Unravelling multi-agent ranked delegations

Arianna Novaro

UNIVERSITÉ PARIS 1
PANTHÉON SORBONNE

joint work with: Rachael Colley and Umberto Grandi
(IRIT, University of Toulouse)

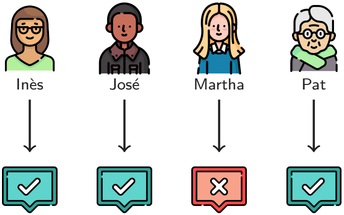
OSGAD seminar · November 24, 2021

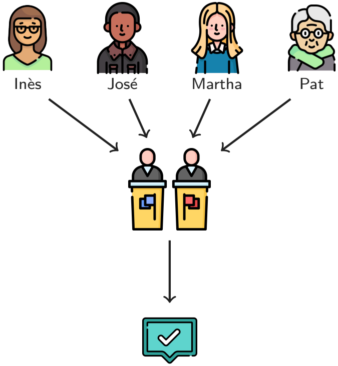
Democracy: direct vs. representative

A group of agents has to take a *collective decision* on some issues.

Democracy: direct vs. representative

A group of agents has to take a *collective decision* on some issues.



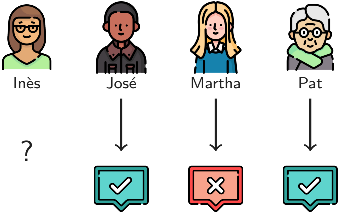


Direct democracy
Each agent votes on each issue.

Representative democracy
Agents elect some representatives.

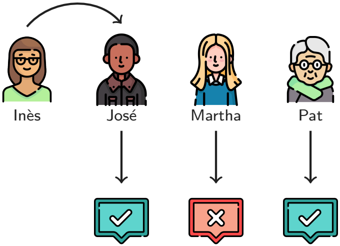
Delegative (liquid) democracy

For each issue, agents can either vote directly or delegate.



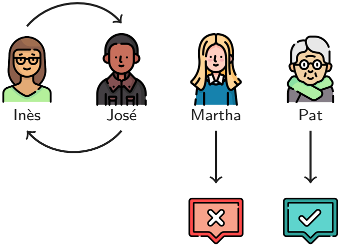
Delegative (liquid) democracy

For each issue, agents can either vote directly or delegate.



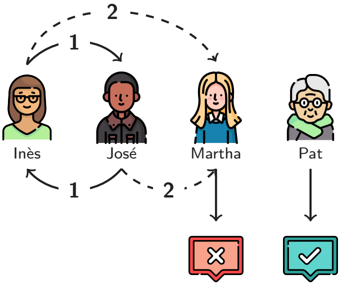
Delegation cycles in liquid democracy

What to do in case of *delegation cycles*?



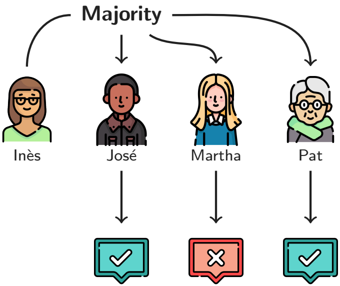
Ranked delegations

Agents can specify a *ranking* of preferred delegates.



Multi-agent delegations

Agents can specify *complex delegations* to multiple agents.



Talk overview

- A model for multi-agent ranked delegations;
- Different unravelling procedures for delegation profiles;
- Delegation profiles restricted to specific languages;
- Study computational and axiomatic properties for procedures.

The overall process

1. Agents write their ballots (with votes and ranked delegations);
2. We check that the ballots are valid;
3. An unravelling procedure transforms delegations into votes;
4. A decision is taken with a voting rule.

R. Colley, U. Grandi, A. Novaro. Smart Voting. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence* (IJCAI-2020).

The model

- A finite set of n agents $\mathcal{N}$ decide on issue i with domain $D(i)$.

$\Rightarrow$ For simplicity we focus on a *single* issue—thus drop the i everywhere.

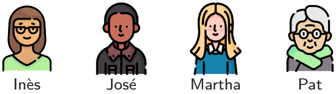
- An agent's ballot is an ordering $((S^1, F^1) > \cdots > (S^k, F^k) > x)$ where each S^h is a set of agents, F^h is a resolute function, and $x \in D$ is a back-up vote.

- A valid ballot B for agent a is such that for all $h, \ell \leq k$ *(i)* if $S^h \cap S^\ell \neq \emptyset$ then F^h and F^ℓ are not equivalent; *(ii)* $a \notin S^h$.

- A profile $\boldsymbol{B} = (B_1, \ldots, B_n)$ is a vector of agents' ballots.

Examples of delegation ballots

Shall we try a new take-away restaurant (YES) *or cook at home* (NO)*?*



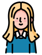
Some possible valid ballots for Inès (domain of F not shown):

- **Boolean formulas:** (José $\wedge$ (Martha $\vee$ Pat)) $>$ José $>$ YES.
- **Ranked single-agent:** José $>$ Martha $>$ Pat $>$ YES.
- **Quota rules:** $Majority$(José, Martha, Pat) $>$ YES.

Unravelling procedures

 : (José $\wedge$ (Martha $\vee$ Pat)) $>$ José $>$ YES

 : YES

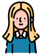
 : $Majority$(José, Inès, Pat) $>$ NO

 : José $>$ Inès $>$ NO

$\rightarrow$

 : YES

 : YES

 : YES

 : NO

An unravelling procedure $\mathcal{U}$ for issue i and agents in $\mathcal{N}$ is a function:

$$\mathcal{U} : (B_1 \times \cdots \times B_n) \rightarrow D^n.$$

Certificate

- A certificate $\boldsymbol{c}$ for a profile $\boldsymbol{B}$ is a vector of n entries specifying a preference level for each agent $a \in \mathcal{N}$.
- A certificate $\boldsymbol{c}$ is consistent if there is an ordering σ of the agents which allows to iteratively construct an outcome $X \in \mathcal{D}^n$, using the values in $\boldsymbol{c}$ and the votes computed so far.

Certificate

- A certificate $\boldsymbol{c}$ for a profile $\boldsymbol{B}$ is a vector of n entries specifying a preference level for each agent $a \in \mathcal{N}$.
- A certificate $\boldsymbol{c}$ is consistent if there is an ordering σ of the agents which allows to iteratively construct an outcome $X \in \mathcal{D}^n$, using the values in $\boldsymbol{c}$ and the votes computed so far.

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge C$	D	1
B	1	–	–
C	D	0	–
D	A	B	0

$\boldsymbol{c} = (1, 1, 1, 1)$ is not consistent, while $\boldsymbol{c'} = (3, 1, 1, 1)$ is.

Two optimal procedures

MinSum: Minimize the sum of the preference levels used.

$$\textsc{MinSum}(\boldsymbol{B}) = \{X_{\boldsymbol{c}} \mid \boldsymbol{c} \in \arg\min_{\boldsymbol{c} \in \mathcal{C}(\boldsymbol{B})} \sum_{a \in \mathcal{N}} \boldsymbol{c}_a\}$$

MinMax: Minimize the rank of the worst-off agent.

$$\textsc{MinMax}(\boldsymbol{B}) = \{X_{\boldsymbol{c}} \mid \boldsymbol{c} \in \arg\min_{\boldsymbol{c} \in \mathcal{C}(\boldsymbol{B})} \max(\boldsymbol{c})\}$$

Two optimal procedures

MinSum: Minimize the sum of the preference levels used.

$$\textsc{MinSum}(\boldsymbol{B}) = \{X_{\boldsymbol{c}} \mid \boldsymbol{c} \in \arg\min_{\boldsymbol{c} \in \mathcal{C}(\boldsymbol{B})} \sum_{a \in \mathcal{N}} \boldsymbol{c}_a\}$$

MinMax: Minimize the rank of the worst-off agent.

$$\textsc{MinMax}(\boldsymbol{B}) = \{X_{\boldsymbol{c}} \mid \boldsymbol{c} \in \arg\min_{\boldsymbol{c} \in \mathcal{C}(\boldsymbol{B})} \max(\boldsymbol{c})\}$$

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge C$	D	1
B	1	$-$	$-$
C	D	0	$-$
D	A	B	0

Four greedy procedures

Four variants of a greedy unravelling following two criteria:

(D) *Direct vote priority*: priority given to direct votes (or possibly the backups) over computable delegations.

(R) *Random voter selection*: randomly choose only one agent at a time, whose (computable or backup) vote is added.

This gives us the UNRAVEL procedures: **U**, **DU**, **RU** and **DRU**.

Example of UNRAVEL(**U**)

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge \boldsymbol{C}$	D	1
B	1	–	–
C	$\boldsymbol{D}$	0	–
D	$\boldsymbol{A}$	B	0

We cannot compute A's delegation, since we need C's vote, which depends on D, which depends on A.

Example of UNRAVEL(**U**)

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge C$	D	1
B	①	–	–
C	D	0	–
D	A	B	0

- Take the direct vote of B, first preference. $X = (\Delta, 1, \Delta, \Delta)$.

Example of UNRAVEL(**U**)

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge C$	D	1
B	1	–	–
C	D	0	–
D	A	B	0

- Take the direct vote of B, first preference. $X = (\Delta, 1, \Delta, \Delta)$.
- Cannot add anything at first preference level: move to second.

Example of UNRAVEL(**U**)

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge C$	D	1
B	1	–	–
C	D	(0)	–
D	A	(B)	0

- Add backup vote of C, second preference. $X = (\Delta, 1, 0, \Delta)$.
- Add D's delegation to B, second preference. $X = (\Delta, 1, 0, 1)$.

Example of $\textsc{Unravel}(\mathbf{U})$

	1^{st}	2^{nd}	3^{rd}
A	$B \wedge C$	D	1
B	1	–	–
C	D	0	–
D	A	B	0

- Add A's delegation to $B \wedge C$, first preference. $X = (0, 1, 0, 1)$.
- Result $\textsc{Unravel}(\mathbf{U}) = (0, 1, 0, 1)$, certificate $\boldsymbol{c} = (1, 1, 2, 2)$.

Language restrictions

We can impose some language restrictions on the agents' ballots:

- LIQUID: language of ranked single-agent delegations.
- BOOL: language of (contingent) propositional formulas expressed as complete DNFs.
- $\mathcal{L}[k]$: language $\mathcal{L}$ where voters express at most k delegations.

Algorithmic analysis

Theorem. The algorithms of the four greedy UNRAVEL procedures always terminate on valid profiles.

Theorem. The four greedy UNRAVEL procedures and MINSUM give the same outcome X LIQUID$[1]_*$ ballots (but the certificate may differ).

Complexity · Optimal procedures

Given a profile $\boldsymbol{B}$ of $\textsc{Bool}$ ballots and $M \in \mathbb{N}$, is there a certificate $\boldsymbol{c}$ that unravels $\boldsymbol{B}$ such that $\sum_{a \in \mathcal{N}} \boldsymbol{c}_a \leq M$?

Theorem. $\textsc{BoundedMinSum}$ is NP-complete.

Given a profile $\boldsymbol{B}$ of $\textsc{Bool}$ ballots and $M \in \mathbb{N}$, is there a certificate $\boldsymbol{c}$ that unravels $\boldsymbol{B}$ such that $\max(\boldsymbol{c}) \leq M$?

Theorem. $\textsc{BoundedMinMax}$ is NP-complete.

Complexity · Greedy procedures

Theorem. Unravelling a Bool profile via the Unravel procedures takes at most $\mathcal{O}(n^2 \cdot \max_p(\boldsymbol{B}) \cdot \max_\varphi(\boldsymbol{B}))$ time.

- $\max_p(\boldsymbol{B})$: highest preference level of any ballot in $\boldsymbol{B}$.
- $\max_\varphi(\boldsymbol{B})$: maximum length of any formula in $\boldsymbol{B}$.

Participation results for LIQUID$_*$

- Cast-participation: a direct voter is always better off by voting directly, rather than expressing any other ballot.
- Guru-participation: a direct voter always benefits from receiving delegations from other agents.

Theorem. Any monotonic rule, paired with UNRAVEL(U) or UNRAVEL(DU), satisfies cast-participation for LIQUID$_*$.

Theorem. Relative majority, with any of the four UNRAVEL procedures, does not satisfy guru-participation for LIQUID$_*$.

Pareto optimality of MinSum

A certificate $\boldsymbol{c}$ Pareto dominates a certificate $\boldsymbol{c'}$ if for every $i \in \mathcal{N}$ we have $\boldsymbol{c}_i \leq \boldsymbol{c}'_i$ and there is some $j \in \mathcal{N}$ such that $\boldsymbol{c}_j < \boldsymbol{c}'_j$.

A certificate $\boldsymbol{c}$ for profile $\boldsymbol{B}$ is Pareto optimal for the consistent certificates $\mathcal{C}(\boldsymbol{B})$ if there exists no $\boldsymbol{c} \in \mathcal{C}(\boldsymbol{B})$ with $\boldsymbol{c} \neq \boldsymbol{c'}$, such that $\boldsymbol{c'}$ Pareto dominates $\boldsymbol{c}$.

Theorem. The certificate $\boldsymbol{c}$ for any outcome $X_{\boldsymbol{c}} \in \textsc{MinSum}(\boldsymbol{B})$ is Pareto optimal for $\mathcal{C}(\boldsymbol{B})$, for any profile $\boldsymbol{B}$.

Conclusions

A setting for multi-agent ranked delegations, with these key points:

- agents give a rank of complex delegations + a backup vote:
  - the delegation language can be restricted (Liquid and Bool).
- unravelling procedures turn delegations into standard votes:
  - two optimal procedures MinSum and MinMax.
  - four greedy procedures (**U**, **DU**, **RU**, **DRU**).
- computational and axiomatic properties for these procedures:
  - results on restricted languages (expressivity vs. complexity).

Credits to Freepik @ `flaticon.com`