

Computational social choice

Jérôme Lang

LAMSADE

CNRS – Université Paris-Dauphine

ICAART-16

Rome, February 26, 2016

Plan

1. From social choice to computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Social choice theory

- ▶ Social choice: *designing and analysing methods for collective decision making*
- ▶ Some examples of social choice problems:
 - ▶ elections
 - ▶ deciding where and when to have dinner altogether tonight
 - ▶ Doodle polls (find a date for a meeting)
 - ▶ in a divorce settlement: deciding how to divide the bank account, who will have the children's custody, who keeps the stereo and who keeps the cat.
 - ▶ in a jury: agreeing on a verdict.

Social choice theory

- ▶ Social choice: *designing and analysing methods for collective decision making*
 - ▶ Some examples of social choice problems:
 - ▶ elections
 - ▶ deciding where and when to have dinner altogether tonight
 - ▶ Doodle polls (find a date for a meeting)
 - ▶ in a divorce settlement: deciding how to divide the bank account, who will have the children's custody, who keeps the stereo and who keeps the cat.
 - ▶ in a jury: agreeing on a verdict
- aggregating preferences

Social choice theory

- ▶ Social choice: *designing and analysing methods for collective decision making*
- ▶ Some examples of social choice problems:
 - ▶ elections
 - ▶ deciding where and when to have dinner altogether tonight
 - ▶ Doodle polls (find a date for a meeting)
 - ▶ in a divorce settlement: deciding how to divide the bank account, who will have the children's custody, who keeps the stereo and who keeps the cat.
 - ▶ in a jury: agreeing on a verdict

vs. aggregating opinions/beliefs

Social choice theory

► Formally:

1. a set of agents $N = \{1, \dots, n\}$;
2. a set of alternatives $X = \{x_1, \dots, x_m\}$;
3. each agent i has some preferences on the alternatives

⇒ choosing a socially preferred alternative

► Two important subdomains of social choice:

- *Voting*: agents (*voters*) express their preferences on a set of alternatives (for instance *candidates*) and must choose one collectively.
- *Fair division*: agents express their preferences over combinations of resources they may receive and an allocation must be found.

Social choice theory

- ▶ each agent i has some **preferences** on the alternatives

Usual models for preferences:

- ▶ *cardinal preferences*: each agent has a *utility function* $u: \mathcal{X} \rightarrow \mathbb{R}$
- ▶ *ordinal preferences*: each agent has a *preference relation* \succeq on \mathcal{X} (**most common assumption in social choice**)
- ▶ *dichotomous preferences*: each agent has a partition $\{\text{Good}, \text{Bad}\}$ of \mathcal{X}

A very rough history of social choice

1. end of 18th century: early stage, with Condorcet and Borda (talk at ICAART-1789)
2. 1951: birth of modern social choice
 - ▶ results are mainly *axiomatic* (economics/mathematics)
 - ▶ impossibility theorems: *incompatibility of a small set of seemingly innocuous conditions*, such as **Arrow's theorem**:
With at least 3 alternatives, an aggregation function satisfies unanimity and independence of irrelevant alternatives if and only if it is a dictatorship.
 - ▶ computational issues are neglected
3. early 90's: computer scientists come into play
⇒ **Computational social choice**

What is computational social choice?

Two research streams:

from computer science to social choice

Using computational notions and techniques (mainly from Artificial Intelligence and Operations Research) for solving complex collective decision problems.

from social choice to computer science

Using social choice concepts and procedures for solving questions arising in CS / AI application domains, especially in *multi-agent systems*.

- ▶ managing societies of autonomous agents
- ▶ fair division of computational resources
- ▶ ranking systems for internet search engines
- ▶ group recommendation
- ▶ etc.

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Voting

1. a finite *set of voters* $N = \{1, \dots, n\}$;
2. a finite *set of alternatives (candidates)* $\mathcal{X} = \{x_1, \dots, x_m\}$;
3. a *profile* = a collection of n preference relations on \mathcal{X}

$$P = (\succ_1, \dots, \succ_n)$$

\succ_i = linear order over \mathcal{X} = *vote* expressed by voter i .

Here is a 100-voter profile over $\mathcal{X} = \{a, b, c, d, e\}$

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

Aggregation functions, rules, correspondences

1. a finite *set of voters* $N = \{1, \dots, n\}$;
2. a finite *set of alternatives (candidates)* $X = \{x_1, \dots, x_m\}$;
3. a *profile* = a collection of n preference relations on X

$$P = (\succ_1, \dots, \succ_n)$$

\succ_i = linear order over X = *vote* expressed by voter i .

Resolute voting rule

$P \mapsto r(P) \in X$: collectively preferred candidate

Irresolute voting rule

$P \mapsto C(P) \in 2^X \setminus \{\emptyset\}$: set of collectively preferred candidates, or *co-winners*. (Only one of them will be the final winner but the rule does not specify it.)

Resolute vs. irresolute rules

- ▶ Suppose we have only two candidates a, b
- ▶ Obvious voting rule: **majority**
- ▶ Suppose that P consists of two votes:

$$P = (a \succ b, b \succ a)$$

- ▶ With irresolute rules: this is not a problem: a, b *cowinners*

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

- ▶ With voting rules: we need a *tie-breaking mechanism*
 - ▶ give up *neutrality*: use a predefined priority relation on candidates (e.g. preference for the oldest candidate)
 - ▶ give up *anonymity*: use a predefined strict relation on voters (e.g. priority given to the chair)

Resolute vs. irresolute rules

The usual way of defining voting rules:

- ▶ we first define an **irresolute rule** F
- ▶ a **resolute rule** is defined from F by using a **tie-breaking priority** T
- ▶ usual assumption: $T =$ linear order on \mathcal{X}
- ▶ $F_T(P) = \max(T, F(P))$: F_T resolute rule

Example:

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

In the rest of the talk, we usually define irresolute rules, from which resolute rules are induced by a tie-breaking priority.

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?

Voting with more than three candidates

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? Sometimes yes:

33 votes: $a \succ b \succ c \succ d \succ e$

16 votes: $b \succ d \succ c \succ e \succ a$

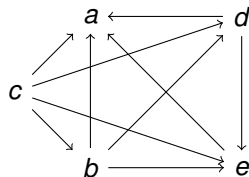
3 votes: $c \succ d \succ b \succ a \succ e$

8 votes: $c \succ e \succ b \succ d \succ a$

18 votes: $d \succ e \succ c \succ b \succ a$

22 votes: $e \succ c \succ b \succ d \succ a$

associated majority graph



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

Winner: c

Voting with more than three candidates

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? Sometimes no:

33 votes: $a \succ b \succ \mathbf{d} \succ \mathbf{c} \succ e$

16 votes: $b \succ d \succ c \succ e \succ a$

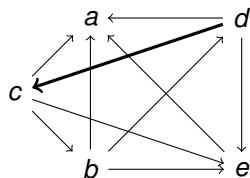
3 votes: $c \succ d \succ b \succ a \succ e$

8 votes: $c \succ e \succ b \succ d \succ a$

18 votes: $d \succ e \succ c \succ b \succ a$

22 votes: $e \succ c \succ b \succ d \succ a$

associated majority graph

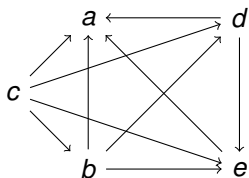


Collective preference relation: $\{b \succ d \succ c \succ b \succ \dots\} \succ e \succ a$;

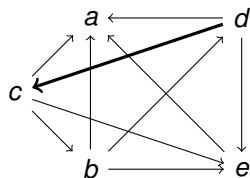
Winner: ?

Condorcet winner

- ▶ $N(x,y) = \#\{i, x \succ_i y\}$ number of voters who prefer x to y .
- ▶ x *Condorcet winner* if for all $y \neq x$, $N(x,y) > \frac{n}{2}$



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.

Approval voting

- ▶ not all rules take rankings as input.
- ▶ *approval vote* = a subset of (approved) candidates $A \subseteq X$
- ▶ *approval profile* = a collection of approval votes

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

- ▶ Winner(s): candidate(s) approved most often.
 - ▶ $X = \{a, b, c, d, e\}$
 - ▶ $n = 5$
 - ▶ $P = \langle \{a, c\}, \{b, c, d\}, \emptyset, \{a, b, c, d, e\}, \{d\} \rangle$
 - ▶ c and d approved by 3 voters: a and b by 2 voters; e by 1.
 - ▶ cowinners: $\{c, d\}$

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Positional scoring rules

- ▶ n voters, m candidates
- ▶ fixed list of m integers $s_1 \geq \dots \geq s_m$, with $s_1 > s_m$
- ▶ if voter i ranks candidate x in position j then $score_i(x) = s_j$
- ▶ winner(s): candidate(s) maximizing

$$s(x) = \sum_{i=1}^n score_i(x)$$

Three important examples:

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

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

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

Positional scoring rules

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

Positional scoring rules

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 $s_3 = 0$.

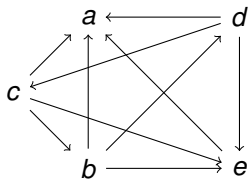
- ▶ $S(a) = 6s_1 + 7s_2$
- ▶ $S(b) = 8s_1 + 6s_2$
- ▶ $S(b) - S(a) = 2s_1 - s_2 = s_1 + (s_1 - s_2) > 0$
- ▶ $S(b) > S(a)$ whatever the value of s_1 and s_2
- ▶ but a is a Condorcet winner!

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 .
- ▶ For the sake of simplicity, we assume an odd number of voters; in this case the majority graph is a complete asymmetric graph: a **tournament**.

Copeland

- ▶ $C(x)$ = number of candidates y such that $M(P)$ contains $x \rightarrow y$.
- ▶ Copeland winner(s): maximize(s) C .



$$C(a) = 0$$

$$C(b) = 3$$

$$C(c) = 3$$

$$C(d) = 3$$

$$C(e) = 1$$

cowinners: **b, c, d**

Condorcet-consistent rules

- ▶ $N_P(x, y) = \#\{i, x \succ_i y\}$ number of voters 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 .

maximin

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

N_P	a	b	c	d	e	$S_m(\cdot)$
a	—	33	33	33	36	33
b	67	—	49	79	52	49
c	67	51	—	33	60	33
d	67	21	67	—	70	21
e	66	48	40	30	—	30

winner: b

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 $y \succ_{n+1} x$.

- ▶ for $m \geq 4$, no Condorcet-consistent rule satisfies participation
- ▶ Proof for maximin:

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

$$S_m(a) = N(a, c) = 6;$$

$$b, d : 5; c : 4;$$

maximin winner: a

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

$$S_m(a) = 6; S_m(b) = 7; \text{ etc.}$$

maximin winner: b

- ▶ The four new voters had rather stayed home!
- ▶ Similar paradox for *reinforcement* (if two disjoint electorates elect the same candidate x then their union still elects x)

Multiple round rules

plurality with runoff

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

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

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

Multiple round rules

Single transferable vote (STV)

Repeat

x := candidate ranked first by the fewest voters;

eliminate x from all ballots

{votes for x transferred to the next best remaining candidate}

Until some candidate y is ranked first by more than half of the votes;

Winner: y

- ▶ 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

33	a	✓	b	✓	d	✓	e
16	b	✓	d	✓	e	✓	a
3	d	✓	b	✓	a	✓	e
8	e	✓	b	✓	d	✓	a
18	d	✓	e	✓	b	✓	a
22	e	✓	b	✓	d	✓	a

33	a	✓	d	✓	e
16	d	✓	e	✓	a
3	d	✓	a	✓	e
8	e	✓	d	✓	a
18	d	✓	e	✓	a
22	e	✓	d	✓	a

33	a	✓	d
16	d	✓	a
3	d	✓	a
8	d	✓	a
18	d	✓	a
22	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]
- ▶ plurality with runoff and STV also fail *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

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]
- ▶ plurality with runoff and STV also fail *monotonicity*:

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

▶ b eliminated

▶ winner: a

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

▶ c eliminated

▶ winner: b

- ▶ they also fail participation and reinforcement

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Computing voting rules

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

- ▶ scoring rules, plurality with runoff, approval: $O(nm)$
- ▶ Copeland, maximin, STV: $O(nm^2)$.

But some voting rules are NP-hard.

Hard rules: Kemeny

- ▶ *Kemeny distance*: $d_K(\gamma, \gamma') =$ number of $(x, y) \in \mathcal{X}^2$ on which γ and γ' disagree

$$d_K(\gamma, \langle \gamma_1, \dots, \gamma_n \rangle) = \sum_{i=1, \dots, n} d_K(\gamma, \gamma_i)$$

- ▶ *Kemeny consensus* = order minimizing $d_K(\cdot, \langle \gamma_1, \dots, \gamma_n \rangle)$
- ▶ *Kemeny winner* = candidate ranked first in a Kemeny consensus

Hard rules: Kemeny

Observation: Kemeny is based on the weighted majority graph.

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

Computing $d(a \succ b \succ c, \langle \succ_1, \dots, \succ_9 \rangle)$:

N	a	b	c
a	—	6	4
b	3	—	7
c	5	2	—

- ▶ 3 voters disagree with $a \succ b$
- ▶ 5 voters disagree with $a \succ c$
- ▶ 2 voters disagree with $b \succ c$
- ▶ hence $d(a \succ b \succ c, \langle \succ_1, \dots, \succ_9 \rangle) = 10$.

Kemeny scores:

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

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

Hard rules: Kemeny

- ▶ NP-hard (Bartholdi *et al.*, 89; Hudry, 89)
- ▶ exact complexity: deciding whether a candidate is a Kemeny winner is Θ_2^P -complete (Hemaspaandra *et al.*, 04)
- ▶ existence of polynomial approximation algorithms
- ▶ good heuristics
- ▶ otherwise, satisfy many good properties
- ▶ several other important rules are also hard to compute

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.
 - ▶ The approximation of a voting rule is a new voting rule that may be interesting *per se*.

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. **Combinatorial domains**
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Voting in combinatorial domains

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

Example 1 choosing a common menu:

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

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 recruiting committee (3 positions, 6 candidates):

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

Combinatorial domains:

- ▶ $\mathcal{V} = \{X_1, \dots, X_p\}$ set of *variables*, or *issues*;
- ▶ $\mathcal{X} = D_1 \times \dots \times D_p$
- ▶ for each i , D_i is a finite value domain for variable X_i

Voting in combinatorial domains

- ▶ Two binary variables:
 - ▶ S (build a new swimming pool or npt)
 - ▶ T (build a new tennis court or not)
- ▶ 5 voters with their preferences:

voters 1 and 2 $S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$

voters 3 and 4 $\bar{S}T \succ ST \succ \bar{S}\bar{T} \succ \bar{S}T$

voter 5 $ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$

Voting in combinatorial domains

- ▶ Two binary variables:
 - ▶ S (build a new swimming pool or npt)
 - ▶ T (build a new tennis court or not)

- ▶ 5 voters with their preferences:

voters 1 and 2 $S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$

voters 3 and 4 $\bar{S}T \succ ST \succ \bar{S}\bar{T} \succ ST$

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 $\{T, \bar{T}\}$

Voting in combinatorial domains

- ▶ Two binary variables:
 - ▶ S (build a new swimming pool or npt)
 - ▶ T (build a new tennis court or not)

- ▶ 5 voters with their preferences:

voters 1 and 2 $S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$

voters 3 and 4 $\bar{S}T \succ ST \succ \bar{S}\bar{T} \succ ST$

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!

- ▶ Several classes of methods for solving this problem.

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. **Strategic behaviour**
7. Communication issues and incomplete preferences
8. Other issues

Manipulation and strategyproofness

- ▶ *Manipulation*: a coalition of voters report insincere preferences so as to elect a better candidate.
- ▶ Example: $r = \textit{plurality with runoff}$

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

Manipulation and strategyproofness

- ▶ *Manipulation*: a coalition of voters report insincere preferences so as to elect a better candidate.
- ▶ Example: $r = \text{plurality with runoff}$

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

2 $c \succ a \succ b$
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?

Manipulation and strategyproofness

- ▶ *Manipulation*: a coalition of voters report insincere preferences so as to elect a better candidate.
- ▶ Example: $r = \text{plurality with runoff}$

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

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

1st round: c eliminated
2nd round: b elected

1st round: b eliminated
2nd round: a elected

- ▶ Is this a specific flaw of plurality with runoff?
- ▶ Unfortunately no...

Gibbard and Satterthwaite's theorem (73/75)

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

Escaping Gibbard and Satterthwaite

One solution for *nearly escaping* Gibbard and Satterthwaite:

Computational barrier

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

CONSTRUCTIVE MANIPULATION for voting rule r :

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

Question Is it possible for voters $k + 1, \dots, n$ 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

- ▶ 4 voters:

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

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

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

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

- ▶ Current Borda scores

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

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

$a?$

Complexity of manipulation

- ▶ Manipulating the Borda rule by a single voter

- ▶ 4 voters:

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

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

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

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

- ▶ Current Borda scores

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

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

$a?$ $b?$

Complexity of manipulation

- ▶ Manipulating the Borda rule by a single voter

- ▶ 4 voters:

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

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

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

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

- ▶ Current Borda scores

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

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

$a?$ $b?$ $c?$

Complexity of manipulation

- ▶ Manipulating the Borda rule by a single voter

- ▶ 4 voters:

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

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

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

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

- ▶ Current Borda scores

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

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

$a?$ $b?$ $c?$ $d?$

Complexity of manipulation

- ▶ Manipulating the Borda rule by a single voter

- ▶ 4 voters:

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

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

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

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

- ▶ Current Borda scores

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

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:

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

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

Complexity of manipulation

Existence of a manipulation for the Borda rule:

- ▶ *for a single voter* : in P
- ▶ *for a coalition of at least two voters* : NP-complete

Complexity of manipulation

Number of manipulators	1	at least 2
Copeland	P	NP-complete
STV	NP-complete	NP-complete
veto	P	P
cup	P	P
maximin	P	NP-complete
ranked pairs	NP-complete	NP-complete
Bucklin	P	P
Borda	P	NP-complete

Complexity of manipulation

An important concern:

- ▶ a worst-case NP-hardness results only says that *sometimes* (maybe rarely), computing a manipulation will be hard
- ▶ *negative* results about the *average hardness* of manipulation: under reasonable assumptions, with a high probability, if there is a manipulation then it can be found by a greedy algorithm a manipulation with high probability.

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Incomplete knowledge and communication

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)

Incomplete knowledge and communication

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:

$a: 14 ; b: 10 ; c: 14 ; d: 10$

- ▶ only need to know the last voter's preference between a and c

Possible and necessary winners

- ▶ *incomplete knowledge* of the voters' preferences.
- ▶ For each voter: a *partial order* on the set of candidates:

$$P = \langle P_1, \dots, P_n \rangle \text{ incomplete profile}$$

- ▶ *Completion of P* : full profile $T = \langle T_1, \dots, T_n \rangle$ 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 .

Possible and necessary winners

$a \succ b, a \succ c$	$b \succ a$	$c \succ a \succ b$	plurality with tie-breaking $b > a > c$	Condorcet
<i>abc</i>	<i>cba</i>	<i>cab</i>	c	c
<i>abc</i>	<i>bca</i>	<i>cab</i>	b	-
<i>abc</i>	<i>bac</i>	<i>cab</i>	b	a
<i>acb</i>	<i>cba</i>	<i>cab</i>	c	c
<i>acb</i>	<i>bca</i>	<i>cab</i>	b	c
<i>acb</i>	<i>bac</i>	<i>cab</i>	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.
- ▶ x is a possible winner if the coalition $N \setminus A$ has a constructive manipulation for 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 interviewed
- ▶ Question: *who among the initial candidates can win?*
- ▶ Example :
 - ▶ 12 voters; initial candidates $\{a, b, c\}$; one new candidate y
 - ▶ voting rule = plurality with tie-breaking priority $a > b > c > y$
 - ▶ plurality scores before y is a candidate: $a \mapsto 5, b \mapsto 4, c \mapsto 3$
 - ▶ y and a possible winners (obvious)
 - ▶ b possible winner: if three of the voters who voted for a now vote for y then the new scores are $a \mapsto 2, b \mapsto 4, c \mapsto 3, y \mapsto 3$
 - ▶ c not a possible winner; we'd need 4 of the voters who voted for a and 2 of the voters who voted for b to vote for y ; but then y would have a score ≥ 5 and c has a score ≤ 3 .

Plan

1. Social choice and computational social choice
2. Voting
3. Voting rules: easy to compute
4. Voting rules: hard to compute
5. Combinatorial domains
6. Strategic behaviour
7. Communication issues and incomplete preferences
8. Other issues

Computational social choice: beyond voting

- ▶ Iterated voting
- ▶ Fair division of indivisible resources
- ▶ Fair division of divisible resources (cake-cutting protocols)
- ▶ Coalition formation, stable matching
- ▶ Judgment aggregation

Bibliography

- ▶ *Handbook of Computational Social Choice* (F. Brandt, V. Conitzer, U. Endriss, J. Lang, A. Procaccia, eds.). Cambridge University Press, 2016 (to appear extremely soon).

An experimental voting platform: *Whale* (developed by Sylvain Bouveret, University of Grenoble)

<http://whale3.noiraudes.net/>

Contacting me: `lang@lamsade.dauphine.fr`