Multiagent Ranked Delegations in Liquid Democracy

Umberto Grandi

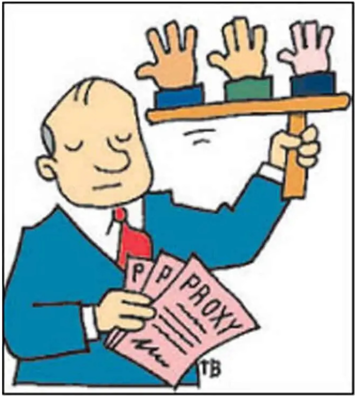
IRIT – University of Toulouse

12 January 2020

Joint work with Rachael Colley and Arianna Novaro

Delegations in voting - proxies

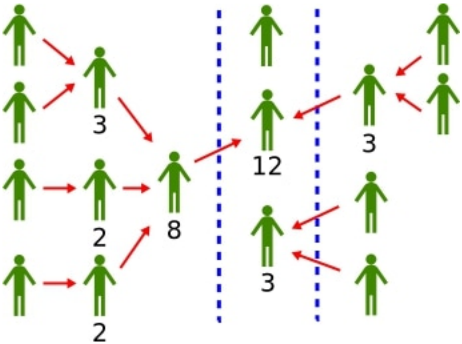
Delegating one's voting power to a proxy is common in shareholders' meetings:



Classical studies are Miller (1969) and Tullock (1992).

Delegations in voting - liquid

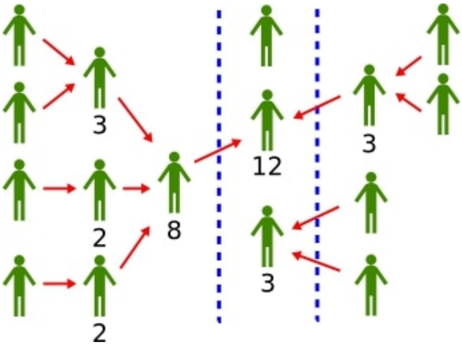
Liquid democracy allows a proxy to delegate her voting power and the delegated voting power received to another voter. Delegations becomes transitive:



By Ilmari Karonen CC BY-SA 3.0 https://commons.wikimedia.org/w/index.php?curid=23953030

Delegations in voting - liquid

Liquid democracy allows a proxy to delegate her voting power and the delegated voting power received to another voter. Delegations becomes transitive:



The term liquid actually comes from the fact that delegations can be withdrawn at any time during deliberation.

First implementation: `https://liquidfeedback.org/`

Intermezzo I: preference aggregation

Voting rules can be used to aggregate the preferences or tastes of a set of individuals over a set of alternatives:

How to decide which rule to use? Typically by checking its axiomatic properties, such as unanimity, resistance to clones, Condorcet consistency...but also its computational ones.

Intermezzo II: reconstructing the truth

The same voting rules can be used to track a ground truth starting noisy estimates by a set of individuals.

The classical result is Condorcet's jury theorem:

- two alternatives c and $\bar{c}$, with c the correct one
- each voter has an independent probability p to guess the correct alternative
- if $p > 1/2$ the probability that the majority vote is the correct alternative tends to 1 increasing the size of the electorate

We can also say that the majority rule is the maximum likelyhood estimator for the noise model described above.

Liquid democracy

Liquid democracy has been studied under both perspectives.

1. As a maximum likelihood estimator by Green-Armytage (2015), Kahng et al (2018), and Cohensius et al, 2017.
   No definitive answer: unless modified substantially liquid democracy does not seem to be a better estimator than voting without delegations.

Liquid democracy

Liquid democracy has been studied under both perspectives.

1. As a maximum likelihood estimator by Green-Armytage (2015), Kahng et al (2018), and Cohensius et al, 2017.
   No definitive answer: unless modified substantially liquid democracy does not seem to be a better estimator than voting without delegations.

2. As a preference aggregator:

   - How much power should be given to the delegates?
     (Boldi et al 2009, Kotsialou and Riley, 2020, Gölz et al 2018)
   - How to delegate on parts of a voter ballot, when voting on more than one issue or with preferences over alternatives?
     (Grossi and Christoff 2017, Brill and Talmon 2018)
   - Analyse the delegation game defined by expressing rankings over delegates or when voters have types unknown to them.
     (Bloembergen et al 2019, Escoffier et al 2020)

Time permitting quick examples of the three last work.

My encounter with liquid democracy

- A reading group on e-democracy and interactive democracy in Toulouse in 2016: `www.irit.fr/~Umberto.Grandi/teaching/directdem/`
- Markus Brill forwarded me "The principles of Liquid Feedback": `https://principles.liquidfeedback.org/`

Lots of discussions with Markus and Davide Grossi. At the time I was very skeptical about going towards more direct democracy given the experience of the "5 star movement" in Italy (they are big fan of liquid feedback but use a direct democracy platform called Rousseau, which is not open. They were the first party in the last 2018 election with around 30% of votes).

- At the industry day in COMSOC 2016 we invited a member of the Iterakive Demokratie foundation. Workshop on eDemocracy in 2017: `www.irit.fr/~Umberto.Grandi/e-democracy2017/`.
- Meeting with the creators of Liquid Feedback in Berlin with Markus Brill. Very interesting, but their foundation is based on volunteering. Unclear if they want their system to be used again in politics.
- The rest you hear in this talk, mostly ongoing PhD of Rachael Colley

Motivation I: elicit social influence

Research in opinion diffusion and social influence starts from a given network and a given model of social influence:

- Threshold models, independent cascades, deGroot model...
- In recent work we studied the diffusion of constrained opinions and preferences, using aggregation functions to model social influence. For an introduction see the second part of the IJCAI-2020 tutorial: `https://sites.google.com/view/opinionaggregationmas/home`

Question

How to elicit the social influence structure from voters (both network and arbitrary influence functions)?

Liquid democracy can be seen as a social influence elicitor: the result is a network where social influence copies the opinion of the delegate.

Motivation II: solve cycles

In the Liquid Feedback implementation cycles of influence are ignored:

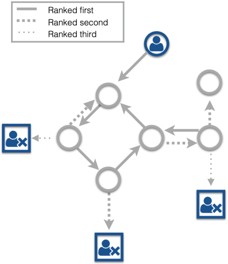


Figure by Markus Brill

- Problem first identified by Grossi and Christoff (2017), cycles result in complete abstentions, propose back-up votes to solve the problem.
- Kotsialou and Riley (2020) use ranked delegations. Two ways of breaking cycles of first delegations: breadth-first, depth-first.

Question

Are there other *polynomial procedures to break cycles* *with ranked delegations and back-up votes? How to assess them as preference aggregators?*

Motivation III: smart contracts

Electronic institutions can be developed and run using distributed ledgers and smart contracts (= self-executable code). Elections are one such example:

- Dhillon et al. (2019) give a detailed plan of the infrastructure required for a decentralised online voting platform, detailing strengths and weaknesses
- Kotsialou et al (2020) observe that during the COVID-19 pandemic having a reliable system of electronic distance voting would have been helpful!

Question

Expressing delegations in voting should be left on the user side (the voter). What are general languages for smart ballot expressions?

Two possible implementations in mind: decentralised for full transparency and accountability (issues with vote secrecy and buying), or with a central trusted authority (interesting computational question of providing certificates)

Overview

I will now present our setting for multiagent ranked delegations in voting:

1. Basic definitions of (valid) smart ballots
2. Unravelling procedures: optimal and greedy
3. Examples of greedy unravelling
4. Algorithmic, complexity, and (preliminary) axiomatic results

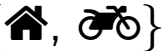
Rachael Colley, Umberto Grandi, and Arianna Novaro. Smart Voting. In *Proceeding of the the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.
+ working paper with optimal procedures and complexity results under preparation.

The overall vision of smart voting

1. Agents create and send their smart ballots, containing votes and ranked multiagent delegations;
2. We check if the ballots are valid, i.e. that they abide by the rules of the election;
3. The smart ballots are turned into a “standard” voting profile by an unravelling procedure;
4. A standard voting rule gives the collective decision (e.g. plurality, majority).

For the record, a webpage with the same name already exists using Ardor blockchain (not developed by us, not necessarily using delegations): `https://smartvoting.net/`

Formal model

- $\mathcal{N}$, a set of n agents.
  Running example: $\mathcal{N} = \{$  $\}$
- $\mathcal{I}$, a set of m issues.
  Running example: $\mathcal{I} = \{$  $\}$
- for each $i \in \mathcal{I}$, there is a domain of alternatives for this issue i, $D(i)$.
  Running example: $D($  $) = \{$  $\}$

Smart ballots

Smart ballot, $B_{🧍, 🍴}$

Smart ballots are linear orders consisting of:

- *Ranked delegations, each consisting of a set of agents,* $\{$🧍, 🧍, 🧍$\}$ *and an aggregation function,* $F(\{$🧍, 🧍, 🧍$\})$.
- *Final back-up direct vote:* 🏍 $\in D($🍴$)$

Smart ballots

Smart ballot, B🧍, 🍴

Smart ballots are linear orders consisting of:

- *Ranked delegations, each consisting of a set of agents,* {🧍, 🧍, 🧍} *and an aggregation function,* F({🧍, 🧍, 🧍}).
- *Final back-up direct vote:* 🏍 $\in D($🍴$)$

Valid smart ballot

Basic requirements:

i *No repetitions in the ranking: no two aggregations with same domain and same function.*

ii *An agent cannot delegate to themselves.*

Smart ballot example

🧍(purple), 🧍(dark blue), 🧍(light blue) and 🧍(orange), wonder whether to try a new takeaway restaurant (🏍) or to cook at home (🏠).

$\mathcal{N} = \{$🧍(purple), 🧍(dark blue), 🧍(light blue), 🧍(orange)$\}$, $\mathcal{I} = \{$🍴$\}$ and $Dom($🍴$) = \{$🏠, 🏍$\}$.

Here are some possible valid smart ballots that 🧍(purple) could give (omitting domains):

- Boolean function: $B_{\text{🧍(purple)}, \text{🍴}} =$ 🧍(dark blue) $\wedge$ (🧍(light blue) $\vee$ 🧍(orange)) $>$ 🧍(dark blue) $>$ 🏍.

Smart ballot example

[purple person], [dark blue person], [light blue person] and [orange person], wonder whether to try a new takeaway restaurant ([motorcycle]) or to cook at home ([house]).

$\mathcal{N} = \{$[purple person], [dark blue person], [light blue person], [orange person]$\}$, $\mathcal{I} = \{$[cutlery]$\}$ and $Dom($[cutlery]$) = \{$[house], [motorcycle]$\}$.

Here are some possible valid smart ballots that [purple person] could give (omitting domains):

- Boolean function: $B_{\text{[purple person]},\,\text{[cutlery]}} =$ [dark blue person] $\wedge$ ([light blue person] $\vee$ [orange person]) $>$ [dark blue person] $>$ [motorcycle].
- Ranked single-agent: $B_{\text{[purple person]},\,\text{[cutlery]}} =$ [dark blue person] $>$ [light blue person] $>$ [orange person] $>$ [motorcycle].

Smart ballot example

🧍(pink), 🧍(navy), 🧍(light blue) and 🧍(orange), wonder whether to try a new takeaway restaurant (🏍) or to cook at home (🏠).

$\mathcal{N} = \{$🧍(pink), 🧍(navy), 🧍(light blue), 🧍(orange)$\}$, $\mathcal{I} = \{$🍴$\}$ and $Dom($🍴$) = \{$🏠, 🏍$\}$.

Here are some possible valid smart ballots that 🧍(pink) could give (omitting domains):

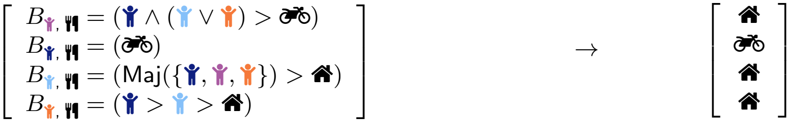
- Boolean function: $B_{🧍(pink), 🍴} =$ 🧍(navy) $\wedge$ (🧍(light blue) $\vee$ 🧍(orange)) $>$ 🧍(navy) $>$ 🏍.
- Ranked single-agent: $B_{🧍(pink), 🍴} =$ 🧍(navy) $>$ 🧍(light blue) $>$ 🧍(orange) $>$ 🏍.
- Thresholds: $B_{🧍(pink), 🍴} = \text{Maj}(\{$🧍(navy), 🧍(light blue), 🧍(orange)$\}) >$ 🏍.

Unravelling procedures

Definition: Unravelling procedure

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

$$\mathcal{U} : (B_{1i} \times \cdots \times B_{ni}) \to D(i)^n.$$



Two optimal unravelling procedures

A consistent certificate $\mathbf{c}$ for profile $\boldsymbol{B}$ is a vector of delegation ranks for the voters not leading to cycles. Example: certificate $(1, 3, 1)$ implies that the first and last voters use their first ranked delegation or direct vote, and the second voter uses her third ranked delegation or direct vote (cycles to be checked).

Definition - OPT

The optimal unravelling minimises the sum of the ranks of the delegations used:

$$\mathit{Opt}(\boldsymbol{B}_i) := \{X_{\mathbf{c}} \mid \mathbf{c} \in \arg\min_{\mathbf{c} \in \mathcal{C}(\boldsymbol{B}_i)} (\mathit{rank}(\mathbf{c}))\}.$$

Definition - MINMAX

The minmax unravelling minimises the rank of the worst-off delegation:

$$\mathit{MinMax}(\boldsymbol{B}_i) := \{X_{\mathbf{c}} \mid \mathbf{c} \in \arg\min_{\mathbf{c} \in \mathcal{C}(\boldsymbol{B}_i)} \max(\mathbf{c})\}.$$

Four greedy unravelling procedures

We define four iterative unravelling procedures following two criteria:

(D) Direct vote priority: prioritise the expression of direct votes (eventually back-ups) over delegations

(R) Random voter selection: add only one voter delegation or direct vote at the time

The resulting rules:

	Without (D)	**With (D)**
Without (R)	U	DU
With (R)	RU	DRU

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

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

There exists a delegation cycle in the agents' first preferences:
$A \to C \to D \to A$

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

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

$\texttt{Unravel}(\mathbf{U})$

Step 1: $X = (\Delta, \Delta, \Delta, \Delta)$

We take the direct vote from agent B.

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

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

$\texttt{Unravel}(\mathbf{U})$

Step 2: $X = (\Delta, 1, \Delta, \Delta)$

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

Unravel(**U**)

Step 2: $X = (\Delta, 1, \Delta, \Delta)$

- No votes can be found at the first preference level.
- At the second preference level, we can add the direct vote of agent C, as well as D's delegation to to B.

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

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

`Unravel`(**U**)

Step 3: $X = (\Delta, 1, 0, 1)$

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

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

$\texttt{Unravel}(\mathbf{U})$

Step 3: $X = (\Delta, 1, 0, 1)$

- Add A's first preference delegation, where $B \wedge C$ evaluates to 0.

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

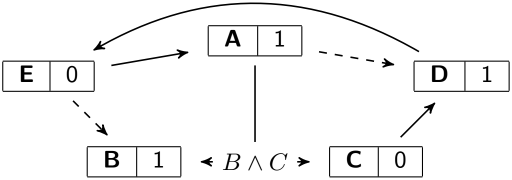
	1st	2nd	3rd
A	$B \wedge C$	D	1
B	1	-	-
C	D	0	-
D	A	B	0

$\texttt{Unravel}(\mathbf{U})$

$\texttt{Unravel}(\mathbf{U}) = (0, 1, 0, 1)$

Example of $\texttt{Unravel}(\mathbf{RU})$

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



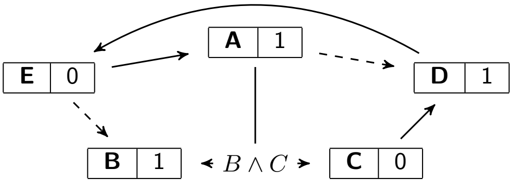
$\texttt{Unravel}(\textbf{RU})$

Step 1: $X = (\Delta, \Delta, \Delta, \Delta, \Delta)$

We take the direct vote from agent B.

Example of $\texttt{Unravel}(\mathbf{RU})$

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

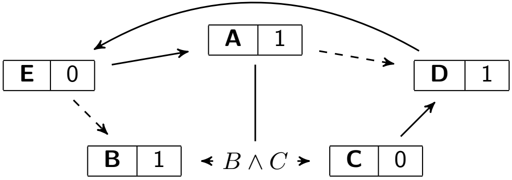


$\texttt{Unravel}(\mathbf{RU})$

Step 2: $X = (\Delta, 1, \Delta, \Delta, \Delta)$

Example of Unravel($\mathbf{RU}$)

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



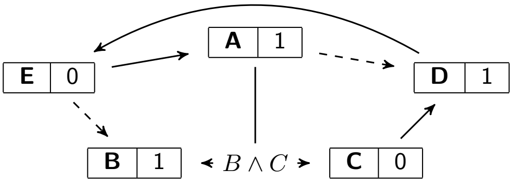
Unravel(**RU**)

Step 2: $X = (\Delta, 1, \Delta, \Delta, \Delta)$

- As there are no votes can be added from the first preference, we can add either C, D or E's vote.

Example of Unravel(RU)

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



Unravel(**RU**)

Step 2: $X = (\Delta, 1, \Delta, \Delta, \Delta)$

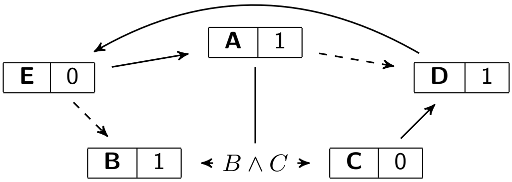
- As there are no votes can be added from the first preference, we can add either C, D or E's vote.

Choosing C

Unravel(**RU**) =
$(0, 1, 0, 0, 0)$

Example of $\texttt{Unravel}(\mathbf{RU})$

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



$\texttt{Unravel}(\mathbf{RU})$

Step 2: $X = (\Delta, 1, \Delta, \Delta, \Delta)$

- As there are no votes can be added from the first preference, we can add either C, D or E's vote.

Choosing C

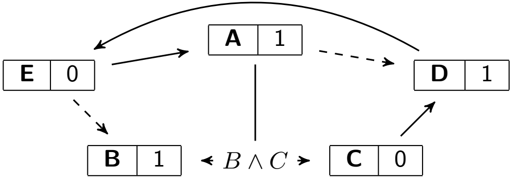
$\texttt{Unravel}(\mathbf{RU}) = (0, 1, 0, 0, 0)$

Choosing D

$\texttt{Unravel}(\mathbf{RU}) = (1, 1, 1, 1, 1)$

Example of Unravel(**RU**)

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



Unravel(**RU**)

Step 2: $X = (\Delta, 1, \Delta, \Delta, \Delta)$

- As there are no votes can be added from the first preference, we can add either C, D or E's vote.

Choosing C
Unravel(**RU**) $=$
$(0, 1, 0, 0, 0)$

Choosing D
Unravel(**RU**) $=$
$(1, 1, 1, 1, 1)$

Choosing E
Unravel(**RU**) $=$
$(1, 1, 1, 1, 1)$

Results

We focused on three types of **results**:

- **Algorithmic analysis** of unravelling procedures:
  Do the procedures always terminate?
  Are the procedures different?

- **Computational complexity** problems:
  How hard is it to check that a ballot is valid?
  In how many steps a procedure terminates?

- **Comparison** with liquid democracy (with multiple delegations):
  What is the relation with "classical" liquid democracy?
  Are agents better off by participating (or receiving delegations)?

Theorem - Different outcomes

There exists a valid smart profile B for which the four greedy unravelling procedures and the two optimal procedures give different outcomes.

Also, the breadth-first and depth-first procedures by Kotsialou and Riley are different from the ones we propose, mainly in their treatment of abstentions.

Algorithmic analysis

Theorem - Different outcomes

There exists a valid smart profile B for which the four greedy unravelling procedures and the two optimal procedures give different outcomes.

Also, the breadth-first and depth-first procedures by Kotsialou and Riley are different from the ones we propose, mainly in their treatment of abstentions.

Theorem - Termination of greedy unravelling

The algorithms of the four greedy unravelling procedures always terminate on valid smart profiles.

Algorithmic analysis

Theorem - Different outcomes

There exists a valid smart profile $\boldsymbol{B}$ for which the four greedy unravelling procedures and the two optimal procedures give different outcomes.

Also, the breadth-first and depth-first procedures by Kotsialou and Riley are different from the ones we propose, mainly in their treatment of abstentions.

Theorem - Termination of greedy unravelling

The algorithms of the four greedy unravelling procedures always terminate on valid smart profiles.

Theorem - Unravelling on single agent delegations

The four greedy unravelling procedures and $\texttt{Opt}$ coincide for liquid democracy profiles (one single-agent delegation per agent). $\texttt{MinMax}$ differs.

Language Restrictions

Various restrictions can be imposed on the agents' smart ballots:

- $\textsc{Liquid}$: language of ranked single delegations
- $\textsc{Bool}$: language of (contingent) propositional formulas in DNF
- $\textsc{Bool}^+$: language of (contingent) positive propositional formulas in DNF
- $\mathcal{L}^k$: language $\mathcal{L}$ where voters express at most k delegations

$B = ((\{B, C\}, B \wedge C), (\{D\}, D), 1)$ belongs to $\textsc{Bool}^2$.

$B = ((\{B\}, B), (\{C\}, C), (\{D\}, D), 1)$ belongs to $\textsc{Liquid}$.

$\textsc{Liquid}^1$ is the basic liquid democracy setting (+ backup vote).

Complexity of greedy unravelling

Theorem - Greedy unravelling is polynomial

Unravelling a smart profile with procedures $\mathbf{U}, \mathbf{DU}, \mathbf{RU}, \mathbf{DRU}$ *takes at most* $\mathcal{O}(n^2 \cdot \max_p(\boldsymbol{B}) \cdot \max_\varphi(\boldsymbol{B}))$ *time steps for* BOOL.

Expected result. Still it is true in the most general ballot language.

Complexity of Opt

BoundedOpt	
Instance:	Profile $\boldsymbol{B}$ in Bool, $M \in \mathbb{N}$
Question:	Is there a consistent certificate that unravels $\boldsymbol{B}$ with sum of ranks below M?

Complexity of $\texttt{Opt}$

BoundedOpt	
Instance:	Profile $\boldsymbol{B}$ in Bool, $M \in \mathbb{N}$
Question:	Is there a consistent certificate that unravels $\boldsymbol{B}$ with sum of ranks below M?

Theorem

BoundedOpt *is* NP*-complete.*

Proof sketch. For membership guess a certificate, test consistency and rank bound. For completeness reduction from SAT-DNF.

Theorem

An outcome of $\texttt{Opt}$ *can be found in polynomial time if ballots are* LIQUID.

Proof sketch. Non-trivial use of Edmonds algorithm for minimum spanning arborescence tree.

Complexity of `MinMax`

BoundedMinMax	
Instance:	Profile $\boldsymbol{B}$ in Bool, $M \in \mathbb{N}$
Question:	Is there a consistent certificate that unravels $\boldsymbol{B}$ with maximal rank below M?

Complexity of MinMax

BoundedMinMax	
Instance:	Profile $\boldsymbol{B}$ in Bool, $M \in \mathbb{N}$
Question:	Is there a consistent certificate that unravels $\boldsymbol{B}$ with maximal rank below M?

Theorem

BoundedMinMax *is* NP*-complete.*

Proof sketch. For membership guess a certificate, test consistency and rank bound. For completeness reduction from SAT-DNF.

Theorem

An outcome of MinMax *can be found in polynomial time if ballots are* LIQUID.

Proof sketch. Direct proof.

Participation results for liquid democracy

Liquid democracy participation axioms for voting rules:

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

Theorem: Cast-participation monotonic rules

Any monotonic rule paired with any of our unravelling procedures satisfies cast-participation in ranked liquid democracy.

Theorem: Guru-participation relative majority

Relative majority paired with any of our unravelling procedures does not satisfy guru-participation in ranked liquid democracy.

Conclusions

We started from three research questions:

1. elicit the social influence structure with multiagent delegations
2. break cycles with ranked delegations and back-up votes
3. define languages for expressing complex delegations on the user side

Conclusions

We started from three research questions:

1. elicit the social influence structure with multiagent delegations
2. break cycles with ranked delegations and back-up votes
3. define languages for expressing complex delegations on the user side

We presented a setting for multi-agent ranked delegations where:

- a smart ballot is a ranked list of multiagent delegations plus a backup vote
- an unravelling procedure transforms a profile of smart ballots into a standard voting profile
- two optimal and four greedy unravelling procedures
- algorithmic, computational, and (preliminary) axiomatic study

Rachael Colley, Umberto Grandi, and Arianna Novaro. Smart Voting. In *Proceeding of the the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.
+ working paper with optimal procedures and complexity results under preparation.

References

- Grammateia Kotsialou, Amrita Dhillon, Peter McBurney, and Luke Riley. Long read: how blockchain can make electronic voting more secure. USApp–American Politics and Policy Blog, 2020.
- Amrita Dhillon, Grammateia Kotsialou, Peter McBurney, Luke Riley, et al. Introduction to voting and the blockchain: some open questions for economists. Technical report, Competitive Advantage in the Global Economy (CAGE), 2019.
- James Green-Armytage. Direct voting and proxy voting. Constitutional Political Economy, 26(2):190–220, 2015.
- Gordon Tullock. Toward a mathematics of politics. Ann Arbor: University of Michigan Press, 1967.
- Gal Cohensius, Shie Mannor, Reshef Meir, Eli A. Meirom, and Ariel Orda. Proxy voting for better outcomes. In Proc. of the 16th Conference on Autonomous Agents and MultiAgent Systems (AAMAS), 2017.
- James C Miller. A program for direct and proxy voting in the legislative process. Public choice, 7(1):107–113, 1969.
- J. Behrens, A. Kistner, A. Nitsche, and B. Swierczek. Principles of Liquid Feedback. Interacktive Demokratie, 2014.

References

- Zoè Christoff and Davide Grossi. Binary voting with delegable proxy: An analysis of liquid democracy. In Proc. of the 16th Conference on Theoretical Aspects of Rationality and Knowledge (TARK), 2017.
- Markus Brill and Nimrod Talmon. Pairwise liquid democracy. In Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI), 2018.
- Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. Viscous democracy for social networks. Communications of the ACM, 54(6):129–137, 2011.
- Anson Kahng, Simon Mackenzie, and Ariel D Procaccia. Liquid democracy: An algorithmic perspective. In Proc. of the 32nd AAAI Conference on Artificial Intelligence (AAAI), 2018.
- Bruno Escoffier, Hugo Gilbert, and Adéle Pass-Lanneau. Iterative delegations in liquid democracy with restricted preferences. In Proc. of the 34th AAAI Conference on Artificial Intelligence (AAAI), 2020.
- Grammateia Kotsialou and Luke Riley. Incentivising participation in liquid democracy with breadth-first delegation. In Proc. of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS), 2020.
- Daan Bloembergen, Davide Grossi, and Martin Lackner. On rational delegations in liquid democracy. In Proc. of the 33rd AAAI Conference on Artificial Intelligence (AAAI), 2019.