# Voting in combinatorial domains: seven ways to proceed

Jérôme Lang
LAMSADE, CNRS & Université Paris-Dauphine

Talk mainly based on the following papers:

- Jérôme Lang & Lirong Xia, Sequential voting on multi-issue domains, *Mathematical Social Science*, 2009 (from *IJCAI-07* + *TARK-07*).
- Lirong Xia, Vincent Conitzer & Jérôme Lang, Voting on multiattribute domains with cyclic preferential dependencies. *AAAI-08*.
- Vincent Conitzer, Jérôme Lang & Lirong Xia, How hard is it to control sequential elections via the agenda? *IJCAI-09*
- Lirong Xia, Vincent Conitzer & Jérôme Lang, Strategic Sequential Voting in Multi-issue Domains and Multiple-Election Paradoxes. *LOFT-10*

## Voting

- a finite *set of voters* $\mathcal{A} = \{1, ..., n\}$;
- a finite *set of candidates (alternatives)* $\mathcal{X}$;
- *profile*:

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

$\succ_i$ = linear order on $\mathcal{X}$ (vote) expressed by voter $i$.

A **voting rule** maps every profile to a candidate.

A **voting correspondence** maps every profile to a nonempty subset of candidates.

Rules can be obtained from correspondences by tie-breaking (usually by using a predefined priority order on candidates).

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

**Example 1** choosing a common menu:

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

**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 (Ephrati & Rosenschein, 93; Klamler & Pfirschy, 07). A group of friends has to travel together to a sequence of possible locations, given some constraints on the possible sequences.

**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 $X_i$)

**How should such a vote be conducted?**

Some classes of solutions:

1. don't bother and vote separately on each variable.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. limit the number of possible alternatives that voters may vote for.
4. 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.
5. ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.
6. *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.
7. use a *compact preference representation language* in which the voters' preferences are represented in a concise way.

**How should such a vote be conducted?**

Some classes of solutions:

1. **don’t bother and vote separately on each variable (simultaneously).**

**How should such a vote be conducted?**

Some classes of solutions:

1. **don't bother and vote separately on each variable**: *multiple election paradoxes* arise as soon as some voters have preferential dependencies between attributes.

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

**How should such a vote be conducted?**

Some classes of solutions:

1. **don't bother and vote separately on each variable**: *multiple election paradoxes* arise as soon as some voters have preferential dependencies between attributes.

**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?**

Some classes of solutions:

1. **don't bother and vote separately on each variable**: *multiple election paradoxes* arise as soon as some voters have preferential dependencies between attributes.

Not too bad when all voters have *separable* preferences: *the preference over the possibles values of a variable is independent from the values of other variables*

A preference relation is separable if for every $X_i \in \mathcal{V}$, every $\vec{x}_{-i}, \vec{x}'_{-i} \in D_{-i}$, and every $x_i, x'_i \in D_i$,

$$(\vec{x}_{-i}, x_i) \succeq (\vec{x}_{-i}, x'_i) \text{ iff } (\vec{x}'_{-i}, x_i) \succeq (\vec{x}'_{-i}, x'_i)$$

Underlying assumption in *multi-winner elections* (Meir *et al.*, 08).

**How should such a vote be conducted?**

Some classes of solutions:

1. don't bother and vote separately on each variable.
2. **ask voters to specify their preference relation by ranking all alternatives *explicitly***.

$\mathcal{V} = \{X_1, \ldots, X_p\}$; $\mathcal{X} = D_1 \times \ldots \times D_p$

There are $\Pi_{1 \leq i \leq p} |D_i|$ alternatives.

$\Rightarrow$ **as soon as there are more than three or four variables, explicit preference elicitation is irrealistic.**

**How should such a vote be conducted?**

Some classes of solutions:

1. vote separately on each variable, in parallel.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. **limit the number of possible alternatives that voters may vote for.**

- *arbitrary* (who decides which alternatives are allowed?)
- so that this solution be realistic, the number of alternatives the voters can vote for has to be low. Therefore, voters only express their preferences on a tiny fraction of the alternatives.

**How should such a vote be conducted?**

Some classes of solutions:

1. don't bother and vote separately on each variable.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. limit the number of possible alternatives that voters may vote for.
4. **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.**

*Results are completely nonsignificant* as soon as the number of variables is much higher than the number of voters ($2^p \gg n$).

5 voters, $2^6$ alternatives; rule : plurality

001010: 1 vote; 010111: 1 vote; 011000: 1 vote; 101001: 1 vote; 111000: 1 vote
all other candidates : 0 vote.

**How should such a vote be conducted?**

Some classes of solutions:

1. don't bother and vote separately on each variable.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. limit the number of possible alternatives that voters may vote for.
4. 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.
5. **ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.**

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

- every voter specifies one 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.

An example of such an approach: **Minimax approval voting** (Brams, Kilgour & Sanver, 2007)

- $n$ voters, $m$ candidates, $k \leq m$ positions to be filled
- each voter casts an approval ballot $V_i = (v_i^1, \ldots, v_i^m) \in \{0,1\}^m$
  $v_i^j = 1$ if voter $i$ approves candidate $j$.
- 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*: $n = 4$, $m = 4$, $k = 2$.

| | 1110 | 1101 | 1010 | 1010 | *sum* | *max* |
|---|---|---|---|---|---|---|
| 1100 | 1 | 1 | 2 | 2 | 6 | **2** |
| 1010 | 1 | 3 | 0 | 0 | **4** | 3 |
| 1001 | 3 | 1 | 3 | 3 | 10 | 3 |
| 0110 | 1 | 3 | 2 | 2 | 8 | 3 |
| 0101 | 3 | 1 | 4 | 4 | 12 | 4 |
| 0011 | 3 | 3 | 2 | 2 | 10 | 3 |

- finding an optimal subset is NP-hard (Frances & Litman, 97)
- (Le Grand, Markakis & Mehta, 07; Caragiannis, Kalaitzis & Markakis, 10): approximation algorithms.

**How should such a vote be conducted?**

Some classes of solutions:

1. don’t bother and vote separately on each variable.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. limit the number of possible alternatives that voters may vote for.
4. 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.
5. ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.
6. ***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.**

## Sequential voting

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

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

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

Fix the order $S > T$.

**Step 1** elicit preferences on $\{S, \bar{S}\}$

if voters report preferences optimistically: $3 : S \succ \bar{S}$; $2 : \bar{S} \succ S$

**Step 2** compute the local outcome and broadcast the result

$$S$$

**Step 3** elicit preferences on $\{T, \bar{T}\}$ given the outcome on $\{S, \bar{S}\}$

4: $S : \bar{T} \succ T$; 1: $S : T \succ \bar{T}$

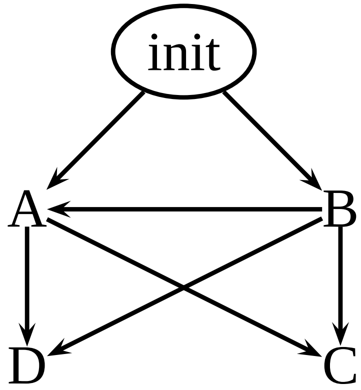
**Step 4** compute the final outcome

$$S\bar{T}$$

Sequential voting: another example

Inspired from (Ephrati & Rosenschein, 93) and (Klamler & Pfirschy, 07)

*A group of agents has to travel together to a set of places of interest. Time constraints imply that they can only visit two places. Transportation constraints imply that not all edges are feasible. At each time step they decide which location they should visit next.*




Three agents with separable preferences:

| | |
|---|---|
| agent 1 | $A \succ B \succ C \succ D$ |
| agent 2 | $B \succ D \succ A \succ C$ |
| agent 3 | $A \succ D \succ B \succ C$ |

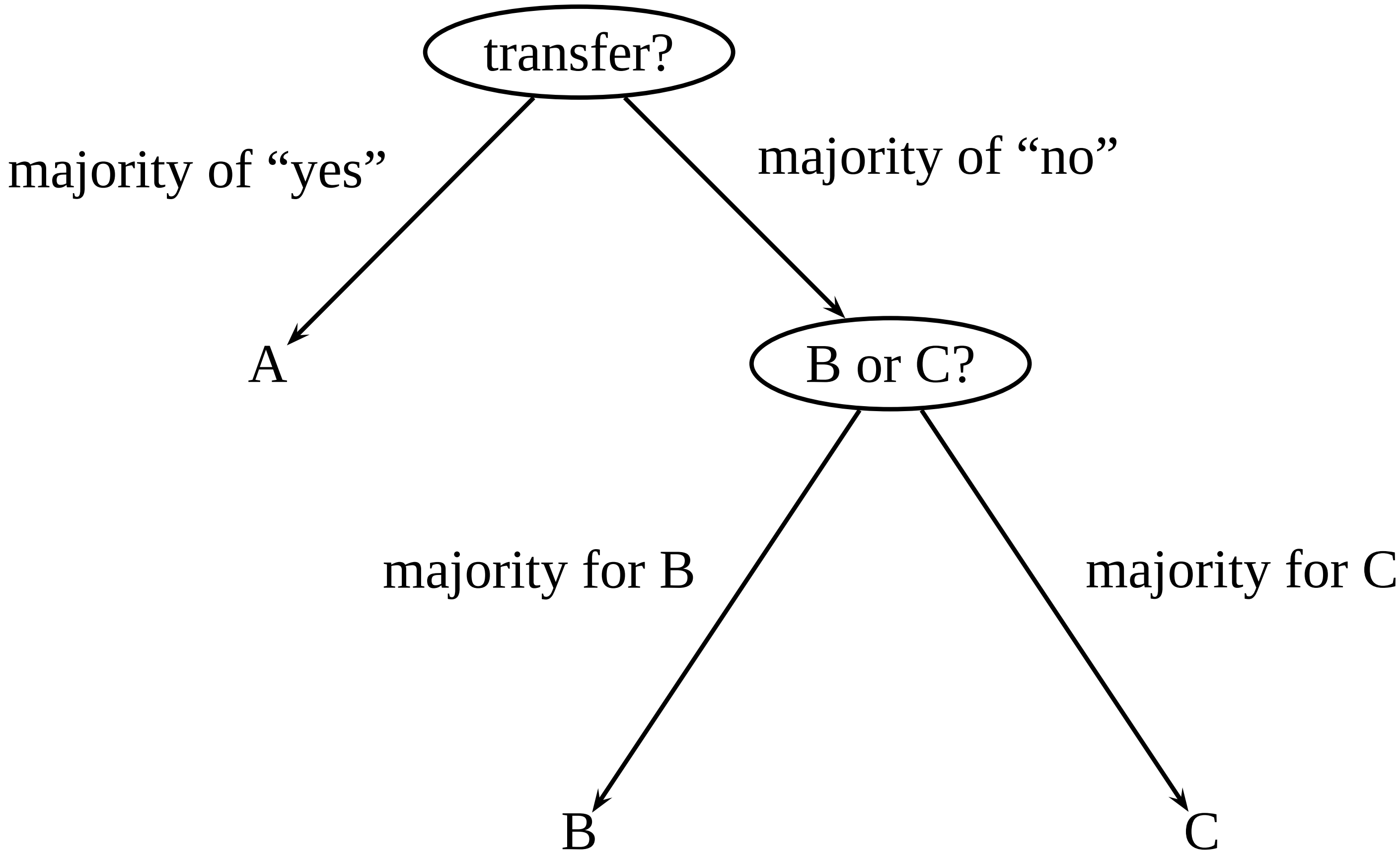
Sequential voting: yet another example

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

*A already has a position in another university.*

*B and C do not have any position.*

*The law requires the recruiting committee to consider transfers first.*



## Sequential voting

- $+$ simple elicitation protocol
- $+$ computationally easy (provided local rules are easy to compute)

Two possibilities:

- restriction-free sequential voting
- “safe” sequential voting

## “Safe” sequential voting

Prerequisite: **conditional preferential independence** (Keeney & Raiffa, 76)

$\{\mathcal{X}, \mathcal{Y}, \mathcal{Z}\}$ partition of $\mathcal{V}$.

$D_\mathcal{X} = \times_{X_i \in \mathcal{X}} D_i$ etc.

$\mathcal{X}$ is preferentially independent of $\mathcal{Y}$ (given $\mathcal{Z}$) iff

$$\text{for all } x, x' \in Dom(\mathcal{X}),\ v, v' \in Dom(\mathcal{Y}),\ w \in Dom(\mathcal{Z}),$$
$$(x, y, z) \succeq (x', y, z) \text{ if and only if } (x, y', z) \succeq (x', y', z)$$

*given a fixed value z of Z, the preferences over the possibles values of X are independent from the value of Y*

**"Safe" sequential voting** (Lang & Xia, 09)

$O : X_1 > \ldots > X_p$ order on variables

At step $i$, all voters vote on variable $X_i$, using a local voting rule $r_i$, and the outcome is communicated to the voters before variable $X_{i+1}$ is considered.

Requires the domain restriction

(R) *the preferences of every voter on* $X_i$ *are independent from the values of* $X_{i+1}, \ldots, X_n$.

+ simple elicitation protocol
+ computationally easy (provided local rules are easy to compute)
+ voters have no problem reporting their preferences, nor do they ever experience regret after the final outcome is known

– the number of profiles satisfying (R) is exponentially small; however

  + many "practical" domains comply with (R)

    `main course` > `first course` > `wine`

  + still: much weaker restriction than separability.

## **CP-nets** (Boutilier, Brafman, Hoos and Poole, 99)

Language for specifying preferences on combinatorial domains based on the notion of conditional preferential independence.

$$
\begin{tikzcd}
\mathbf{X} \arrow[r] \arrow[rr, bend left] & \mathbf{Y} \arrow[r] & \mathbf{Z}
\end{tikzcd}
$$

$$\boxed{x \succ \bar{x}} \qquad \boxed{\begin{array}{c} x : y \succ \bar{y} \\ \bar{x} : \bar{y} \succ y \end{array}} \qquad \boxed{\begin{array}{c} x \vee y : z \succ \bar{z} \\ \neg(x \vee y) : \bar{z} \succ z \end{array}}$$

**X** independent of **Y** and **Z**; **Y** independent of **Z**

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

if $X = x$

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

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

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

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

# CP-nets

$$
\begin{tikzcd}
\mathbf{X} \arrow[r] \arrow[rr, bend left] & \mathbf{Y} \arrow[r] & \mathbf{Z}
\end{tikzcd}
$$

$$\boxed{x \succ \bar{x}} \qquad \boxed{\begin{array}{c} x : y \succ \bar{y} \\ \bar{x} : \bar{y} \succ y \end{array}} \qquad \boxed{\begin{array}{c} x \vee y : z \succ \bar{z} \\ \neg(x \vee y) : \bar{z} \succ z \end{array}}$$

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

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

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

$\succ_{\mathcal{C}}$ = transitive closure of $\succ^X \cup \succ^Y \cup \succ^Z$

# CP-nets

$$
\begin{tikzcd}
\mathbf{X} \arrow[r] \arrow[rr, bend left] & \mathbf{Y} \arrow[r] & \mathbf{Z}
\end{tikzcd}
$$

$$\boxed{x \succ \bar{x}} \qquad \boxed{\begin{array}{c} x : y \succ \bar{y} \\ \bar{x} : \bar{y} \succ y \end{array}} \qquad \boxed{\begin{array}{c} x \vee y : z \succ \bar{z} \\ \neg(x \vee y) : \bar{z} \succ z \end{array}}$$

$$
\begin{tikzcd}[column sep=small, row sep=small]
 & & x\bar{y}z \arrow[dr] & & & & & \\
\succ: & xyz \arrow[ur] \arrow[dr] & & x\overline{yz} \arrow[r] & \overline{xyz} \arrow[r] & \overline{xy}z \arrow[r] & \bar{x}yz \arrow[r] & \bar{x}y\bar{z} \\
 & & xy\bar{z} \arrow[ur] & & & & &
\end{tikzcd}
$$

$$
\begin{tikzcd}
\mathbf{X} \arrow[r] \arrow[rr, bend left] & \mathbf{Y} \arrow[r] & \mathbf{Z}
\end{tikzcd}
$$

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

Example: $r_X = r_Y =$ majority rule

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

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

$\succ^X$:

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

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

Example: $r_X = r_Y =$ majority rule

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

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

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

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

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

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

Question: given some property $P$ of voting rules, do we have

$$r_1, \ldots, r_p \text{ satisfy } P \Rightarrow Seq(r_1, \ldots, r_p) \text{ satisfies } P?$$

General study in (Lang & Xia, 09); here we just give an example for *participation*

Question: given some property $P$ of voting rules, do we have

$$r_1, \ldots, r_p \text{ satisfy } P \Rightarrow Seq(r_1, \ldots, r_p) \text{ satisfies } P?$$

General study in (Xia, Lang & Ying, TARK-07) and (Lang and Xia, 09); here we just focus on three properties:

- Condorcet-consistency;
- participation
- strategyproofness

**Sequential Condorcet winner**:

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

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

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

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

$\Rightarrow xy$ sequential Condorcet winner

Properties:

1. $\vec{x}$ Condorcet winner $\Rightarrow \vec{x}$ sequential Condorcet winner
   N.B. The converse does not hold (4 voters prefer $\overline{xy}$ to $xy$).

2. if every $r_i$ is Condorcet-consistent then $Seq(r_1, \ldots, r_p)$ elects the sequential Condorcet winner when there is one (obvious)

**Corollary**: if every $r_i$ is Condorcet-consistent then $Seq(r_1, \ldots, r_p)$ is Condorcet-consistent

Counter-example for *participation*

two variables $X, Y$. $D_X = \{x_0, x_1, x_2\}$; $D_Y = \{y, \overline{y}\}$.

$r_1$ a scoring rule with score vector $(3,2,0)$, $r_2$ = majority.

$r_1$ and $r_2$ satisfy participation.

$V_1, V_2$: $x_0 y \succ x_0\overline{y} \succ x_1\overline{y} \succ x_1 y \succ x_2\overline{y} \succ x_2 y$

$$\boxed{x_0 \succ x_1 \succ x_2} \qquad \boxed{\begin{array}{l} x_0 : y \succ \overline{y} \\ x_1 : \overline{y} \succ y \\ x_2 : \overline{y} \succ y \end{array}}$$

$V_3$: $x_1 y \succ x_2 y \succ x_0 y \succ x_1\overline{y} \succ x_2\overline{y} \succ x_0\overline{y}$

$$\boxed{x_1 \succ x_2 \succ x_0} \qquad \boxed{y \succ \overline{y}}$$

$P = \{V_1, V_2\}$: $Seq(r_1, r_2)(P) = x_0 y$

$P' = \{V_1, V_2, V_3\}$: $Seq(r_1, r_2)(P') = x_1\overline{y}$

But 3 prefers $x_0 y$ to $x_1\overline{y}$.

# Manipulability

Does (non-)manipulability transfers from local rules to their sequential composition?

**Proposition** (obvious): if one of the $r_i$ is manipulable then $Seq(r_1, \ldots, r_p)$ is manipulable.

The converse is false:

**Proposition** Sequential majority for binary issues is manipulable (although the majority rule is not).

Two binary issues $X, Y$.

$V_1 : xy \succ \bar{x}y \succ x\bar{y} \succ \overline{xy}$

$V_2 : x\bar{y} \succ xy \succ \bar{x}y \succ \overline{xy}$

$V_3 : \bar{x}y \succ \overline{xy} \succ x\bar{y} > xy.$

$\{V_1, V_2, V_3\}$ is compatible with $x > y$.

If 1 votes sincerely: $Seq(maj_1, maj_2)(V_1, V_2, V_3) = x\bar{y}.$

If 2 votes for $\bar{x}$ instead of $x$: $Seq(maj_1, maj_2)(V_1', V_2, V_3) = \bar{x}y.$

$\Rightarrow$ if 1 knows the preferences of 2 and 3 then he has no interest to vote sincerely.

## Decomposability

A voting rule $r$ on $\mathcal{X} = D_1 \times \ldots \times D_p$ is **decomposable** if there exist $n$ voting rules $r_1, \ldots, r_p$ on $D_1, \ldots, D_p$ such that: for any linear order $\mathcal{O} = X_1 > \ldots > X_p$ on $\mathcal{V}$ and for any preference profile $R = (R_1, ..., R_N)$ compatible with $\mathcal{O}$, we have $Seq(r_1, \ldots, r_p)(R) = r(R)$.

**Example**: Positional scoring rules are not decomposable.

| 4 voters | 3 voters | 3 voters | 2 voters |
|---|---|---|---|
| $xy$ | $x\bar{y}$ | $\bar{x}y$ | $\bar{x}y$ |
| $\bar{x}y$ | $xy$ | $\overline{xy}$ | $\overline{xy}$ |
| $x\bar{y}$ | $\bar{x}y$ | $xy$ | $x\bar{y}$ |
| $\overline{xy}$ | $\overline{xy}$ | $x\bar{y}$ | $xy$ |

The profile follows the order $X > Y$.

Sequential winner = $xy$.

$Score(xy) = 4s_1 + 3s_2 + 3s_3 < Score(\bar{x}y) = 5s_1 + 4s_2 + 3s_3$
$\Rightarrow$ no scoring rule elects $xy$.

A voting rule $r$ on $\mathcal{X} = D_1 \times \ldots \times D_p$ is **decomposable** iff there exist $n$ voting rules $r_1, \ldots, r_p$ on $D_1, \ldots, D_p$ such that: for any linear order $\mathcal{O} = X_1 > ... > X_p$ on $\mathcal{V}$ and for any preference profile $R$ = ($R_1$, ..., $R_N$) following $\mathcal{O}$, we have $Seq(r_1, \ldots, r_p)(R) = r(R)$.

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

Obviously:

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

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

**Proposition** (Xia and Lang, 09): if $C$ is a decomposable, neutral and nondictatorial correspondence, then $C(R) = \mathcal{X}$ for all $R$.

"Safe" sequential voting: relies on a strong domain restriction:

(R) *all voters have a preference relation compatible with the order* $X_1 > \ldots > X_n$

$+$ simple elicitation protocol for these sequential voting rules

$-$ the number of profiles satisfying (R) is exponentially small

even if

$+$ many "practical" domains comply with (R)

`main course` $>$ `first course` $>$ `wine`

What can we do when (R) is not reasonable?

## Sequential voting

+ simple elicitation protocol
+ computationally easy (provided local rules are easy to compute)

- restriction-free sequential voting
  + always applicable
  - voters may feel ill at ease reporting a preference on some attributes, or experience regret after the final outcome is known
  - the outcome depends on the order in which the attributes are decided

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

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

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

Suppose voters behave optimistically, and that the chair knows that.

**S** $>$ **T**

3 votes for $S$, 2 votes for $\bar{S}$; *local outcome*: $S$

given **S** = $S$, 4 votes for $\bar{T}$, 1 vote for $T \Rightarrow \bar{T}$; *final outcome*: $S\bar{T}$

**T** $>$ **S**

3 votes for $T$, 2 votes for $\bar{T}$; *local outcome*: $T$

given **T** = $T$, 4 votes for $\bar{S}$, 1 vote for $S \Rightarrow \bar{S}$; *final outcome*: $\bar{S}T$

The chair's strategy:

- if she prefers $S\bar{T}$ to $\bar{S}T$: choose the order **S** $>$ **T**
- if she prefers $\bar{S}T$ to $S\bar{T}$: choose the order **T** $>$ **S**

Note that $ST$ and $\bar{S}\bar{T}$ cannot be obtained.

**The chair can (sometimes) control the election by fixing the agenda**

**Now: how hard is it to control the outcome by fixing the agenda?**

Several types of control:

- local/global dichotomy:
  - *global control*: the chair tries to determine the winning alternative
  - *local control*: the chair tries to determine the outcome of a single issue
- constructive/destructive dichotomy:
  - *constructive control*: the chair tries to ensure that a particular alternative or a particular value for an issue wins,
  - *destructive control*: the chair tries to ensure that a particular alternative or a particular value for an issue does *not* win.

## Formulating the problem

*Input*:

- a set of voters $\{1, \ldots, n\}$
- a set of binary issues $I = \{X_1, \ldots, X_p\}$ ($\Rightarrow$ local rules: majority)
- an outcome $\vec{x}$ (global control), or a value $x_i$ of an issue (local control), that the chair wants to obtain (or not to obtain)
- for every voter, a compact specification of her vote on each single issue, given partial assignments of some other issues: "*conditional behaviour net*"

$$\mathbf{X} \longrightarrow \mathbf{Y}$$

vote x

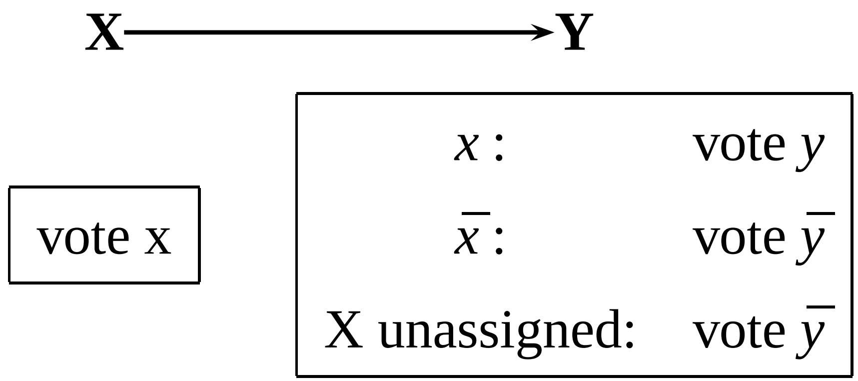
| | |
|---|---|
| $x$ : | vote $y$ |
| $\bar{x}$ : | vote $\bar{y}$ |
| X unassigned: | vote $\bar{y}$ |

CB-nets may be required to be *consistent*: if I vote $Y = y$ both when $X$ is assigned to true and when $X$ is assigned to false, then I should vote $Y = y$ when $X$ is unassigned.

## Formulating the problem

*Input*:

- a set of voters $\{1,\ldots,n\}$
- a set of binary issues $I = \{X_1,\ldots,X_p\}$ ($\Rightarrow$ local rules: majority)
- an outcome $\vec{x}$ (global control), or a value $x_i$ of an issue (local control), that the chair wants to obtain (or not to obtain)
- for every voter, a compact specification of her vote on each single issue, given partial assignments of some other issues: "*conditional behaviour net*"




The chair doesn't need to know the CB-net of every voter. It is enough for her to know the majority CB-net.

## Constructive control

- global constructive control is NP-complete.
- local constructive control is NP-complete.

Membership easy.

Hardness (both for global and local control) by reduction from the restriction of HAMILTONIAN CYCLE to graphs where each node has degree at most 3.

In both cases, NP-hardness holds even if either of the following conditions holds:

1. there is only one voter
2. all CB-nets are consistent and share the same dependency graph.

Can constructive control sometimes be easier?

- if the voters' CB-nets share the same dependency graph $G$, and every node in $G$ has at most one parent, then constructive control, both local and global, is in P.

## Destructive control

- local destructive control is NP-complete (trivially from the NP-completeness of local constructive control)
- global destructive control is NP-complete.

NP-hardness holds even if all the CB-nets are consistent.

NP-hardness proof by reduction from EXACT COVER BY 3-SETS

- if the voters' CB-nets share the same dependency graph $G$, then global destructive control is in P.

## Conclusion: the pros and cons of (restriction-free) sequential voting

+ always applicable
+ elicitation and computation easy
– voters may feel ill at ease, and may experience regret after the final outcome is known
– the chairman can control the process
+ but at the cost of lengthy computations

## Further work

Hardness shown only in the worst case; we do not yet know about whether control is "usually" hard.

*Strategic Sequential Voting* (Xia, Conitzer & Lang, 10)

Three new assumptions:

1. all voters act strategically, and this is common knowledge.
2. the order in which the issues will be voted upon, as well as the local voting rules used at the different steps, are common knowledge;
3. all voters' preferences on the set of alternatives are common knowledge.

Two binary issues $A$, $B$ with domains $\{a, \overline{a}\}$ and $\{b, \overline{b}\}$)

Three voters:

- *voter 1*: $ab \succ \overline{a}b \succ a\overline{b} \succ \overline{a}\overline{b}$;
- *voter 2*: $a\overline{b} \succ ab \succ \overline{a}b \succ \overline{a}\overline{b}$;
- *voter 3*: $\overline{a}b \succ \overline{a}\overline{b} \succ a\overline{b} \succ ab$.

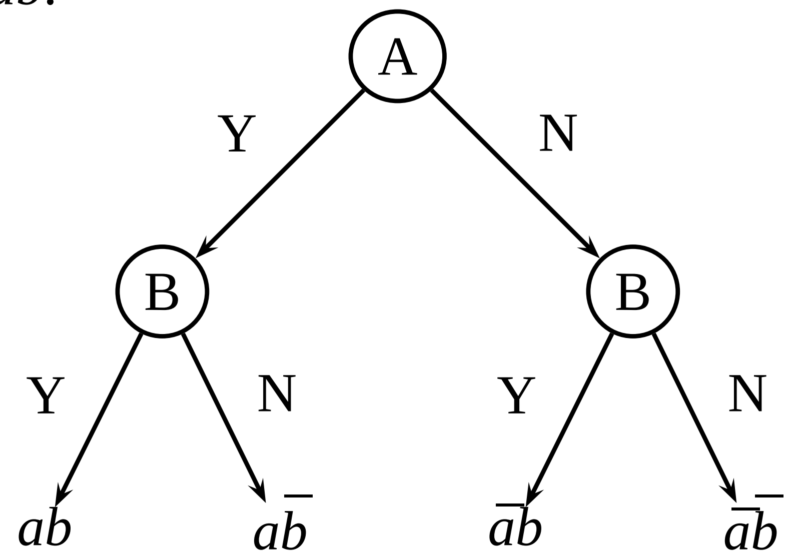
Order in which issues are decided: $A > B$.

Local rule at each step: majority.

Two binary issues $A$, $B$ with domains $\{a, \bar{a}\}$ and $\{b, \bar{b}\}$)

Three voters:

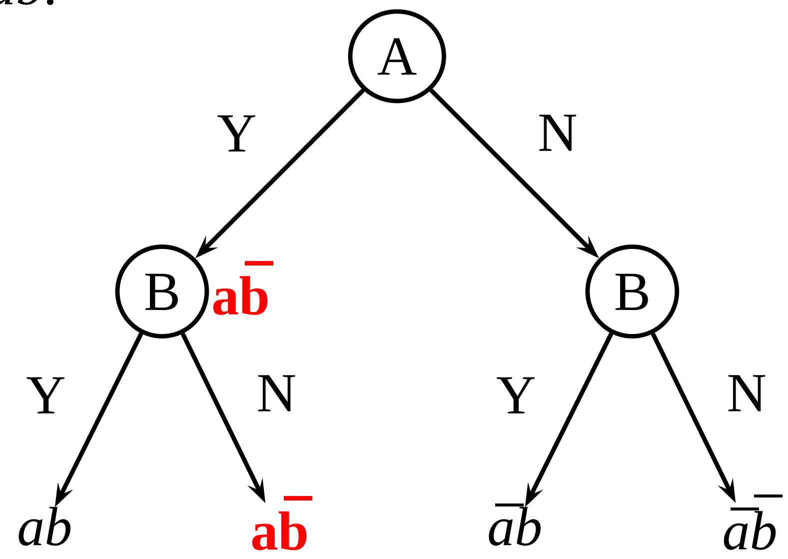
- *voter 1*: $ab \succ \bar{a}b \succ a\bar{b} \succ \bar{a}\bar{b}$;
- *voter 2*: $a\bar{b} \succ ab \succ \bar{a}b \succ \bar{a}\bar{b}$;
- *voter 3*: $\bar{a}b \succ \bar{a}\bar{b} \succ a\bar{b} \succ ab$.



Two binary issues $A$, $B$ with domains $\{a, \bar{a}\}$ and $\{b, \bar{b}\}$)

Three voters:

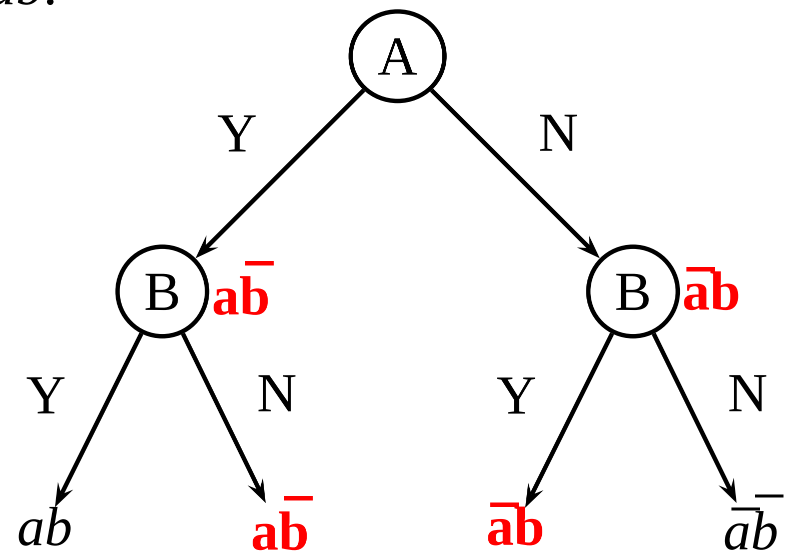
- *voter 1*: $ab \succ \bar{a}b \succ a\bar{b} \succ \bar{a}\bar{b}$;
- *voter 2*: $a\bar{b} \succ ab \succ \bar{a}b \succ \bar{a}\bar{b}$;
- *voter 3*: $\bar{a}b \succ \bar{a}\bar{b} \succ a\bar{b} \succ ab$.



Two binary issues $A$, $B$ with domains $\{a, \bar{a}\}$ and $\{b, \bar{b}\}$)

Three voters:

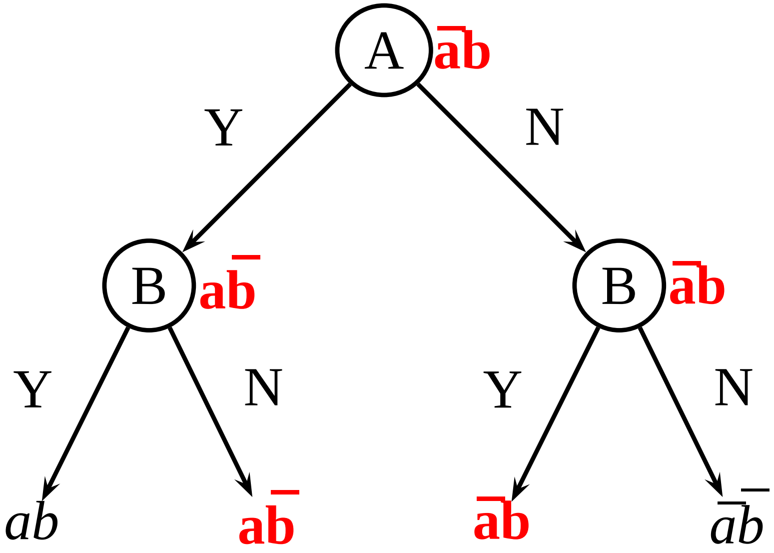
- *voter 1*: $ab \succ \bar{a}b \succ a\bar{b} \succ \bar{a}\bar{b}$;
- *voter 2*: $a\bar{b} \succ ab \succ \bar{a}b \succ \bar{a}\bar{b}$;
- *voter 3*: $\bar{a}b \succ \bar{a}\bar{b} \succ a\bar{b} \succ ab$.



Two binary issues $A$, $B$ with domains $\{a, \bar{a}\}$ and $\{b, \bar{b}\}$)

Three voters:

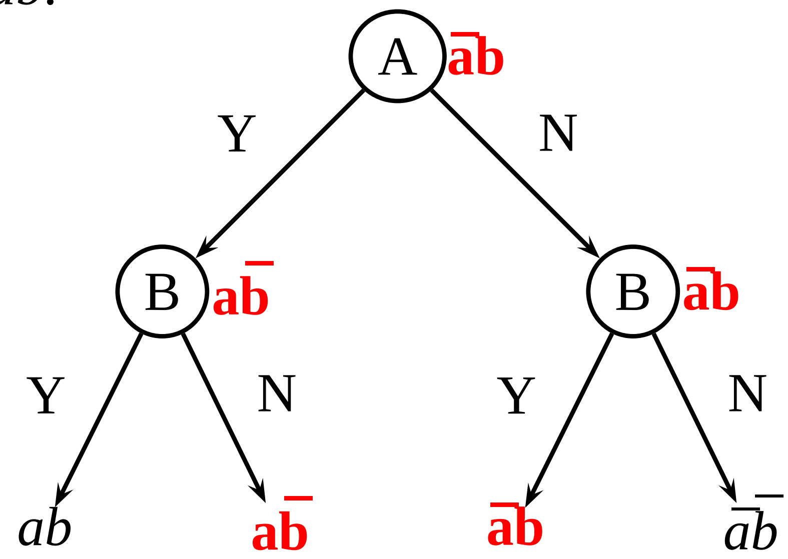
- *voter 1*: $ab \succ \bar{a}b \succ a\bar{b} \succ \bar{a}\bar{b}$;
- *voter 2*: $a\bar{b} \succ ab \succ \bar{a}b \succ \bar{a}\bar{b}$;
- *voter 3*: $\bar{a}b \succ \bar{a}\bar{b} \succ a\bar{b} \succ ab$.



Two binary issues $A$, $B$ with domains $\{a, \bar{a}\}$ and $\{b, \bar{b}\}$)

Three voters:

- *voter 1*: $ab \succ \bar{a}b \succ a\bar{b} \succ \bar{a}\bar{b}$;
- *voter 2*: $a\bar{b} \succ ab \succ \bar{a}b \succ \bar{a}\bar{b}$;
- *voter 3*: $\bar{a}b \succ \bar{a}\bar{b} \succ a\bar{b} \succ ab$.




Voter 1's preferences are *separable*: she prefers $a$ to $\bar{a}$ whatever the value of $B$, and $b$ to $\bar{b}$ whatever the value of $A$.

And yet she strategically votes for $\bar{a}$

Two interpretations:

- modeling sequential voting as a complete-information game, which allows us to analyze sequential voting in multi-issue domains from a game-theoretic point of view;
- a new family of voting rules on multi-issue domains (a distinguished subset of the family of voting trees).

[$O = X_1 > \ldots > X_n$ be the order in which the issues are decided

Question 1: in which situations is it in the voters' interest to vote truthfully at every stage?

Answer: yes when

- when every voter has $O$-lexicographic preferences
- when for every $i$, the preferences of every voter on the values of $X_i$ depend only on the values of $X_{i+1}, \ldots, X_p$ (and *a fortiori*, when every voter has separable preferences).

Question 2: do multiple election paradoxes arise?

Answer: unfortunately yes

*For any* $p \in \mathbb{N}$ *and any* $n \geq 2p^2 + 1$, *there exists a profile P such that the outcome is ranked among the* $\lfloor p/2 + 2 \rfloor$ *worst alternatives, and is Pareto-dominated by* $2^p - (p+1)p/2$ *alternatives.*

**How should such a vote be conducted?**

Some classes of solutions:

1. don't bother and vote separately on each variable.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. limit the number of possible alternatives that voters may vote for.
4. 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.
5. ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.
6. *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.
7. **use a *compact preference representation language* in which the voters' preferences are represented in a concise way.**

7. **use a *compact preference representation language* in which the voters' preferences are represented in a concise way.**

+ no domain restriction, provided the language is totally expressive.

– potentially expensive in elicitation and/or computation: computing the winner is generally NP-hard or coNP-hard.

Examples of such approaches:

- using GAI-nets: (Gonzalès & Perny, 08);
- using CP-nets: (Xia, Conitzer & Lang, 08);
- using weighted logical formulae: (Uckelman, 09).

Any preference relation on a combinatorial domain is compatible with some CP-net (possibly with cyclic dependencies).

Elicit the CP-net corresponding to each voter and aggregate “locally”.

*Example 1* (swimming pool): 5 voters, 2 binary variables $S$, $T$

| 2 voters | 2 voters | 1 voter |
|---|---|---|
| $S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T} \succ ST$ | $\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$ | $ST \succ \bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T}$ |

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

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

$$\mathbf{S} \qquad \mathbf{T} \qquad \boxed{S \succ \bar{S}} \quad \boxed{T \succ \bar{T}}$$

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

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

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

*Example 2*: 3 voters, 2 binary variables $A$, $B$

$$\mathbf{A} \longrightarrow \mathbf{B}$$

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

$$\mathbf{A} \longleftarrow \mathbf{B}$$

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

$$\mathbf{A} \qquad \mathbf{B}$$

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

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

$$\mathbf{A} \longleftrightarrow \mathbf{B}$$

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

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

+ always applicable

– elicitation cost: in the worst case, exponential number of queries to each voter

– computation cost: dominance in CP-nets with cyclic dependencies is PSPACE-complete

– there might be no winner; there might be several winners

**How should such a vote be conducted?**

Some classes of solutions:

1. don't bother and vote separately on each variable.
2. ask voters to specify their preference relation by ranking all alternatives *explicitly*.
3. limit the number of possible alternatives that voters may vote for.
4. 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.
5. ask voters their preferred alternative(s) and complete them automatically using a predefined *distance*.
6. *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.
7. use a *compact preference representation language* in which the voters' preferences are represented in a concise way.

Conclusion: *either impose a strong domain restriction, or pay a high communication and computational cost*