Liquid Democracy with Ranked Delegations

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

[1] Research Group Efficient Algorithms, TU Berlin, Germany [2] LAMSADE, Université Paris-Dauphine, France
[3] Department of Informatics, University of Oslo, Norway [4] Databases and Artificial Intelligence Group, TU Wien, Austria
brill@tu-berlin.de, theo.delemazure@ens.fr, annemage@ifi.uio.no, lackner@dbai.tuwien.ac.at,
u.schmidt-kraepelin@tu-berlin.de

Abstract

Liquid democracy is a novel paradigm for collective decision-making that gives agents the choice between casting a direct vote or delegating their vote to another agent. We consider a generalization of the standard liquid democracy setting by allowing agents to specify multiple potential delegates, together with a preference ranking among them. This generalization increases the number of possible delegation paths and enables higher participation rates because fewer votes are lost due to delegation cycles or abstaining agents. In order to implement this generalization of liquid democracy, we need to find a principled way of choosing between multiple delegation paths. In this paper, we provide a thorough axiomatic analysis of the space of *delegation rules*, i.e., functions assigning a feasible delegation path to each delegating agent. In particular, we prove axiomatic characterizations as well as an impossibility result for delegation rules. We also analyze requirements on delegation rules that have been suggested by practitioners, and introduce novel rules with attractive properties. By performing an extensive experimental analysis on synthetic as well as real-world data, we compare delegation rules with respect to several quantitative criteria relating to the chosen paths and the resulting distribution of voting power. Our experiments reveal that delegation rules can be aligned on a spectrum reflecting an inherent trade-off between competing objectives.

1 Introduction

Liquid democracy is a novel decision-making paradigm that aims to reconcile the idealistic appeal of direct democracy with the practicality of representative democracy by giving agents a choice regarding their mode of participation: For every given issue, agents can choose whether they want to vote directly or whether they want to *delegate* their vote to another agent. The mode of participation can differ for different issues and the choice of mode (including the choice of whom to delegate to) can be altered at any time. This enables a flexible and dynamic scheme of representation on a per-issue basis (Blum and Zuber 2016).

An important principle of liquid democracy is the transitivity of delegation (sometimes called *metadelegation*): if A delegates to B and B delegates to C, then C has a total voting weight of 3 (her own vote plus those of A and B). More generally, delegated votes travel along *delegation paths* until they reach an agent who votes directly. This agent at the endpoint of the path (sometimes referred to as "guru") then has a voting weight corresponding to all the agents whose delegation paths end up with him or her.

Introducing the option of delegating one's vote arguably lowers the bar for participation, as it does not require agents to get informed about an issue before making their vote count (Ford 2002; Boldi et al. 2011; Valsangiacomo 2021). This is particularly true if delegations can be declared globally and/or for whole subject areas.[1] As a result, liquid democracy has the potential to achieve substantially higher rates of (direct and indirect) participation than direct voting.

However, vote delegations do not always work out as intended. For instance, if a delegation path ends in an agent who *abstains* (i.e., neither delegates nor votes directly), the vote is inevitably lost. Moreover, if a group of agents delegate to each other in a cyclic fashion and no group member votes directly, they form a "delegation cycle" and all their votes, together with the votes of agents whose delegation paths lead to the cycle, are lost. Finally, delegation paths can potentially get very long, calling into question the degree of trust between the agents at the beginning and end of the path.

To address these issues, we consider an extension of liquid democracy that lets agents declare *multiple* potential delegates, together with a ranking among them specifying the agent's delegation preferences: By ranking potential delegate X higher than potential delegate Y, an agent indicates that she would prefer delegating her vote to X, but would also be happy with a delegation to Y in the case that the delegation to X leads to one of the problems mentioned in the previous paragraph. This functionality was implemented, e.g., in a liquid democracy experiment at *Google* (Hardt and Lopes 2015). Allowing agents to declare *ranked delegations* enlarges the space of possible delegation paths and thereby increases the likelihood of "successful" delegation paths. In the case that multiple such paths exist, however, we need a principled way to choose among them. This is accomplished by so-called *delegation rules*, which select delegation paths based on the delegation preferences of agents.

[1] For instance, this is possible in the open-source software *LiquidFeedback* (Behrens et al. 2014), where more fine-grained delegations override global delegations, and direct voting on any particular issue overrides all delegations.

Our Contribution We propose a graph-theoretic model capturing the ranked delegation setting and study the space of delegation rules. We define *sequence rules* as the subclass of delegation rules that can be rationalized by an ordering over delegation paths that only takes the ranks of edges into account. For most delegation rules considered in the literature, we establish their membership in this class by uncovering their respective order. Our systematic approach leads us to introduce novel sequence rules with attractive properties.

In our axiomatic analysis, we generalize a result by Kotsialou and Riley (2020), showing that all but one rule studied in our paper satisfy *guru-participation*. We also study the axiom *copy-robustness*, which is motivated by practical considerations. Notwithstanding a strong impossibility result for the subclass of sequence rules, we construct a copy-robust delegation rule by taking a "global" approach. Lastly, we give axiomatic characterizations for two sequence rules.

We complement our theoretical results with an extensive experimental analysis using synthetic and real-world data. We empirically compare delegation rules with respect to several quantitative criteria such as the length of chosen paths or the distribution of voting power that results from those paths. Interestingly, we find that the delegation rules form a spectrum, reflecting an inherent trade-off between short paths and paths containing top-ranked delegations.

Related Work As illustrated by Behrens (2017), some of the ideas underlying liquid democracy date back to Dodgson (1884), Tullock (1967), and Miller (1969). Liquid democracy has been studied theoretically, and applied practically, in various ways in recent years (Paulin 2020).

Liquid democracy settings with ranked delegations have been considered by several authors. Behrens and Swierczek (2015) illustrate the incompatibility of a set of axioms, several of which we discuss in Section 4. Kotsialou and Riley (2020) focus on two simple delegation rules and on participation axioms. Their observation that *breadth-first delegation* satisfies guru-participation also follows from our more general result in Section 4. Kavitha et al. (2021) establish a connection between ranked delegations and branchings in directed graphs, which we build upon in Section 3.3, and focus on the computational complexity of finding "popular" branchings. Colley, Grandi, and Novaro (2020) propose a generalisation of the ranked delegations setting where agents can express complex delegations involving *sets* of delegates on each level of the ranking. When restricted to our setting, their "unravelling procedures" reduce to variants of the rule we call *Diffusion*. Both Kavitha et al. and Colley et al. assume that agents specify *backup votes*, i.e., votes that are used when no delegation path exists. In contrast, our model does not necessitate this rather demanding assumption.

Other settings with multiple delegations have been considered as well. Gölz et al. (2018) let agents specify multiple delegation options, but without specifying preferences among them. Their objective is to assign delegations in such a way as to minimize the maximal voting weight of agents. Kahng, Mackenzie, and Procaccia (2021) consider an epistemic setting in which each agent has a competence level (i.e., probability of making the "correct" voting decision) and an approved subset of other agents. They are interested in (possibly randomized) delegation mechanisms that increase the likelihood of a correct decision compared to direct voting. Since delegation is only allowed to more competent agents, there can be no delegation cycles in their model. Finally, Christoff and Grossi (2017) and Brill and Talmon (2018) let agents delegate different decisions to different delegates and explore ways to ensure individual rationality.

2 The Model

We start our exploration of the ranked delegation setting by presenting a graph-theoretic model. We focus on a single issue that is to be decided upon and assume that we are given (1) the set of agents that are casting a direct vote (*casting* voters) as well as (2) for each non-casting agent, a (possibly empty) set of other agents together with a ranking over them representing delegation preferences.[2] Based on this information, we want to assign non-casting agents to casting voters by choosing delegation paths. Our model focuses on determining the voting weights of casting voters and is, therefore, independent of any particular method for aggregating the votes of the casting voters. Separating the delegation mechanism from the voting method allows us to analyze the former in isolation, and also simplifies the model.[3]

We represent a ranked delegation instance as a pair (G, r), where $G = (C \cup D \cup I, E)$ is a directed graph. The set of vertices of G corresponds to the set of all voters (agents) and is partitioned into three sets C, D, and I with the following properties:

- nodes in C have no outgoing edges in G (we refer to voters in C as *casting voters*);
- for each node in D, there exists at least one path in G to a voter in C (we refer to voters in D as *delegating voters*);
- for each node in I, there exists no path in G leading to a voter in C (we refer to voters in I as *isolated voters*).

For the set of all voters we write $V = C \cup D \cup I$. Note that for any graph $G = (V, E)$ with a distinguished set $C \subseteq V$ of casting voters, the sets D and I are uniquely defined. The second element of the instance, r, is a *rank function* on the set of edges that encodes, for every node, a linear order over its set of outgoing edges. Formally, $r : E \to \mathbb{N}_{\geq 1}$ is a function such that $\{r(e) \mid e \in \delta^+_G(v)\} = \{1, \dots, |\delta^+_G(v)|\}$ for all $v \in V \setminus C$, where $\delta^+_G(v)$ is the set of outgoing edges of v in G. If $r((v, x)) < r((v, y))$ this is interpreted as voter v preferring to delegate to voter x over delegating to voter y. For an example of a ranked delegation instance, see Figure 1.

For a non-casting voter $v \in V \setminus C$, we denote the set of all simple paths from v to some casting voter by

$$\mathcal{P}_v := \{P \mid P \text{ is a simple } v\text{-}w\text{-path and } w \in C\}.$$

Observe that $\mathcal{P}_v$ is empty if and only if $v \in I$. This is in particular the case when $\delta^+_G(v) = \emptyset$ (in which case we call v an

[2] An empty set of potential delegates corresponds to abstaining.

[3] As we will see in Section 4, axioms formulated in more complex models (Behrens and Swierczek 2015; Kotsialou and Riley 2020; Colley, Grandi, and Novaro 2020) can be translated into ours.

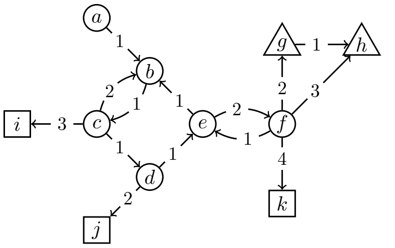


Figure 1: Example of a ranked delegation instance (G, r). Casting voters are indicated by squares, delegating voters by circles and isolated voters by triangles.

abstaining voter), but it also happens if v has only outgoing edges towards other isolated voters. In Figure 1, $\mathcal{P}_g = \emptyset$ and $\mathcal{P}_d = \{(d, j), (d, e, b, c, i), (d, e, f, k)\}$. [4]

Our task is now to define sensible rules which assign delegating voters to casting ones (via a path in G).

Definition 1. *A* delegation rule *f is a function that takes as input a ranked delegation instance (G, r) and a delegating voter $v \in D$ and outputs a path from v to a casting voter, i.e., $f(G, r, v) \in \mathcal{P}_v$ for all $v \in D$. We also write $f(v)$, whenever the instance (G, r) is clear from the context.*

Since isolated voters cannot be contained in any of the paths in $\mathcal{P}_v$, their influence on the problem is limited. Thus, it will often be convenient to omit the isolated voters from the instance. However, we argue that the information about their existence should still be reflected within the rank function r.[5] More formally, we define the graph $\bar{G} = (\bar{V}, \bar{E})$ with $\bar{V} = C \cup D$ and $\bar{E} = \{(u, v) \in E \mid u, v \in \bar{V}\}$, and $\bar{r}(e) = r(e)$ for all $e \in \bar{E}$ and refer to $(\bar{G}, \bar{r})$ as the *reduced instance* of (G, r).

3 Delegation Rules

Before introducing and analyzing several delegation rules, we formalize two fundamental concepts that will help us to classify delegation rules: *confluence* and *sequence rules*.

To motivate the former, observe that the definition of a delegation rule does not forbid that selected paths cross. Consider for example the instance in Figure 1: There exists a delegation rule assigning path $(c, d, e, f, \dots)$ to agent c but path $(d, e, b, \dots)$ to agent d. This can be seen as counter-intuitive as it is incompatible with the view that voter e takes the decision on how to delegate incoming delegations based on its delegation preferences. Also, it can be difficult for voter e to keep track and evaluate the decisions made by all casting voters who have been reached via delegation paths passing through e. In this context, Gölz et al. (2018) argue that a single representative per delegating voter is needed “to preserve the high level of accountability guaranteed by classical liquid democracy.” We define a natural subclass of delegation rules in which this is guaranteed. In reference to *confluent flows* in networks (Chen, Rajaraman, and Sundaram 2006), we call such delegation rules *confluent*.

Definition 2. *A delegation rule f is* confluent *if, for every instance (G, r), every delegating voter $v \in D$ has outdegree exactly one in $(V, \bigcup_{v \in D} f(G, r, v))$.*

Besides confluence, two natural—and often conflicting—objectives are minimizing the length of paths and minimizing the ranks of edges contained in paths. Short paths are particularly motivated by the fact that delegations are a form of trust propagation and it is debatable to what extent trust is transitive. At the same time, votes should be delegated to highly trusted agents, motivating the selection of paths with top-ranked edges. Both of these objectives can be evaluated by only considering the sequence of ranks appearing along a path. Indeed, most delegation rules introduced in the literature can be expressed as choosing, among all available delegation paths, the one with the “best” rank sequence. We formalize this subclass of delegation rules as *sequence rules*.

To do so, we need some notation. Let $\mathscr{S}$ be the set of all finite sequences of numbers in $\mathbb{N}_{\geq 1}$.[6] We define the *sequence* of a path $P = (e_1, e_2, \dots, e_\ell)$ as $s(P) := (r(e_1), \dots, r(e_\ell))$ and denote the set of all sequences from v to some casting voter by $\mathcal{S}_v := \{s(P) \mid P \in \mathcal{P}_v\}$. For a sequence s, we write s_i to refer to the i-th element of s, $|s|$ for the length of s, $\max(s)$ for the maximal entry, and use the notation (s, x) to denote s extended by a number $x \in \mathbb{N}_{\geq 1}$ or a sequence x.

We call a delegation rule a sequence rule if it, explicitly or implicitly, defines a relation $\rhd$ over $\mathscr{S}$ and, when confronted with $\mathcal{P}_v$, guarantees a unique maximum element of $\mathcal{S}_v$ w.r.t. $\rhd$ and selects the corresponding path. It is clearly sufficient to define $\rhd$ on sets of *comparable* sequences:

Definition 3. *Two sequences $s, s' \in \mathscr{S}$ are said to be comparable if there exists an instance (G, r) and a vertex $v \in V$ such that $\mathcal{S}_v = \{s, s'\}$. A set $\mathcal{S} \subseteq \mathscr{S}$ is said to be comparable if all elements are pairwise comparable.*

Not all pairs of sequences are comparable; e.g. , there is no instance with $\mathcal{S}_v = \{(1), (1, 2)\}$ for some v, as the first elements of (1) and $(1, 2)$ would correspond to the same edge e (of rank 1) that is outgoing from v. Thus, the head of e is a casting voter (due to $(1) \in \mathcal{S}_v$) but has an outgoing edge (due to $(1, 2) \in \mathcal{S}_v$), a contradiction. This observation can be extended to any situation in which s is a prefix of s' (i.e. $|s| < |s'|$ and $s_i = s'_i$ for all $i \in \{1, \dots, |s|\}$). Proofs for results marked by (★) can be found in the full version of this paper (Brill et al. 2021).

Proposition 4 (★)**.** *Two distinct non-empty sequences $s, s' \in \mathscr{S}$ are comparable iff none is a prefix of the other.*

We are now ready to define the class of sequence rules.

Definition 5. *A delegation rule f is a* sequence rule *if there exists a relation $\rhd$ over $\mathscr{S}$, which, if restricted to any comparable subset of $\mathscr{S}$, is a linear order and for any instance (G, r) and $v \in V$ it holds that $s(f(v)) = max_{\rhd}\{\mathcal{S}_v\}$.*

[4] We interpret paths as *edge* sequences. However, for brevity, we state *node* sequences whenever referring to paths in Figure 1. For convenience, we interpret paths as sets of edges in Definition 2.

[5] E.g., consider the edge from agent f to agent k in Fig. 1. If we were to delete all isolated agents and adjust the rank function, the new rank of this edge would be 2, even though k is only f's fourth choice. Such examples can be made arbitrarily “extreme.”

[6] For technical reasons, $\mathscr{S}$ also includes the empty sequence $()$ with length 0 and (by convention) maximum rank 0.

If a relation $\rhd$ proves that f is a sequence rule, we also say that $\rhd$ *induces* f.

3.1 Basic Sequence Rules

In the following we describe rules which are induced by natural linear orders over the entire set $\mathscr{S}$. The first two rules have also been considered by Kotsialou and Riley (2020). Let $\rhd_{\text{lex}}$ be the *lexicographic order* over $\mathscr{S}$. That is, for two distinct sequences $s, s' \in \mathscr{S}$, let i be the smallest index such that $s_i \neq s'_i$. Then, $s \rhd_{\text{lex}} s'$ iff $s_i < s'_i$. If no such index exists, then $s \rhd_{\text{lex}} s'$ iff $|s| < |s'|$.

Depth-first delegation (DFD): The sequence rule induced by the lexicographic order $\rhd_{\text{lex}}$ over $\mathscr{S}$.

Breadth-first delegation (BFD): The sequence rule induced by the linear order over $\mathscr{S}$ that prefers shorter sequences over longer ones and uses $\rhd_{\text{lex}}$ for tie-breaking.

While DFD focuses on the objective of selecting delegations with top ranks, BFD primarily selects short paths. As a consequence, each of them have at least one obvious drawback: DFD is mostly indifferent about the length of a path and may select very long paths, whereas BFD mostly ignores the ranks of a path and hence can select paths containing edges that are ranked extremely low.

We propose a third, natural delegation rule, which arguably strikes a better balance between the two objectives of minimizing the length and the ranks of a path.

MinSum: The sequence rule induced by the linear order over $\mathscr{S}$ that orders sequences by their sum of ranks and uses the lexicographic order for tie-breaking.

To illustrate these rules, consider again the example instance in Figure 1. For delegating voter a, DFD selects the path (a, b, c, d, e, f, k) (with sequence $(1, 1, 1, 1, 2, 4)$), BFD selects the path (a, b, c, i) (with sequence $(1, 1, 3)$), and MinSum selects (a, b, c, d, j) (with sequence $(1, 1, 1, 2)$).

Next, we characterize confluent sequence rules.

Theorem 6 (★). *A sequence rule induced by $\rhd$ is confluent if and only if for all $s \in \mathscr{S}$ we have (i) if s is comparable to some $s' \in \mathscr{S}$ and $x \in \mathbb{N}_{\geq 1}$, then $s \rhd s' \Leftrightarrow (x, s) \rhd (x, s')$ and (ii) if $s' \in \mathscr{S}$ and s is comparable to (s', s), then $s \rhd (s', s)$.*

Making use of this characterization we show:

Corollary 7 (★). *BFD and MinSum are confluent. DFD is not confluent.*

3.2 Advanced Sequence Rules

We introduce the new delegation rule *Diffusion* which is inspired by a propagation process similar to those studied in the opinion diffusion literature. We give a motivation for this connection upfront, and then define the rule formally.

Confronted with an instance of our setting, we argue that there are certain delegation paths which are, in a sense, *best possible*: Let x be the minimum rank of an incoming edge of *any* casting voter. Then, all delegating voters that have a direct edge to a casting voter with rank x should be assigned this path. The rationale behind this statement is that for every voter v, every path in $\mathcal{P}_v$ contains at least one edge with rank at least x. Hence a one-step path with rank x seems preferable to any other path. However, typically, not all voters have such a path. A natural continuation of our argument goes as follows: In a second round, we call casting voters and delegating voters that already got assigned to a casting voter *assigned*. We treat all assigned voters as casting voters and repeat the process until all delegating voters become assigned. The path assignment is then derived by following the one-step paths. A similar process has been described by Colley, Grandi, and Novaro (2020) within their unravelling procedures "basic update" and "direct vote priority."[7]

Diffusion: Initialize the set of *assigned* voters: $A \leftarrow C$.

While $(A \neq V \setminus I)$, repeat the following steps:

1. $F \leftarrow \arg\min\{r(e) \mid e \in \delta^-_G(A)\}$, where $\delta^-_G(A)$ is the set of edges in G having their head in A and tail in $V \setminus A$.
2. $A \leftarrow A \cup \{v \mid (v, w) \in F\}$
3. $f(v) = ((v, w), f(w))$ for all $(v, w) \in F$

The assignment in step 3 is well defined as voters' preferences are strict orders and thus there exist no two edges $(v, w), (v, w') \in F$. This immediately implies confluence.

Proposition 8. *Diffusion is confluent.*

One may wonder whether this seemingly "global" process (in the sense that we need to know the entire graph to determine the minimal rank x) can be explained by an order over $\mathscr{S}$ that can be applied to each delegating voter "locally." Stated differently, is diffusion a sequence rule? Surprisingly, we answer this question in the affirmative: We define the order $\rhd_{\text{diff}}$ (which will prove our claim) for sequences without a joint prefix first and then later extend it to any two comparable sequences in a straightforward way. Let s and s' be two comparable sequences with no joint prefix. We define $s \rhd_{\text{diff}} s'$ if one of the following conditions holds:

(i) $\max(s) < \max(s')$;

(ii) $\max(s) = \max(s')$ and $|\arg\max(s)| < |\arg\max(s')|$;

(iii) $\max(s) = \max(s')$, $|\arg\max(s)| = |\arg\max(s')|$, and ($\bar{s} \rhd_{\text{diff}} \bar{s}'$ or $\bar{s} = ()$);

where $\bar{s}$ (resp. $\bar{s}'$) is defined as the prefix of s (resp. s') ending just before the first entry of rank $\max(s)$.

The relation $\rhd_{\text{diff}}$ can now easily be extended to two comparable sequences $(t, s), (t, s')$ having a joint prefix $t \in \mathscr{S}$. That is, $(t, s) \rhd_{\text{diff}} (t, s')$ if and only if $s \rhd_{\text{diff}} s'$.

Theorem 9 (★). *The relation $\rhd_{\text{diff}}$ induces Diffusion.*

The relation $\rhd_{\text{diff}}$ reveals the decision criteria of Diffusion: If two sequences s and s' have a different maximum rank, Diffusion decides in favor of the sequence with smaller maximum rank (see (i)). If the two sequences have equal maximum rank, Diffusion decides in favor of the sequence for which the maximum rank appears less often (see (ii)). Thus, Diffusion overcomes BFD's shortcoming of selecting sequences with large edge ranks. For example, from $\{(1, 100), (1, 1, 2)\}$, Diffusion selects $(1, 1, 2)$ while BFD selects $(1, 100)$. Moreover, in contrast to DFD, Diffusion

[7] Besides the fact that Colley, Grandi, and Novaro (2020) define their rule for a different setting, they also treat abstaining voters differently as abstentions can be delegated (see also Footnote 8).

cannot be tricked into selecting a sequence with large edge ranks at the end. For example, Diffusion selects (2) from $\{(1,100),(2)\}$ while DFD selects $(1,100)$. Having said this, the very last tie-breaking rule (see (iii)) can be argued to be slightly unnatural as it only compares the parts of s and s' up to the first appearance of the maximum rank. For example, this leads to $(1,5,4,4,4,4) \rhd_{\text{diff}} (2,5)$.

Inspired by this, we define *Leximax*, a delegation rule that shares Diffusion's desirable properties but avoids its artificial tie-breaking. We define the function σ, taking as input a rank sequence and sorting the ranks of the sequence in non-increasing order. For instance, $\sigma((1,3,4,3)) = (4,3,3,1)$.

Leximax: The delegation rule induced by the linear order $\rhd_\sigma$ over $\mathscr{S}$ defined as follows: For sequences $s, s' \in \mathscr{S}$ let $s \rhd_\sigma s'$ if (i) $\sigma(s) \rhd_{\text{lex}} \sigma(s')$, or (ii) $\sigma(s) = \sigma(s')$ and $s \rhd_{\text{lex}} s'$.

In the example in Figure 1, Leximax selects (a,b,c,d,j) for voter a. With the help of Theorem 6 we can show:

Corollary 10 (★). *Leximax is confluent.*

3.3 Branching Rules

We can also view the output of a confluent delegation rule as a special *directed forest* (aka *branching*) in the reduced graph $\bar{G} = (D \cup C, \bar{E})$. More precisely, we call $B \subseteq \bar{E}$ a *C-branching in $\bar{G}$* if B is acyclic and $|\delta_B^+(v)| = 1$ for all $v \in D$. For any confluent rule f, the set $\bigcup_{v \in D} f(v)$ is a C-branching in $\bar{G}$ and this encodes all selected paths.

Considering this interpretation, it seems natural to define delegation rules that optimize directly over the set of C-branchings, as for example selecting one with minimum sum of ranks. It is important to stress that this approach inherently comes with a different perspective on the problem. Namely, it gives equal importance to each of the edges instead of each of the selected paths. Under the premise that each voter is equally important, this is compatible with the assumption that voters care only about the first edge of their path. Though this is different from the approaches presented in the previous sections, this can be a valid view as the voters express their preferences explicitly only over their outgoing edges. Hence, any preferences over paths are "inherited" from other voters' preferences. Below we define BordaBranching, a natural variant of MinSum. Such a branching can be found by using a linear program for solving the *min-cost arborescence problem* (e.g., Korte and Vygen 2012).

BordaBranching: Selects a C-branching B in $\bar{G}$ that minimizes $\sum_{e \in B} r(e)$.

In order for BordaBranching to be resolute, we need to define a tie-breaking rule. We later use a *priority order* tie-breaking (formalized in the proof of Proposition 14). An example shows that BordaBranching is not a sequence rule: Voters v_1 and v_2 have each other as their first choice and casting voters w_1, w_2 as their second choice, respectively. Then, $\mathcal{S}_{v_1} = \mathcal{S}_{v_2} = \{(1,2),(2)\}$ and BordaBranching assigns sequence $(1,2)$ to one voter and (2) to the other.

The above approach is reminiscent of the Borda rule in classical social choice theory, as for every branching the method sums up the position of the branching's edges in the corresponding voters' rankings and compares these scores.

Interestingly, one can also apply an approach in the spirit of Condorcet and consider pairwise majority comparisons between branchings (Kavitha et al. 2021). Lifting the delegation preferences of a voter $v \in D$ to preference relation $\succ_v$ over C-branchings in a straightforward way (by comparing the ranks of v's outgoing edges in the branchings), define the *majority margin* between two C-branchings B and B' as

$$\Delta(B,B') = |\{v \in D \mid B \succ_v B'\}| - |\{v \in D \mid B' \succ_v B\}|.$$

A C-branching B is called *popular* if $\Delta(B,B') \geq 0$ for all B'. It follows from Kavitha et al. (2021) that a popular branching in our setting need not exist. They also define the *unpopularity margin* of B as $\mu(B) = \max_{B'} \Delta(B',B)$. Note that $\mu(B) \geq 0$ for all B and $\mu(B) = 0$ iff B is popular.

In our experiments, we evaluate branchings returned by confluent delegation rules by computing their unpopularity margin via a linear program (Kavitha et al. 2021). Surprisingly, we find that BordaBranching returns a popular branching in most instances.

4 Axiomatic Analysis

In this section, we revisit an axiom that was studied in the literature (*guru-participation*) and we formalize a desideratum whose importance was emphasized by practitioners (*copy-robustness*). We also provide axiomatic characterizations of the sequence rules DFD and BFD.

We define the *relative voting weight* $\omega_f(G,r,c)$ of a casting voter $c \in C$ that results from applying a delegation rule f to an instance (G,r) as

$$\omega_f(G,r,c) = \frac{|\{d \in D \mid f(G,r,d) \text{ ends in } c\}| + 1}{|C| + |D|}.$$

We use the *relative* voting weight in order to cope with multiple elections with distinct numbers of non-isolated voters.

4.1 Guru-Participation

Guru-participation was introduced by Kotsialou and Riley (2020), and a similar axiom has been suggested by Behrens and Swierczek (2015). The axiom demands that a casting voter should not be penalised for being the representative (also called *guru*) of a delegating voter. More precisely, Kotsialou and Riley (2020) consider a model that also comprises voting on a binary issue, which is decided by majority rule. They say that a casting voter c is "penalised" for being the representative of a delegating voter v, if c would prefer the outcome of the election in which, all other things being equal, v abstains. As our model captures the delegation phase only, we need to adapt the axiom to our setting. In the full version of this paper (Brill et al. 2021) we show that our version implies theirs, and, under a very mild assumption on the delegation rule, the two axioms are equivalent.

Definition 11. *A delegation rule f satisfies* guru-participation *if the following holds for every instance (G,r): If $v \in D$ and $f(G,r,v)$ ends in c, then*

$$\omega_f(G,r,u) \leq \omega_f(G',r,u) \text{ for all } u \in C \setminus \{c\},$$

where $G' = (C' \cup D' \cup I', E')$ is the graph derived from $G = (C \cup D \cup I, E)$ by setting $E' = E \setminus \delta_G^+(v)$ and $C' = C$. In particular, this implies that $\omega_f(G,r,c) \geq \omega_f(G',r,c)$.

Kotsialou and Riley (2020) showed that DFD violates guru-participation while BFD satisfies it. We generalize the latter statement to all confluent sequence rules.

Proposition 12 (★)**.** *Every confluent sequence rule satisfies guru-participation.*

As a consequence, three more rules satisfy the axiom.[8]

Corollary 13. *MinSum, Diffusion, and Leximax satisfy guru-participation.*

We show that the same holds for BordaBranching.

Proposition 14 (★)**.** *There exists a tie-breaking rule for which BordaBranching satisfies guru-participation.*

4.2 Copy-Robustness

The issue motivating the next axiom was brought up by Behrens and Swierczek (2015), who are part of the developing team behind *LiquidFeedback*. Consider a delegating voter v who is assigned a path of length one, i.e., this voter has a direct connection to its representative, which we call c. Behrens and Swierczek (2015) argue that there is the threat of a *copy-manipulation*: Using communication channels outside the liquid democracy system, these two voters can arrange that voter v acts as a casting voter by copying c's vote. If this manipulation leads to a different joint voting weight of v and c, then the underlying delegation rule is not *copy-robust*.[9] We formalize this property below.

Definition 15. *A delegation rule f is* copy-robust *if the following holds for every instance (G, r): If $v \in D$ such that $f(G, r, v)$ is of length one and ends in $c \in C$, then*

$$\omega_f(G, r, c) = \omega_f(G', r, c) + \omega_f(G', r, v),$$

where $G' = (C' \cup D' \cup I', E')$ is derived from $G = (C \cup D \cup I, E)$ by setting $E' = E \setminus \delta^+_G(v)$ and $C' = C \cup \{v\}$.

We give a consequence of copy-robustness that illustrates how limiting this property is for sequence rules.

Proposition 16 (★)**.** *If the sequence rule induced by $\rhd$ is copy-robust, then, for all $x \in \mathbb{N}$ and for any comparable sequences s and s', it holds that $(s, x) \rhd s' \Leftrightarrow s \rhd s'$.*

The restrictive nature of copy-robustness becomes even more apparent in the following impossibility result.

Proposition 17 (★)**.** *No sequence rule is both confluent and copy-robust.*

Hence, none of BFD, MinSum, Diffusion, and Leximax is copy-robust. On the other hand, Behrens and Swierczek (2015) have implicitly shown that the non-confluent sequence rule DFD satisfies the axiom (we give a proof in Theorem 20). Our next result shows that the incompatibility between confluence and copy-robustness is indeed restricted to sequences rules: We prove that BordaBranching is copy-robust. We use the same tie-breaking as in Proposition 14.

Proposition 18 (★)**.** *There exists a tie-breaking rule for which BordaBranching satisfies copy-robustness.*

4.3 Characterizations

Finally, we give axiomatic characterizations of DFD and BFD. To this end, we define two additional properties.

Definition 19. *A sequence rule f is* weakly lexicographic *if for two comparable sequences s and s' with $|s| = |s'|$ that only differ in their last rank value, it holds that $s \rhd s' \Leftrightarrow s \rhd_{lex} s'$. The rule f is* strongly lexicographic *if the same holds for any two comparable sequences of equal length.*

Every reasonable sequence rule should be weakly lexicographic, as this can also be seen as avoiding the selection of "Pareto dominated" sequences. The stronger version, however, is technical in nature and has no normative appeal.

Theorem 20 (★)**.** *DFD is the only sequence rule that is weakly lexicographic and copy-robust.*

Theorem 21 (★)**.** *BFD is the only sequence rule that is confluent and strongly lexicographic.*

The full version (Brill et al. 2021) contains a characterization of Diffusion and an overview of our axiomatic results.

5 Experimental Evaluation

To complement our axiomatic results, we conducted an extensive experimental study in order to compare delegation rules with respect to quantitative criteria. To the best of our knowledge, there are no real-world data sets on liquid democracy with ranked delegations, nor do there exist established data generation methods for this setting. Thus, we developed three distinct *methods for generating instances* of our setting. In all cases, we transform a (directed or undirected) *base graph* $H = (V_H, E_H)$ to an instance (G, r) of our setting, with $G = (C \cup D \cup I, E)$. While it is always the case that $V_H = C \cup D \cup I$, the set of casting voters $C \subseteq V_H$, the edges of the graph $E \subseteq E_H$, and the rank function r are selected in a random process specified by the generation method. Hence, one base graph H paired with one of the three methods leads to a wide range of instances. For each method, we use multiple base graphs coming from synthetic and real-world networks (such as partial networks of *Facebook* and *Twitter*). We let n (respectively, m) denote the number of nodes (respectively, edges) of the graph G.

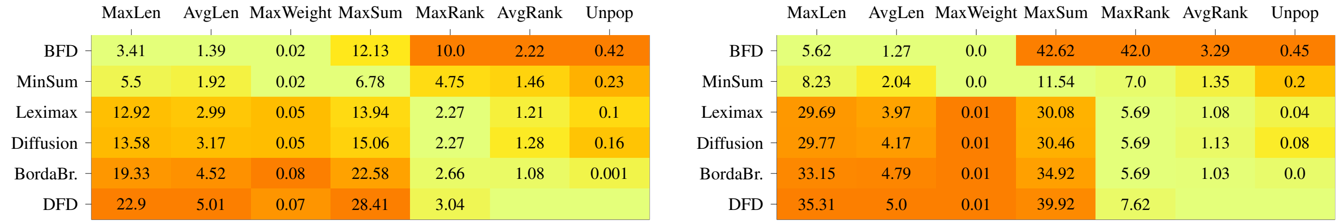
Our experimental setup is described in detail in (Brill et al. 2021), where we also present the results for different generation methods, datasets, and parameters. Our code can be found at https://github.com/TheoDlmz/rankeddelegation.

Data Generation Our three generation methods apply to different types of base graphs: unweighted undirected networks, unweighted directed networks, and weighted directed networks. We describe the first method below, and describe the other methods in the full version.[10]

[8] Colley, Grandi, and Novaro (2020) show that their *unravelling procedures* (two of which reduce to Diffusion up to the fact that abstentions can be delegated) do *not* satisfy guru-participation. This is an artifact of their treatment of abstaining voters.

[9] Behrens and Swierczek (2015) consider an even stronger requirement, according to which the final vote count of a binary election needs to remain equal (just as Kotsialou and Riley, 2020, they capture the voting phase in their model). We remark that our positive result also holds for this stronger version of the axiom.

[10] In our method for unweighted directed networks, which is inspired by Gölz et al. (2018), voters are more likely to delegate to



	MaxLen	AvgLen	MaxWeight	MaxSum	MaxRank	AvgRank	Unpop
BFD	3.41	1.39	0.02	12.13	10.0	2.22	0.42
MinSum	5.5	1.92	0.02	6.78	4.75	1.46	0.23
Leximax	12.92	2.99	0.05	13.94	2.27	1.21	0.1
Diffusion	13.58	3.17	0.05	15.06	2.27	1.28	0.16
BordaBr.	19.33	4.52	0.08	22.58	2.66	1.08	0.001
DFD	22.9	5.01	0.07	28.41	3.04		

(a) Synthetic data, averaged over 1000 instances ($n = 1000$, $\Delta = 4$, $p_c = 0.2$, $\alpha = 2$)

	MaxLen	AvgLen	MaxWeight	MaxSum	MaxRank	AvgRank	Unpop
BFD	5.62	1.27	0.0	42.62	42.0	3.29	0.45
MinSum	8.23	2.04	0.0	11.54	7.0	1.35	0.2
Leximax	29.69	3.97	0.01	30.08	5.69	1.08	0.04
Diffusion	29.77	4.17	0.01	30.46	5.69	1.13	0.08
BordaBr.	33.15	4.79	0.01	34.92	5.69	1.03	0.0
DFD	35.31	5.0	0.01	39.92	7.62		

(b) Real-world data using Facebook network ($n = 63,731$ and $m = 817,035$), averaged over 10 instances ($p_c = 0.2$, $\alpha = 1$)

Figure 2: Evaluation of delegation rules with respect to several quantities for real and synthetic unweighted undirected networks.

When the base graph $H = (V_H, E_H)$ is undirected and unweighted, we assume edges to represent "friendship" links between the nodes. We quantify the "strength" of a friendship between two adjacent nodes v and w by the number of common friends, i.e., $\lambda(v,w) = |\delta_H(v) \cap \delta_H(w)|$, where $\delta_H(\cdot)$ is the set of neighbors of a node in H. We first choose $C \subseteq V_H$ by selecting each voter independently with probability $p_c \in [0,1]$. Then, for every $v \in V \setminus C$, we insert edges in $\delta_H(v)$ one by one as out-going edges for v in G where the probability of selecting the edge $\{v,w\} \in E_H$ is proportional to $(1+\lambda(v,w))^\alpha$, and $\alpha > 0$ is a constant. The ranking r among edges is defined by the order of selection.

The real-world network we used for this method is a subgraph of the Facebook network (Viswanath et al. 2009). For synthetic networks, the input network H is generated by the standard Erdős–Rényi model $G(n,p)$, where we chose the edge probability $p \in [0,1]$ such that the average number of edges per voter is equal to Δ.

Evaluation Metrics We implemented all considered delegation rules and evaluated the output of the delegation rules on various metrics: The maximum rank found on any delegation path (*MaxRank*), the maximum and average length of delegations paths (*MaxLen* and *AvgLen*), and the maximum sum of ranks of a delegation path (*MaxSum*). We also computed *MaxWeight*, the maximum value of the *relative voting weight* $\omega_f(G,r,c)$ of a casting voter c, as a measure of the balancedness of the distribution of voting weight. (This value plays a crucial role in the analysis of Gölz et al. 2018.) Finally, for confluent rules, we computed the average rank of outgoing edges (*AvgRank*) and the unpopularity margin of the selected branchings, divided by the number of non-isolated voters (*Unpop*). For all of these metrics, small values are desirable and correspond to light colors in our tables.

Experimental Results Figure 2 presents the results for unweighted undirected base networks. These results are representative also for other generation methods and data sets.

The results indicate that the rules can be roughly aligned along a spectrum: On the one extreme, BFD leads to a good weight distribution (i.e., low average values of MaxWeight), but high (maximum and average) ranks. On the other extreme, DFD and BordaBranching perform better on the rank metrics but often lead to unbalanced weight distributions and long delegation paths. Other sequence rules fall between these extremes, with MinSum closer to BFD, and Leximax and Diffusion closer to DFD and BordaBranching. MinSum in particular seems to strike an attractive balance among most of the considered evaluation criteria.

Other noteworthy observations include that Leximax outperforms Diffusion on every metric (recall that we introduced the former as a simplification of the latter) and that BordaBranching very often produces popular branchings (for example, this is the case in 93% of instances for the synthetic data set in Figure 2(a)). On average, BordaBranching mostly outperforms DFD, and it also has the advantage of being a confluent rule that moreover satisfies guru-participation and copy-robustness.

6 Discussion

Building upon a graph-theoretic model, we explored the rich space of delegation rules and introduced novel rules that axiomatically and empirically outperform existing ones.

Our experiments revealed a gap between rules such as BFD and MinSum on the one side, and rules such as Leximax on the other. This gap can be filled by defining *weighted* sequence rules: For an increasing function $w : \mathbb{N}_{\geq 1} \to \mathbb{R}^+$, such a rule orders sequences by $\sum_i w(s_i)$. Our axiomatic results for MinSum can be generalized to this class.

In future work, it might be desirable to take preferences over delegation *paths* (rather than only edges) into account. These could then be lifted to preferences over branchings, and approaches like those described in Section 3.3 could still be used to define confluent delegation rules.

While some branching rules are inherently non-neutral, *randomized delegation rules* have the potential to avoid this issue by selecting probability distributions over outgoing edges (Brill 2018). Going beyond deterministic delegations would also raise the necessity for new axioms.

Our experiments indicate a trade-off between minimizing the quantities *unpopularity* and *average rank* on the one hand, and *maximum length* and *maximum weight* on the other hand. It would be interesting to formalize this trade-off by finding worst-case bounds and guarantees.

We also suggest a comparison between outcomes of liquid democracy with and without ranked delegations.

nodes with a high indegree. This is reminiscent of the *preferential attachment* model (Barabási and Albert 1999). For weighted directed networks, we interpret weights as "trust" and rank-order potential delegation edges by decreasing weight.

Acknowledgments

We would like to thank Jan Behrens, Axel Kistner, Andreas Nitsche, and Björn Swierczek from the Association for Interactive Democracy for helpful discussions. This work was supported by the Deutsche Forschungsgemeinschaft under grant BR 4744/2-1, by the Austrian Science Fund (FWF), project P31890, and by the Norwegian Research Council, project nr. 302203.

References

Barabási, A.-L.; and Albert, R. 1999. Emergence of Scaling in Random Networks. *Science*, 286(5439): 509–512.

Behrens, J. 2017. The origins of liquid democracy. *The Liquid Democracy Journal*, 5: 7–17.

Behrens, J.; Kistner, A.; Nitsche, A.; and Swierczek, B. 2014. *The Principles of LiquidFeedback*. Interaktive Demokratie e. V. Berlin.

Behrens, J.; and Swierczek, B. 2015. Preferential Delegation and the Problem of Negative Voting Weight. *The Liquid Democracy Journal*, 3: 6–34.

Blum, C.; and Zuber, C. I. 2016. Liquid democracy: Potentials, problems, and perspectives. *Journal of Political Philosophy*, 24(2): 162–182.

Boldi, P.; Bonchi, F.; Castillo, C.; and Vigna, S. 2011. Viscous democracy for social networks. *Communications of the ACM*, 54(6): 129–137.

Brill, M. 2018. Interactive Democracy. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) Blue Sky Ideas track*, 1183–1187. IFAAMAS.

Brill, M.; and Talmon, N. 2018. Pairwise Liquid Democracy. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, 137–143. IJCAI.

Brill, M.; Delemazure, T.; George, A.-M.; Lackner, M.; and Schmidt-Krapelin, U. 2021. Liquid Democracy with Ranked Delegations. Technical Report, arXiv:2112.07509.

Chen, J.; Rajaraman, R.; and Sundaram, R. 2006. Meet and merge: Approximation algorithms for confluent flows. *Journal of Computer and System Sciences*, 72(3): 468–489.

Christoff, Z.; and Grossi, D. 2017. Binary Voting with Delegable Proxy: An Analysis of Liquid Democracy. In *Proceedings of the 16th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 134–150.

Colley, R.; Grandi, U.; and Novaro, A. 2020. Smart Voting. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, 1734 – 1740. IJCAI.

Dodgson, C. L. 1884. *The Principles of Parliamentary Representation*. Harrison and Sons.

Ford, B. 2002. Delegative Democracy. http://www.brynosaurus.com/deleg/deleg.pdf. Accessed: 2022-04-23.

Gölz, P.; Kahng, A.; Mackenzie, S.; and Procaccia, A. D. 2018. The Fluid Mechanics of Liquid Democracy. In *Proceedings of the 14th International Workshop on Internet and Network Economics (WINE)*, 188–202. Springer.

Hardt, S.; and Lopes, L. 2015. Google Votes: A Liquid Democracy Experiment on a Corporate Social Network. Technical report, Technical Disclosure Commons.

Kahng, A.; Mackenzie, S.; and Procaccia, A. 2021. Liquid Democracy: An Algorithmic Perspective. *Journal of Artificial Intelligence Research*, 70: 1223–1252.

Kavitha, T.; Király, T.; Matuschke, J.; Schlotter, I.; and Schmidt-Kraepelin, U. 2021. Popular Branchings and Their Dual Certificates. *Mathematical Programming*, 1–29.

Korte, B.; and Vygen, J. 2012. *Combinatorial Optimization*. Springer-Verlag, 5th edition.

Kotsialou, G.; and Riley, L. 2020. Incentivising Participation in Liquid Democracy with Breadth First Delegation. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 638–644. IFAAMAS.

Miller, J. C. 1969. A program for direct and proxy voting in the legislative process. *Public choice*, 7(1): 107–113.

Paulin, A. 2020. An Overview of Ten Years of Liquid Democracy Research. In *Proceedings of the 21st Annual International Conference on Digital Government Research (DGO)*, 116–121. ACM.

Tullock, G. 1967. *Towards a Mathematics of Politics*. University of Michigan Press.

Valsangiacomo, C. 2021. Political Representation in Liquid Democracy. *Frontiers in Political Science*, 3: 591853.

Viswanath, B.; Mislove, A.; Cha, M.; and Gummadi, K. P. 2009. On the Evolution of User Interaction in Facebook. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Social Networks (WOSN)*, 37–42. ACM.