, Lang & Xia, 09).

Other kinds of strategic behaviour: **bribery**

GIVEN: a set C of candidates, a set $V = \{1, \ldots, n\}$ of voters specified with their preferences, n integers $p_1, \ldots, p_n$ (p_i = price for making voter i change his vote), a distinguished candidate c, and a nonnegative integer K.

QUESTION: Is it possible to make c a winner by changing the preference lists of voters while spending at most K?

(Rothe, Hemaspaandra and Hemaspaandra, 07):

- for plurality: BRIBERY is in P (and NP-complete for weighted voters)
- for approval voting: BRIBERY is in NP-complete, even for unit prices ($p_i = 1$ for each i)

variations on bribery: nonuniform bribery (Faliszewski, 08), swap bribery (Elkind, Faliszewski an Slinko, 09); *etc.*

1. Social choice and computational social choice
2. Preference aggregation, Arrow’s theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

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 ?

Borda ?

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 ?

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

Incomplete knowledge and communication complexity

More general problems to be considered:

- Which elements of information should we ask the voters and when on order to determine the winner of the election while minimizing communication?
- When the votes are only partially known: is the winner already determined? Which candidates can still win?
- When only a part of the electorate have expressed their votes, how can we synthesize the information expressed by this subelectorate as succinctly as possible?
- When the voters have expressed their votes on a set of candidates and then some new candidates come in, who among the initial candidates can still win?
- How should sincerity and strategyproofness be reformulated when agents express incomplete preferences?

Possible and necessary winners

More generally: *incomplete knowledge* of the voters' preferences.

For each voter: a *partial order* on the set of candidates:

$P = \langle P_1, \ldots, P_n \rangle$ incomplete profile

Completion of P: full profile $T = \langle T_1, \ldots, T_n \rangle$ of P, where each T_i is a linear ranking extending P_i.

Given a voting rule r, an incomplete profile P, and a candidate c:

- c is a *possible winner* if there exists a completion of P in which c is elected.
- c is a *necessary winner* if c is elected in every completion of P.

(Konczak and Lang, 05; Pini *et al.*, 07, 10; Xia & Conitzer, 08; Betzler and Dorn, 09; Betzler *et al.*, 09; Baumeister and Rothe, 10; Bachrach *et al.*; 10; Chevaleyre *et al.*, 10; Xia *et al.*, 11. Lu and Boutilier, 11; Baumeister *et al.*, 12; Brill *et al.*, 12; etc.)

Possible and necessary winners

$a \succ b, a \succ c$	$b \succ a$	$c \succ a \succ b$	plurality with tie-breaking priority $b > a > c$	Condorcet
abc	*cba*	*cab*	c	c
abc	*bca*	*cab*	b	-
abc	*bac*	*cab*	b	a
acb	*cba*	*cab*	c	c
acb	*bca*	*cab*	b	c
acb	*bac*	*cab*	c	a

Possible Condorcet winners: $\{a, c\}$; possible plurality$_{b>a>c}$-winners: $\{b, c\}$.

Possible and necessary winners

Two particular cases:

possible/necessary winners with respect to addition of voters

A subset of voters A have reported a full ranking; the other ones have not reported anything.

Links with coalitional manipulation:

- x is a possible winner if the coalition $N \setminus A$ has a constructive manipulation for x.
- x is a necessary winner if the coalition $N \setminus A$ has no destructive manipulation against x.

possible/necessary winners with respect to addition of candidates

The voters have reported a full ranking on a subset of candidates X (and haven't said anything about the remaining candidates).

Possible and necessary winners with respect to addition of candidates

New candidates sometimes come while the voting process is going on:

- Doodle: new dates become possible
- recruiting committee: a preliminary vote can be done before the last applicants are inrerviewed

Obviously: for any reasonable voting rule, any new candidate must be a possible winner.

Question: *who among the initial candidates can win?*

Example :

- $n = 12$ voters; initial candidates : $X = \{a, b, c\}$; one new candidate y.
- voting rule = plurality with tie-breaking priority $a > b > c > y$
- plurality scores before y is taken into account: $a \mapsto 5$, $b \mapsto 4$, $c \mapsto 3$.

Who are the possible winners?

Possible and necessary winners with respect to addition of candidates

General result for plurality: if P_X is the profile, X the initial candidates, $ntop(P_X, x)$ the number of voters who rank x in top position in P_X; then: $x \in X$ is a possible winner for P_X with respect to the addition of k new candidates *iff*

$$ntop(P_X, x) \geq \frac{1}{k}. \sum_{x_i \in X} \max(0, ntop(P_X, x_i) - ntop(P_X, x))$$

where $ntop(P_X, x)$ is the plurality score of x in P_X.

Possible and necessary winners with respect to addition of candidates

Example 2 :

- $n = 4$ voters; initial candidates : $X = \{a, b, c, d\}$; k new candidates $y_1, \ldots, y_k$.
- voting rule = Borda
- initial profile: $P = \langle bacd, bacd, bacd, dacb \rangle$.
  Borda scores: $a \mapsto 8$, $b \mapsto 9$, $c \mapsto 4$, $d \mapsto 3$.

Who are the possible winners, depending on the value of k?

Possible and necessary winners with respect to addition of candidates

Example 2 :

- $n = 4$ voters; initial candidates : $X = \{a, b, c, d\}$; k new candidates $y_1, \ldots, y_k$.
- voting rule = Borda
- initial profile: $P = \langle bacd, bacd, bacd, dacb \rangle$.

A useful lemma: x is a possible winner for P_X w.r.t. the addition of k new candidates if and only if x is the Borda winner for the profile on $X \cup \{y_1, \ldots, y_k\}$ obtained from P_X by putting $y_1, \ldots, y_k$ right below x (in an arbitrary order) in every vote of P_X.

Who are the possible winners, depending on the value of k??

- for any $k \geq 1$, a and b are possible winners;
- for any $k \geq 5$, a, b and d are possible winners;
- for any value of k, c is not a possible winner.

More general results in (Chevaleyre *et al.*, 10).

Introduction to protocols and communication complexity

Two key references:

- A.C Yao, Some complexity questions related to distributed computing, Proc. 11th ACM Symposium on Theory of Computing, 1979, 209-213
- E. Kushilevitz and N. Nisan, Communication complexity, Cambridge University Press, 1997.

Communication problem: a set of n agents has to compute a function $f(x_1, \ldots, x_n)$ given that the input is distributed among the agents: initially, agent 1 knows only x_1, $\ldots$, and agent n knows x_n.

Protocol: binary tree where each node is labelled with an agent and an action policy specifying a bit the agent should communicate, depending on her knowledge.

Informally: a protocol is similar to an algorithm, except that instructions are replaced by communication actions between agents, and such that communication actions are based on the *private information* of the agents.

Communication complexity of voting rules

From (Conitzer & Sandholm, 05).

- *Voting rule*

  $$r : \mathcal{P}^n \longrightarrow \mathcal{X}$$

  A voting rule does not specify how the votes are elicited from the voters by the central authority.

- *Protocol for a voting rule r*
  Communication protocol for computing $r(V_1, \ldots, V_n)$, given that V_i is the private information of agent (voter) i.

- *Communication complexity of a voting rule r*: minimum cost of a protocol for r.

Communication complexity of voting rules

A protocol for any voting rule r:

step 1 every voter i sends V_i to the central authority

$\hookrightarrow n \log(p!)$ bits

step 2 [the central authority sends back the name of the winner to all voters]

$\hookrightarrow n \log p$ bits

Corollary The communication complexity of an arbitrary voting rule r is at most $n. \log(p!) [+ n \log p]$

From now on, we shall ignore step 2.

Communication complexity of voting rules

Example 1: plurality

A simple protocol:

voters send the name of their most preferred candidate to the central authority
$\hookrightarrow n \log p$ bits

Corollary The communication complexity of plurality is at most $n . \log p$

Communication complexity of voting rules

Obtaining a lower bound: via the *fooling set* technique.

Details on request (off-line)

Proposition: the communication complexity of plurality with runoff is in $\Theta(n.\log p)$ (Conitzer & Sandholm, 05)

Communication complexity of voting rules

Example 2: plurality with runoff.

A protocol:

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

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

step 2 the central authority 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 the central authority

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

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

Corollary: the communication complexity of plurality with runoff is in $O(n. \log p)$.

The lower bound matches:

Proposition: the communication complexity of plurality with runoff is in $\Theta(n. \log p)$ (Conitzer & Sandholm, 05)

Communication complexity of voting rules

Example 3: Single Transferable Vote (STV): a protocol

step 1 voters send their most preferred candidate to the central authority (C)

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

step 2 let x be the candidate to be eliminated. All voters who had x ranked first receive a message from C asking them to send the name of their next preferred candidate. There were at most $\frac{n}{p}$ such voters

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

step 3 similarly with the new candidate y to be eliminated. At most $\frac{n}{p-1}$ voters voted for y

$\hookrightarrow$ $\mathbf{2\frac{n}{p-1}}\log\mathbf{p}$ bits

etc.

total $\leq 2n\log p(1+\frac{1}{p}+\frac{1}{p-1}+\ldots+\frac{1}{2}) = O(n.(\log p)^2)$.

Lower bound matches (Conitzer & Sandholm, 05)

Example 4: **Bucklin rule**:

Let q the smallest integer such that there exists a candidate x such that more than half of the voters rank x among their q preferred candidates. (Necessarily, $1 \leq q \leq \frac{p}{2}$.)

Then the winner is the candidate ranked in the q preferred candidates by the largest number of voters.

Optimal protocol for Bucklin?

Compilation complexity

(Chevaleyre *et al.*, 09; Xia and Conitzer, 10.)

Context: sometimes the votes do not come all together at the same time

- votes of the citizens living abroad known only a few days after the rest of the votes;
- choosing a date for a meeting: some participants vote later than others.

$\Rightarrow$ preprocess the information given by the subelectorate so as to prepare the ground for the time when the last votes are known, using *as little space as possible*.

Input only $m \leq n$ votes have been expressed.
$P = \langle V_1, \ldots, V_m \rangle$ = corresponding *partial* prole.

Question what is the minimal size needed to compile P, while still being able to compute r when the last votes come in?

A context where it is useful to compile the vote of a subelectorate: *verification of the outcome of a vote by the population*.

- the electorate is split into different districts; each district counts its ballots separately and communicates the outcome to the Ministry of Innner Affairs, which, after gathering the outcomes from all districts, determines the final outcome;
- in each district, the voters can check that the local results are sound;
- local results are made public and voters can check the final outcome from these local outcomes.

space needed to synthesize the votes of a district

= amount of information the district has to send to the central authority

If this amount of information is too large, it is impractical to publish the results locally, and therefore, difficult to check the final outcome and voters may be reluctant to accept the voting rule.

Compilation complexity

$\rho(\sigma(P), R) = r(P \cup R)$

σ compilation function

Example: r_B = Borda.

$\sigma(P)$: vector of partial Borda scores $\langle s_B(x \mid P)\rangle_{x \in X}$

$P = \langle abc, abc, cba, bca\rangle \mapsto \sigma(P) = \langle a:3; b:5; c:3\rangle$.

$\rho(\sigma(P), R) = \text{argmax}_{x \in X}\,(s_B(x \mid P) + s_B(x \mid R))$

$R = \langle cab, abc\rangle \mapsto \langle a:3+3; b:5+1; c:3+2\rangle \mapsto \rho(\sigma(P), R) = b$.

Compilation complexity

Size of a compilation function

Let σ be a compilation function for r

$$Size(\sigma) = \max\{|\sigma(P)| \mid P \text{ partial profile}\}$$

Compilation complexity of r:

$$C(r) = \min\{Size(\sigma) \mid \sigma \text{ compilation function for } r\}$$

$C(r)$ is the minimum space needed to compile the partial profile P

Compilation complexity and one-round communication complexity

One-round communication complexity :

- two agents A and B have to compute a function f.
- each of them knows only a part of the input.
- *one-round protocol*: A sends only one message to B, and then B sends the output to A.
- *one-round communication complexity* of f: worst-case number of bits of the best one-round protocol for f.

One-round communication complexity $\approx$ compilation complexity

- A = set of voters having already expressed their votes
- B = set of remaining voters;
- compilation of the votes of A = information that A must send to B.
- minor difference: B does not send back the output to A.

Equivalent profiles for a voting rule

r voting rule;
k number of remaining voters.

Two partial profiles P and Q are *equivalent for* r if no matter the remaining votes, they will lead to the same outcome:

$$\text{for every } R \text{ we have } r(P \cup R) = r(Q \cup R)$$

Example: r_P = plurality with tie-breaking priority $b > a > c$.

- $\langle abc, abc, bac, bac \rangle$ and $\langle acb, acb, bca, bca \rangle$ are equivalent for r_P;
- $P_1 = \langle abc, abc \rangle$ and $P_2 = \langle abc, bac \rangle$ are not equivalent for r_P: take $R = \langle bca, bca \rangle$, then $r_P(P_1 \cup R) = a \neq r_P(P_2 \cup R) = b$.

A useful result (similar result in (Kushilevitz & Nisan, 97)):

- r voting rule.
- m number of initial voters
- p number of candidates.

If the equivalence relation for r has $g(m,p)$ equivalence classes then

$$C(r) = \lceil \log g(m,p) \rceil$$

Corollary:

- for any voting rule r, $C(r) \leq m \log(p!)$;
- for any anonymous voting rule r, $C(r) \leq \min(m \log(p!), p! \log m)$.
- the compilation complexity of a dictatorship is $\log p$;
- the compilation complexity of r is 0 if and only if r is constant.

Compilation complexity of voting rules:

- *plurality*: P and P' are equivalent *iff* for all x, $ntop(P,x) = ntop(P',x)$, where $ntop(P,x)$ be the number of votes in P ranking x first.
- *Borda*: P and P' are equivalent *iff* for all x, $score_B(x,P) = score_B(x,P')$, where $score_B(x,P)$ = Borda score of x obtained from the partial profile P
- *rules based on the majority graph*: For any Condorcet-consistent rule based on the (unweighted/weighted) majority graph, P and P' are equivalent *iff* $\mathcal{M}_P = \mathcal{M}_{P'}$, where $\mathcal{M}_P$ is the *weighted* majority graph associated with P.
- *plurality with runoff*: P and Q are equivalent *iff* these two conditions hold: (a) for every x, $ntop(P,x) = ntop(Q,x)$; and (b) $\mathcal{M}_P = \mathcal{M}_Q$.
- *STV*: P and Q are equivalent *iff* for all subset of candidates V and $x \in V$, $ntop(P_{-V},x) = ntop(Q_{-V},x)$
  (For any set of candidates that can possibly be eliminated, the plurality scores of the remaining candidates must be the same in P and Q.)

Results on compilation complexity follow from these results by computing bounds on the number of equivalence classes.

Incomplete knowledge and communication complexity

Other issues:

- voting with partial ballots: strategical issues (Pini *et al.*, 07; Endriss *et al.*, 09)
- communication issues with single-peaked preferences (Trick, 89; Doignon, 05; Conitzer, 08; Escoffier *et al.*, 08)
- sequential announcements of votes (Pacuit and Parikh, 06; Airiau and Endriss, 09; Elkind *et al.*, 09; Xia and Conitzer, 10)
- voting with potentially unavailable candidates (Lu and Boutilier, 10; Boutilier *et al.*, 12;).

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

Fair division

A very rough taxonomy:

- nature of the goods: indivisible (hard candies) or divisible (cake)?
- nature of the protocol: centralized (elicitation + allocation) or distributed?

In the rest of talk we focus on *centralized* protocols for *indivisible* goods.

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
= a specific group decision making problem with a combinatorial set of alternatives

Resource allocation $\neq$ fair division

Combinatorial auctions

$V_i : 2^{\mathcal{R}} \rightarrow I\!N$ 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:

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

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* $u_i : 2^{\mathcal{R}} \rightarrow L$ 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*

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

Fair division

- social choice theory: *axiomatic study of criteria*
- AI & OR: elicitation/communication, compact representation, computation.

Examples of recent works:

- approximate envy-freeness (Lipton *et al.*, Markakis, Mossel and Saberi 04; Chevaleyre, Endriss and Maudet, 07]
- logical representation + complexity results for envy-free allocations (Bouveret and Lang 05; de Keizer *et al.*, 09; Rothe *et al.*, 10; etc.)
- complexity and communication issues in *distributed* allocation (Dunne *et al.*, 05; Chevaleyre *et al.*, 04)
- sequential elicitation-free allocation: agents pick items in an 'optimal' predefined sequence (Brams and Taylor, 00; Bouveret and Lang, 10, Davies *et al.*, 12).

1. Social choice and computational social choice
2. Preference aggregation, Arrow's theorem, and how to escape it
3. Voting rules: easy
4. Voting rules: hard
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Fair division and social welfare
9. Judgment aggregation
10. Other issues

Computational social choice: other issues

- learning voting rules
- robustness of voting rules
- complexity issues in strategic sequential voting
- complexity or manipulating stable marriage problems
- fairness and uncertainty
- complexity issues in judgment aggregation
- etc.