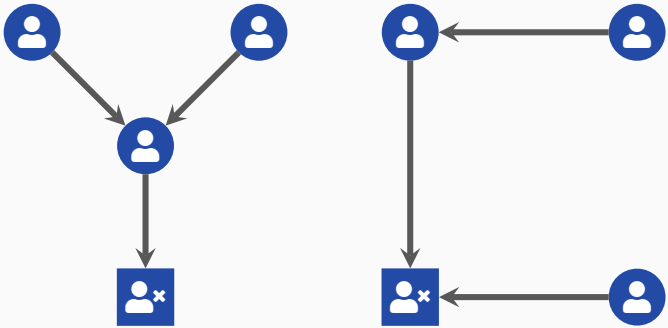
Liquid Democracy with Ranked Delegations

Markus Brill[1] Théo Delemazure[2] Anne-Marie George[3] Martin Lackner[4]
Ulrike Schmidt-Kraepelin[1]

[1]TU Berlin [2]Université Paris-Dauphine [3]University of Oslo [4]TU Wien

Formal Model of Democracy

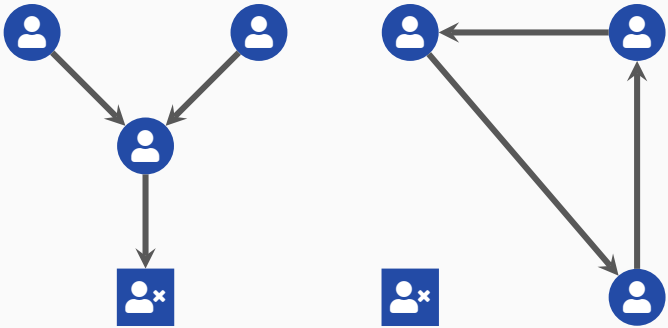
Liquid Democracy with Ranked Delegations



Voters can delegate their vote to **one** other voter.

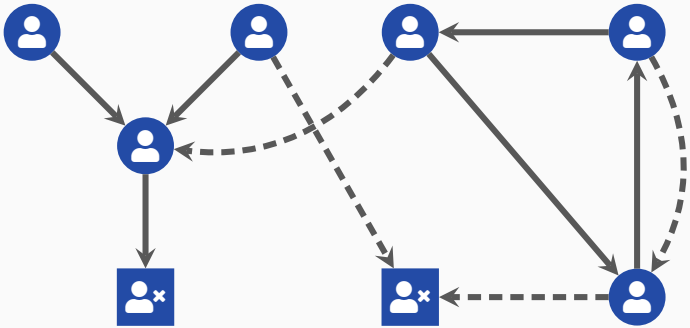
Implementations: LiquidFeedback, Sovereign, GoogleVotes

Liquid Democracy with Ranked Delegations



Implementations: LiquidFeedback, Sovereign, GoogleVotes

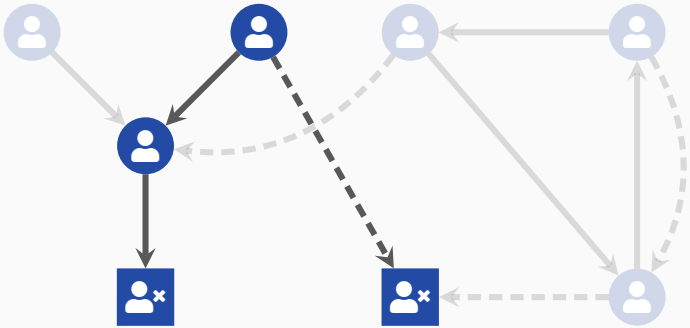
Liquid Democracy with Ranked Delegations



Voters can state **a set of approved delegates** together with a **ranking** among them.

Implementations: LiquidFeedback, Sovereign, GoogleVotes

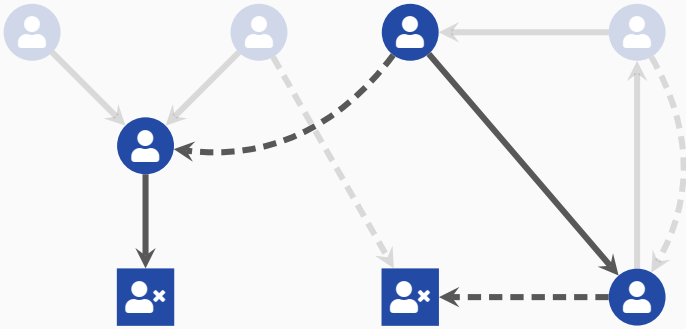
Liquid Democracy with Ranked Delegations



Voters can state **a set of approved delegates** together with a **ranking** among them.

Implementations: LiquidFeedback, Sovereign, GoogleVotes

Liquid Democracy with Ranked Delegations



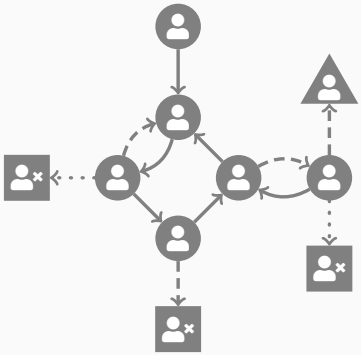
Voters can state **a set of approved delegates** together with a **ranking** among them.

Implementations: LiquidFeedback, Sovereign, GoogleVotes

Delegation Rules

Input: A directed delegation graph with a **rank** for every edge, and a partition of V into:

- **casting** voters : no outgoing edges
- **delegating** voters : reach at least one
- **isolated** voters : do not reach any



Delegation Rules

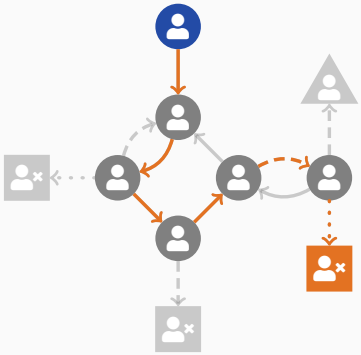
Input: A directed delegation graph with a **rank** for every edge, and a partition of V into:

- **casting** voters : no outgoing edges
- **delegating** voters : reach at least one
- **isolated** voters : do not reach any

Output: for each **delegating voter** :

- a path to a **casting voter**

A delegation rule indirectly outputs a **weight distribution** over casting voters.



We introduce a simple **graph-theoretical model** that can capture **rules** and **axioms** studied in the literature.

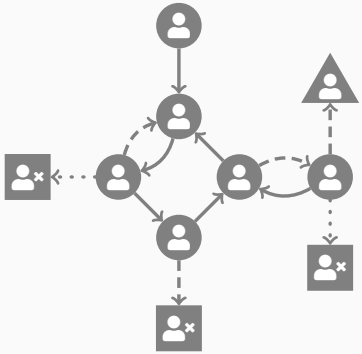
We introduce a simple **graph-theoretical model** that can capture **rules** and **axioms** studied in the literature.

We identify a natural **subclass** of delegation rules, perform an extensive **axiomatic analysis**, and compare all studied rules **empirically**.

Sequence Rules

Sequence Rules

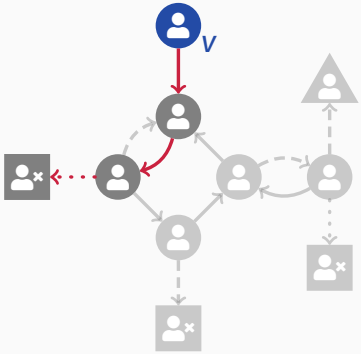
let $\mathcal{S}_v$ be the set of **rank sequences** of paths leading to casting voters for a delegating voter ***v***



$\mathcal{S}_v = \{$, , $\}$

Sequence Rules

let $\mathcal{S}_v$ be the set of **rank sequences** of paths leading to casting voters for a delegating voter v

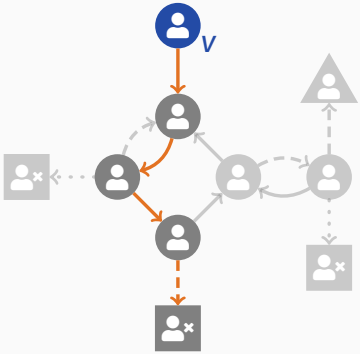


$\mathcal{S}_v = \{(1,1,3),$ $,$

$\}$

Sequence Rules

let $\mathcal{S}_v$ be the set of **rank sequences** of paths leading to casting voters for a delegating voter ***v***

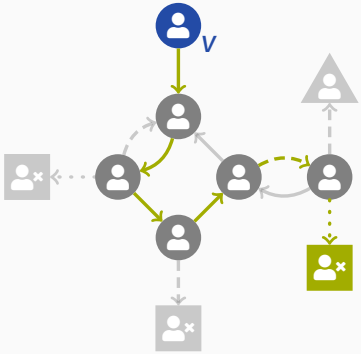


$\mathcal{S}_v = \{(1,1,3), (1,1,1,2),$

$\}$

Sequence Rules

let $\mathcal{S}_v$ be the set of **rank sequences** of paths leading to casting voters for a delegating voter ***v***

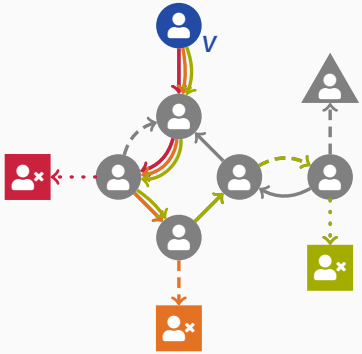


$\mathcal{S}_v = \{(1,1,3), (1,1,1,2), (1,1,1,1,2,3)\}$

Sequence Rules

let $\mathcal{S}_v$ be the set of **rank sequences** of paths leading to casting voters for a delegating voter v

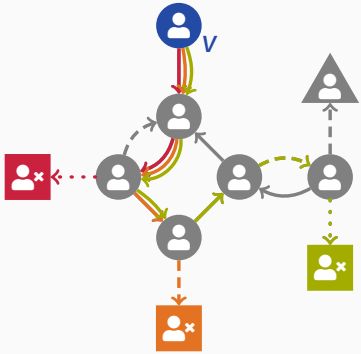
sequence rule: outputs $\max_{\triangleright}\{\mathcal{S}_v\}$ for each delegating voter v, where $\triangleright$ is an order over rank sequences



$\mathcal{S}_v = \{(1,1,3), (1,1,1,2), (1,1,1,1,2,3)\}$

Sequence Rules

Let $\triangleright_{\text{lex}}$ be the lexicographical order.



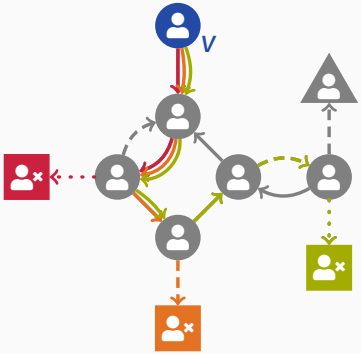
$$\mathcal{S}_v = \{(1,1,3), (1,1,1,2), (1,1,1,1,2,3)\}$$

Sequence Rules

Let $\triangleright_{\text{lex}}$ be the lexicographical order.

- depth-first delegation: rule induced by $\triangleright_{\text{lex}}$
- breadth-first delegation: orders sequences by length, tie-breaking according to $\triangleright_{\text{lex}}$

[Kotsialou and Riley (AAMAS 2020)]



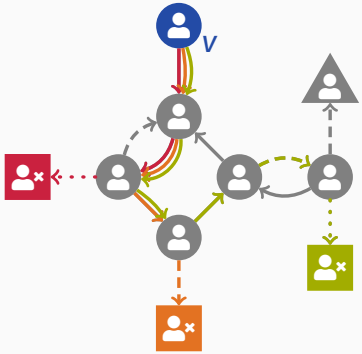
$\mathcal{S}_v = \{(1,1,3), (1,1,1,2), (1,1,1,1,2,3)\}$

Sequence Rules

Let $\triangleright_{\text{lex}}$ be the lexicographical order.

- depth-first delegation: rule induced by $\triangleright_{\text{lex}}$
- breadth-first delegation: orders sequences by length, tie-breaking according to $\triangleright_{\text{lex}}$

  [Kotsialou and Riley (AAMAS 2020)]
- min-sum: orders sequences by the sum of ranks, breaks ties according to $\triangleright_{\text{lex}}$



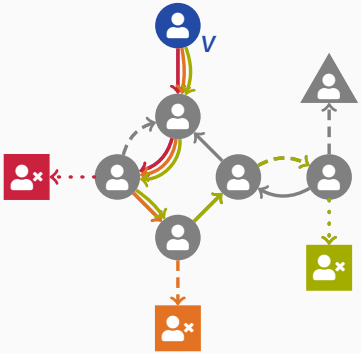
$$\mathcal{S}_v = \{(1,1,3), (1,1,1,2), (1,1,1,1,2,3)\}$$

Sequence Rules

Let $\triangleright_{\text{lex}}$ be the lexicographical order.

- **depth-first delegation**: rule induced by $\triangleright_{\text{lex}}$
- **breadth-first delegation**: orders sequences by length, tie-breaking according to $\triangleright_{\text{lex}}$

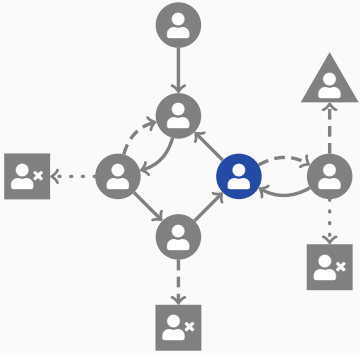
  [Kotsialou and Riley (AAMAS 2020)]
- **min-sum:** orders sequences by the sum of ranks, breaks ties according to $\triangleright_{\text{lex}}$
- **leximax:** $s \triangleright s'$ iff $\sigma(s) \triangleright_{\text{lex}} \sigma(s')$, where σ sorts s by non-increasing ranks, e.g., $\sigma(1,1,1,2) = (2,1,1,1) \triangleright_{\text{lex}} (3,1,1) = \sigma(1,1,3)$



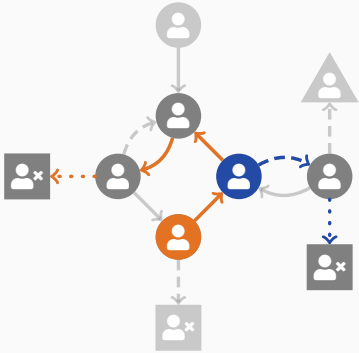
$\mathcal{S}_v = \{(1,1,3), (1,1,1,2), (1,1,1,1,2,3)\}$

Axiomatic Analysis

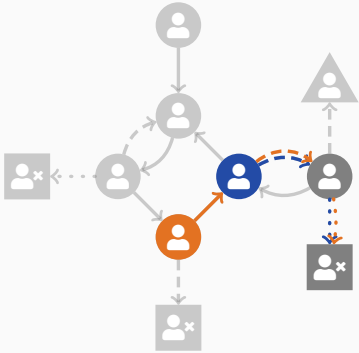
Confluence: for all : all paths intersecting with use the same outgoing edge of .



Confluence: for all 👤: all paths intersecting with 👤 use the same outgoing edge of 👤.



Confluence: for all : all paths intersecting with use the same outgoing edge of .

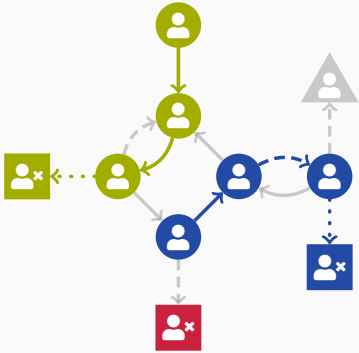


Confluence

Confluence: for all : all paths intersecting with use the same outgoing edge of .

- output of the delegation rule can be **communicated** more easily
- a single representative helps “to preserve the high level of **accountability** guaranteed by classical liquid democracy.”

[Gölz et al., WINE 2018]



Confluence: for all 👤: all paths intersecting with 👤 use the same outgoing edge of 👤.

Theorem

*Building upon a **characterization** of orders ▷ that induce **confluent** sequence rules, we show:*

- *breadth-first delegation*, *min-sum*, *diffusion*, *and leximax are confluent*
- *depth-first delegation is not confluent*

Copy-robustness

Copy-robustness: A delegating voter has a direct path to its casting voter . If becomes a casting voter, the joint voting power of & remains equal.

[Behrens & Swierczek (LDJ, 2015)]

Copy-robustness

Copy-robustness: A delegating voter has a direct path to its casting voter . If becomes a casting voter, the joint voting power of & remains equal.

[Behrens & Swierczek (LDJ, 2015)]

Impossibility Theorem

No sequence rule is both confluent and copy-robust. Hence, breadth-first delegation, min-sum, diffusion, and leximax are not copy-robust.

Copy-robustness

Copy-robustness: A delegating voter has a direct path to its casting voter. If becomes a casting voter, the joint voting power of & remains equal.
[Behrens & Swierczek (LDJ, 2015)]

Impossibility Theorem

No sequence rule is both **confluent** and **copy-robust**. Hence, **breadth-first delegation**, **min-sum**, **diffusion**, and **leximax** are not copy-robust.

Characterization

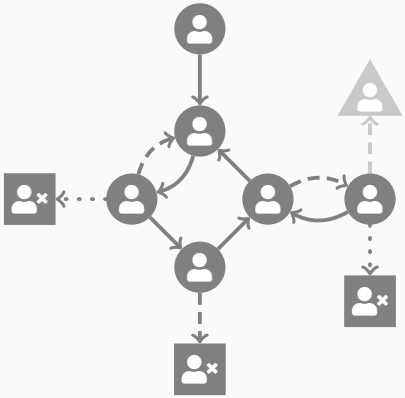
Depth-first delegation is the only **sequence rule** that is **copy-robust** and satisfies **weak lexicographical tie-breaking**.

Can we obtain **confluence** and **copy-robustness** by going **beyond** sequence rules?

Branching Rules

Branching rules

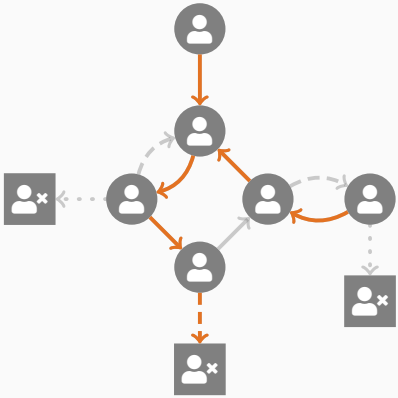
C-branching: Acyclic subgraph such that all delegating voters have **exactly one** outgoing edge.



Branching rules

C-branching: Acyclic subgraph such that all delegating voters have **exactly one** outgoing edge.

Branching rules select delegations on a global level while **Sequence rules** select delegations for each voter

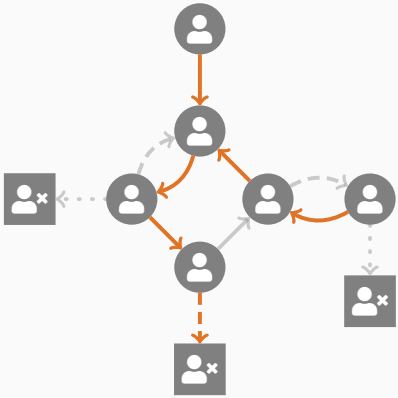


Branching rules

C-branching: Acyclic subgraph such that all delegating voters have **exactly one** outgoing edge.

Branching rules select delegations on a global level while **Sequence rules** select delegations for each voter

Borda branching: Select a *C*-branching *B* that minimizes the total sum of ranks

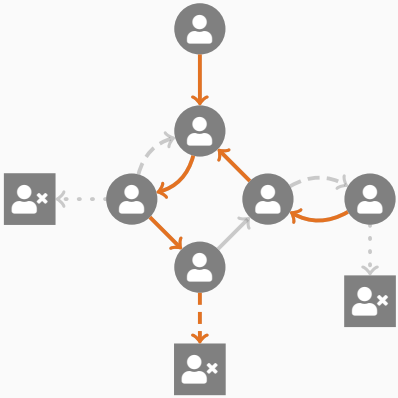


Branching rules

C-branching: Acyclic subgraph such that all delegating voters have **exactly one** outgoing edge.

Branching rules select delegations on a global level while **Sequence rules** select delegations for each voter

Borda branching: Select a *C*-branching *B* that minimizes the total sum of ranks



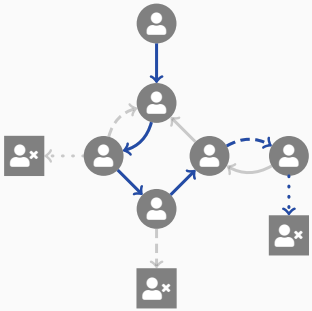
Theorem

*Borda branching (with an appropriate tie-breaking rule) satisfies **confluence** and **copy-robustness**.*

Popular Branchings [Kavitha et al. (Math. Prog., 2021)]

Pairwise majority comparisons:

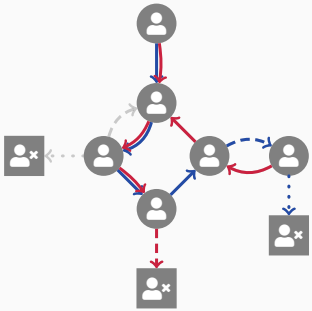
$$\Delta(B_1, B_2) := \quad \#\text{ nodes in favor of } B_1 \\ - \#\text{ nodes in favor of } B_2$$



Popular Branchings [Kavitha et al. (Math. Prog., 2021)]

Pairwise majority comparisons:

$$\Delta(B_1, B_2) := \quad \#\text{ nodes in favor of } B_1 \\ - \#\text{ nodes in favor of } B_2$$

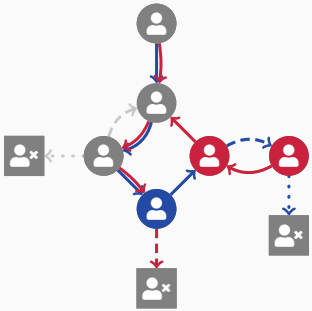


Popular Branchings [Kavitha et al. (Math. Prog., 2021)]

Pairwise majority comparisons:

$\Delta(B_1, B_2) :=$ # nodes in favor of B_1

$-$ # nodes in favor of B_2



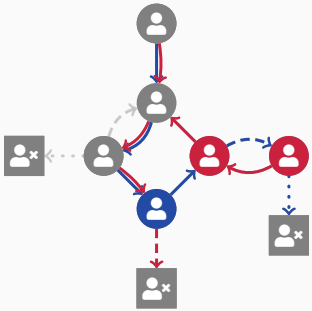
Popular Branchings [Kavitha et al. (Math. Prog., 2021)]

Pairwise majority comparisons:

$$\Delta(B_1, B_2) := \#\text{ nodes in favor of } B_1 - \#\text{ nodes in favor of } B_2$$

Unpopularity margin:

$$\text{unpopularity}(B) := \max_{B'}(\Delta(B', B))$$



Theorem

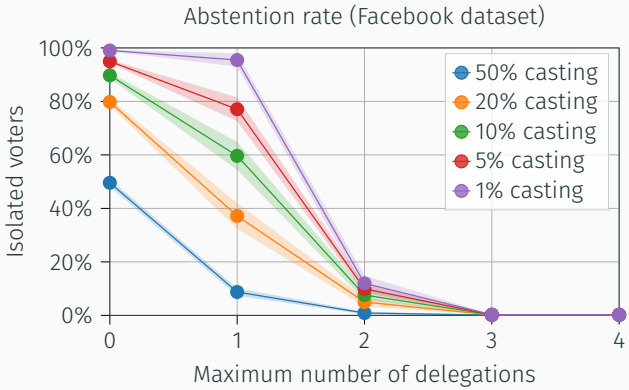
*A **popular branching**, i.e., a branching with unpopularity $= 0$ does not always exist.*

Empirical Results

Impact of backup delegation on abstention rate

On the classic liquid democracy setting, each voter can delegates to **at most one voter**. This cause the issue of **delegation cycles** and **lost ballots**.

With ranked delegation, we achieve **far better participation rate**, even when only 1% of all voters are actually voting.



Results

Twitter dataset ($n = 456626$)	Unpop.	AvgRank	AvgLen	MaxWeight
Breadth-first	223746	3.4	1.16	27397
MinSum	105023	1.37	1.89	31963
Leximax	13699	1.04	5.59	118722
BordaBranching	16	1.0	6.0	132421
Depth-first			6.05	127855

Facebook dataset ($n = 63731$)	Unpop.	AvgRank	AvgLen	MaxWeight
Breadth-first	28678	3.29	1.27	162
MinSum	12746	1.35	2.04	224
Leximax	2567	1.08	3.97	539
BordaBranching	12	1.03	4.79	748
Depth-first			5.0	713

MaxWeight: Maximum accumulated voting weight of a casting voter. Mechanism avoiding **super voters** were studied by Gölz et al. (WINE, 2018).

Unpopularity: Worst-case **majority comparison** [Kavitha et al. (Math. Prog. 2021)]

Results

Twitter dataset ($n = 456626$)	Unpop.	AvgRank	AvgLen	MaxWeight
Breadth-first	223746	3.4	1.16	27397
MinSum	105023	1.37	1.89	31963
Leximax	13699	1.04	5.59	118722
BordaBranching	16	1.0	6.0	132421
Depth-first			6.05	127855

Facebook dataset ($n = 63731$)	Unpop.	AvgRank	AvgLen	MaxWeight
Breadth-first	28678	3.29	1.27	162
MinSum	12746	1.35	2.04	224
Leximax	2567	1.08	3.97	539
BordaBranching	12	1.03	4.79	748
Depth-first			5.0	713

MaxWeight: Maximum accumulated voting weight of a casting voter. Mechanism avoiding **super voters** were studied by Gölz et al. (WINE, 2018).

Unpopularity: Worst-case **majority comparison** [Kavitha et al. (Math. Prog. 2021)]

Observations

- **trade-off** between minimizing unpopularity and maximum weight
- delegation rules can be aligned on a **spectrum**
- **leximax** outperforms **diffusion** on all metrics

Summary

Summary

In this talk:

- introduction of a simple graph-theoretical model
- formalization of the class of sequence rules
- impossibility result for copy-robust and confluent sequence rules
- Borda branching satisfies copy-robustness and confluence
- characterization of depth-first delegation via copy-robustness

Summary

In this talk:

- introduction of a simple graph-theoretical model
- formalization of the class of sequence rules
- impossibility result for copy-robust and confluent sequence rules
- Borda branching satisfies copy-robustness and confluence
- characterization of depth-first delegation via copy-robustness

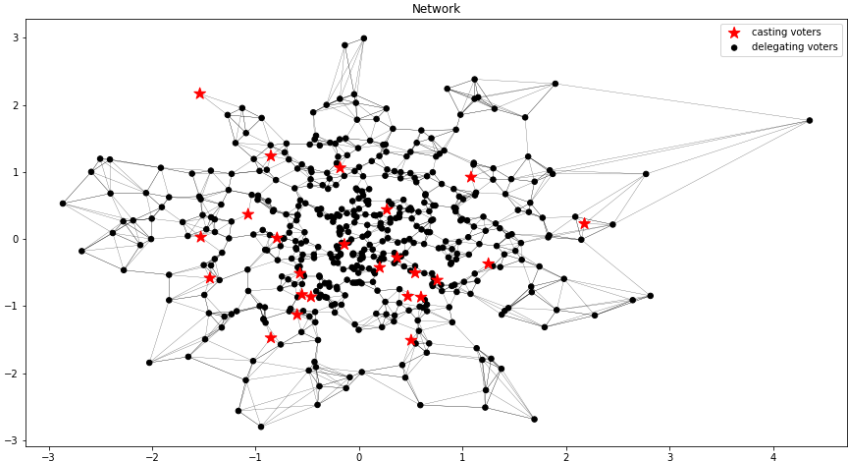
Not mentioned in this talk:

- a generalization of a result by Kotsialou and Riley (AAMAS 2020) implying that almost all studied sequence rules satisfy guru participation
- Borda branching satisfies guru participation
- an axiomatic characterization of breadth-first delegation
- a proof that diffusion is a sequence rule by uncovering its respective order

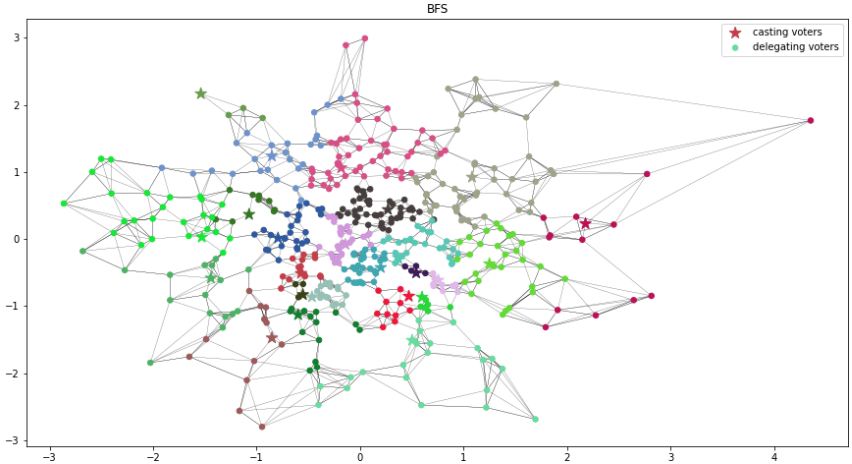
Thanks

Thanks for your attention !

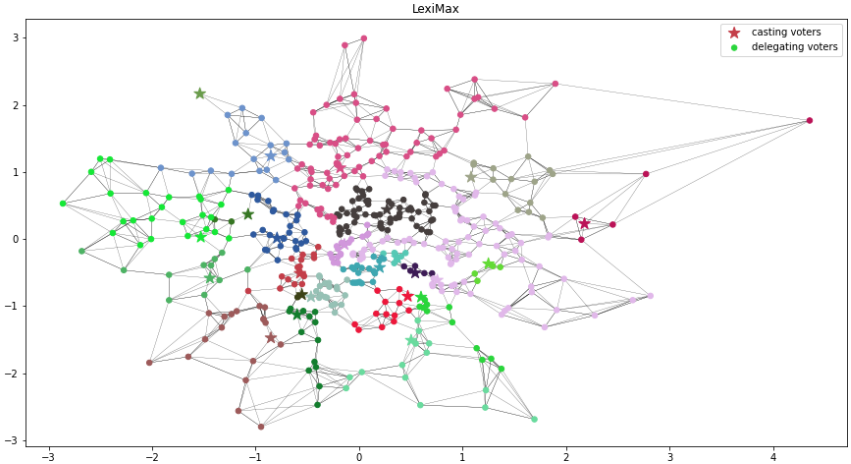
Bonus : The distance-based method



Bonus : The distance-based method



Bonus : The distance-based method



Bonus : The distance-based method