The Fluid Mechanics of Liquid Democracy

Paul Gölz(✉), Anson Kahng, Simon Mackenzie, and Ariel D. Procaccia

Computer Science Department, Carnegie Mellon University, Pittsburgh, USA
{pgoelz,akahng,simonm,arielpro}@cs.cmu.edu

Abstract. *Liquid democracy* is the principle of making collective decisions by letting agents transitively delegate their votes. Despite its significant appeal, it has become apparent that a weakness of liquid democracy is that a small subset of agents may gain massive influence. To address this, we propose to change the current practice by allowing agents to specify multiple delegation options instead of just one. Much like in nature, where—fluid mechanics teaches us—liquid maintains an equal level in connected vessels, so do we seek to control the flow of votes in a way that balances influence as much as possible. Specifically, we analyze the problem of choosing delegations to approximately minimize the maximum number of votes entrusted to any agent, by drawing connections to the literature on confluent flow. We also introduce a random graph model for liquid democracy, and use it to demonstrate the benefits of our approach both theoretically and empirically.

1 Introduction

Liquid democracy is a potentially disruptive approach to democratic decision making. As in direct democracy, agents can vote on every issue by themselves. Alternatively, however, agents may delegate their vote, i.e., entrust it to any other agent who then votes on their behalf. Delegations are transitive; for example, if agents 2 and 3 delegate their votes to 1, and agent 4 delegates her vote to 3, then agent 1 would vote with the weight of all four agents, including herself. Just like representative democracy, this system allows for separation of labor, but provides for stronger accountability: Each delegator is connected to her transitive delegate by a path of personal trust relationships, and each delegator on this path can withdraw her delegation at any time if she disagrees with her delegate's choices.

Although the roots of liquid democracy can be traced back to the work of Miller [15], it is only in recent years that it has gained recognition among practitioners. Most prominently, the German Pirate Party adopted the platform *LiquidFeedback* for internal decision-making in 2010. At the highest point, their installation counted more than 10 000 active users [12]. More recently, two parties—the Net Party in Argentina, and Flux in Australia—have run in national elections on the promise that their elected representatives would vote according

G. Christodoulou and T. Harks (Eds.): WINE 2018, LNCS 11316, pp. 188–202, 2018.
https://doi.org/10.1007/978-3-030-04612-5_13

to decisions made via their respective liquid-democracy-based systems. Although neither party was able to win any seats in parliament, their bids enhanced the promise and appeal of liquid democracy.

However, these real-world implementations also exposed a weakness in the liquid democracy approach: Certain individuals, the so-called *super-voters*, seem to amass enormous weight, whereas most agents do not receive any delegations. In the case of the Pirate Party, this phenomenon is illustrated by an article in Der Spiegel, according to which one particular super-voter's "vote was like a decree," even though he held no office in the party. As Kling et al. [12] describe, super-voters were so controversial that "the democratic nature of the system was questioned, and many users became inactive." Besides the negative impact of super-voters on perceived legitimacy, super-voters might also be more exposed to bribing. Although delegators can retract their delegations as soon as they become aware of suspicious voting behavior, serious damage might be done in the meantime. Furthermore, if super-voters jointly have sufficient power, they might find it more efficient to organize majorities through deals between super-voters behind closed doors, rather than to try to win a broad majority through public discourse. Finally, recent work by Kahng et al. [11] indicates that, even if delegations go only to more competent agents, a high concentration of power might still be harmful for social welfare, by neutralizing benefits corresponding to the Condorcet Jury Theorem.

While all these concerns suggest that the weight of super-voters should be limited, the exact metric to optimize for varies between them and is often not even clearly defined. For the purposes of this paper, we choose to minimize the weight of the heaviest voter. As is evident in the Spiegel article, the weight of individual voters plays a direct role in the perception of super-voters. But even beyond that, we are confident that minimizing this measure will lead to substantial improvements across all presented concerns.

Just how can the maximum weight be reduced? One approach might be to restrict the power of delegation by imposing caps on the weight. However, as argued by Behrens et al. [3], delegation is always possible by coordinating outside of the system and copying the desired delegate's ballot. Pushing delegations outside of the system would not alleviate the problem of super-voters, just reduce transparency. Therefore, we instead adopt a voluntary approach: If agents are considering multiple potential delegates, all of whom they trust, they are encouraged to leave the decision for one of them to a centralized mechanism. With the goal of avoiding high-weight agents in mind, our research challenge is twofold: *First, investigate the algorithmic problem of selecting delegations to minimize the maximum weight of any agent, and, second, show that allowing multiple delegation options does indeed provide a significant reduction in the maximum weight compared to the status quo.*

Put another (more whimsical) way, we wish to design liquid democracy systems that emulate the *law of communicating vessels*, which asserts that liquid will find an equal level in connected containers.

1.1 Our Approach and Results

We formally define our problem in Sect. 2. In general, our problem is closely related to minimizing congestion for confluent flow as studied by Chen et al. [5]. Not only does this connection suggest an optimal algorithm based on mixed integer linear programming, but we also get a polynomial-time $(1 + \log |V|)$-approximation algorithm, where V is the set of direct voters.[1] In addition, we show that approximating our problem to within a factor of $\frac{1}{2} \log_2 |V|$ is NP-hard.

In Sect. 3, to evaluate the benefits of allowing multiple delegations, we propose a probabilistic model for delegation behavior—inspired by the well-known *preferential attachment* model [2]. In a certain class of parameter settings, moving from single delegations to two delegation options per agent decreases the maximum weight doubly exponentially. Our analysis draws on a phenomenon called the *power of choice* that can be observed in many different load balancing models.

In Sect. 4, we show through simulations that our approach continues to outperform classical preferential attachment in more general parameter settings. These improvements in terms of maximum weight persist even if just some fraction of delegators gives two options while the others specify a single delegate. Finally, we find that the approximation algorithm and even a greedy heuristic lead to close-to-optimal maximum weights in our model.

1.2 Related Work

Kling et al. [12] conduct an empirical investigation of the existence and influence of super-voters. The analysis is based on daily data dumps, from 2010 until 2013, of the German Pirate Party installation of LiquidFeedback. As noted above, Kling et al. find that super-voters exist, and have considerable power. The results do suggest that super-voters behave responsibly, as they "do not fully act on their power to change the outcome of votes, and they vote in favour of proposals with the majority of voters in many cases." Of course, this does not contradict the idea that a balanced distribution of power would be desirable.

There are only a few papers that provide theoretical analyses of liquid democracy [7,9,11]. We would like to stress the differences between our approach and the one adopted by Kahng et al. [11]. They consider binary issues in a setting with an objective ground truth, i.e., there is one "correct" outcome and one "incorrect" outcome. In this setting, voters are modeled as biased coins that each choose the correct outcome with an individually assigned probability, or *competence level*. The authors examine whether liquid democracy can increase the probability of making the right decision over direct democracy by having less competent agents delegate to more competent ones. By contrast, our work is completely independent of the (strong) assumptions underlying the results of Kahng et al. In particular, our approach is agnostic to the final outcome of the voting process, does not assume access to information that would be inaccessible in practice, and is compatible with any number of alternatives and choice of

[1] Throughout this paper, let log denote the natural logarithm.

voting rule used to aggregate votes. In other words, the goal is not to use liquid democracy to promote a particular outcome, but rather to adapt the process of liquid democracy such that more voices will be heard.

2 Algorithmic Model and Results

Let us consider a delegative voting process where agents may specify multiple potential delegations. This gives rise to a directed graph, whose nodes represent agents and whose edges represent potential delegations. A distinguished subset of nodes corresponds to agents who have voted directly, the *voters*. Since voters forfeit the right to delegate, the voters are a subset of the sinks of the graph. We call all non-voter agents *delegators*.

Each agent has an inherent voting weight of 1. When the delegations will have been resolved, the weight of every agent will be the sum of weights of her delegators plus her inherent weight. We aim to choose a delegation for every delegator in such a way that the maximum weight of any voter is minimized.

This task closely mirrors the problem of congestion minimization for confluent flow (with infinite edge capacity): There, a flow network is also a finite directed graph with a distinguished set of graph sinks, the *flow sinks*. Every node has a non-negative *demand*. If we assume unit demand, this demand is 1 for every node. Since the flow is *confluent*, for every non-sink node, the algorithm must pick exactly one outgoing edge, along which the flow is sent. Then, the *congestion* at a node n is the sum of congestions at all nodes who direct their flow to n plus the demand of n. The goal in congestion minimization is to minimize the maximum congestion at any flow sink.

In spite of the similarity between confluent flow and resolving potential delegations, the two problems differ when a node has no path to a voter/flow sink. In confluent flow, the result would simply be that no flow exists. In our setting however, this situation can hardly be avoided. If, for example, several friends assign all of their potential delegations to each other, and if all of them rely on the others to vote, their weight cannot be delegated to any voter. Our mechanism cannot simply report failure as soon as a small group of voters behaves in an unexpected way. Thus, it must be allowed to leave these votes unused. At the same time, of course, our algorithm should not exploit this power to decrease the maximum weight, but must primarily maximize the number of utilized votes. We formalize these issues in the following section.

2.1 Problem Statement

All graphs $G = (N, E)$ mentioned in this section will be finite and directed. Furthermore, they will be equipped with a set of voters $V \subseteq \mathit{sinks}(G)$.

Some of these graphs represent situations in which all delegations have already been resolved and in which each vote reaches a voter: We call a graph (N, E) with voters V a *delegation graph* if it is acyclic, its sinks are exactly the set V, and every other vertex has outdegree one. In such a graph, define the *weight* $w(n)$ of a node $n \in N$ as

$$w(n) := 1 + \sum_{(m,n) \in E} w(m).$$

This is well-defined because E is a well-founded relation on N.

Resolving the delegations of a graph G with voters V can now be described as the MinMaxWeight problem: Among all delegation subgraphs (N', E') of G with voting vertices V of maximum $|N'|$, find one that minimizes the maximum weight of the voting vertices.

2.2 Connections to Confluent Flow

We recall definitions from the flow literature as used by Chen et al. [5]. We slightly simplify the exposition by assuming unit demand at every node.

Given a graph (N, E) with V, a *flow* is a function $f : E \to \mathbb{R}_{\geq 0}$. For any node n, set $in(n) := \sum_{(m,n) \in E} f(m, n)$ and $out(n) := \sum_{(n,m) \in E} f(n, m)$. At every node $n \in N \setminus V$, a flow must satisfy *flow conservation*: $out(n) = 1 + in(n)$. The *congestion* at any node n is defined as $1 + in(n)$. A flow is *confluent* if every node has at most one outgoing edge with positive flow. We define MinMaxCongestion as the problem of finding a confluent flow on a given graph such that the maximum congestion is minimized.

In the full version of this paper [8], we give translations between instances of MinMaxWeight and MinMaxCongestion that preserve the optimization objective value.

2.3 Algorithms

These translations allow us to apply algorithms—even approximation algorithms—for MinMaxCongestion to our MinMaxWeight problem, that is, we can reduce the latter problem to the former.

Theorem 1. *Let $\mathcal{A}$ be an algorithm for* MinMaxCongestion *with approximation ratio $c \geq 1$. Let $\mathcal{A}'$ be an algorithm that, given (N, E) with V, runs $\mathcal{A}$ on the active subgraph, and translates the result into a delegation subgraph by eliminating all zero-flow edges. Then $\mathcal{A}'$ is a c-approximation algorithm for* MinMaxWeight.

We relegate the proof to the full version [8]. Note that Theorem 1 works for $c = 1$, i.e., even for exact algorithms. Therefore, it is possible to solve MinMaxWeight by adapting a standard mixed integer linear programming (MILP) formulation of MinMaxFlow.

Since this algorithm is based on solving an NP-hard problem, it might be too inefficient for typical use cases of liquid democracy with many participating agents. Fortunately, it might be acceptable to settle for a slightly non-optimal maximum weight if this decreases computational cost. To our knowledge, the best polynomial approximation algorithm for MinMaxCongestion is due to Chen et al. [5] and achieves an approximation ratio of $1 + \log |V|$. Their algorithm starts by computing the optimal solution to the splittable-flow version of the problem, by solving a linear program. The heart of their algorithm is a non-trivial, deterministic rounding mechanism. This scheme drastically outperforms the natural, randomized rounding scheme, which leads to an approximation ratio of $\Omega(|N|^{1/4})$ with arbitrarily high probability [6].

2.4 Hardness of Approximation

In this section, we demonstrate the NP-hardness of approximating the MinMaxWeight problem to within a factor of $\frac{1}{2}\log_2 |V|$. On the one hand, this justifies the absence of an exact polynomial-time algorithm. On the other hand, this shows that the approximation algorithm is optimal up to a multiplicative constant.

Theorem 2. *It is NP-hard to approximate the* MinMaxWeight *problem to a factor of* $\frac{1}{2}\log_2 |V|$*, even when each node has outdegree at most* 2*.*

Again, the proof can be found in the full version [8]. Not surprisingly, we derive hardness via a reduction from MinMaxCongestion, i.e., a reduction in the opposite direction from the one given in Theorem 1. As shown by Chen et al. [5], approximating MinMaxCongestion to within a factor of $\frac{1}{2}\log_2 |V|$ is NP-hard. However, in our case, nodes have unit demands. Moreover, we are specifically interested in the case where each node has outdegree at most 2, as in practice we expect outdegrees to be very small, and this case plays a special role in the following section.

3 Probabilistic Model and Results

Our generalization of liquid democracy to multiple potential delegations aims to decrease the concentration of weight. Accordingly, the success of our approach should be measured by its effect on the maximum weight in real elections. Since, at this time, we do not know of any available datasets,[2] we instead propose a probabilistic model for delegation behavior, which can serve as a credible proxy. Our model builds on the well-known preferential attachment model, which generates graphs possessing typical properties of social networks.

The evaluation of our approach will be twofold: In Sects. 3.2 and 3.3, for a certain choice of parameters in our model, we establish a striking separation between traditional liquid democracy and our system. In the former case, the maximum weight at time t is $\Omega(t^{\beta})$ for a constant β with high probability,

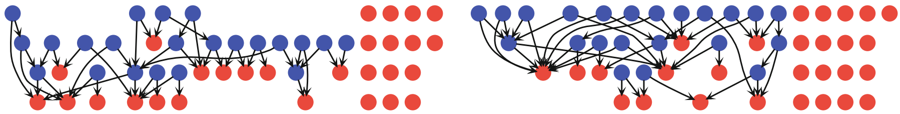
[2] There is one relevant dataset that we know of, which was analyzed by Kling et al. [12]. However, due to stringent privacy constraints, the data privacy officer of the German Pirate Party was unable to share this dataset with us.

whereas in the latter case, it is in $\mathcal{O}(\log \log t)$ with high probability, even if each delegator only suggests two options. For other parameter settings, we empirically corroborate the benefits of our approach in Sect. 4.

3.1 The Preferential Delegation Model

Many real-world social networks have degree distributions that follow a power law [13,16]. Additionally, in their empirical study, Kling et al. [12] observed that the weight of voters in the German Pirate Party was "power law-like" and that the graph had a very unequal indegree distribution. In order to meld the previous two observations in our liquid democracy delegation graphs, we adapt a standard preferential attachment model [2] for this specific setting. On a high level, our *preferential delegation* model is characterized by three parameters: $0 < d < 1$, the probability of delegation; $k \geq 1$, the number of delegation options from each delegator; and $\gamma \geq 0$, an exponent that governs the probability of delegating to nodes based on current weight.

At time $t = 1$, we have a single node representing a single voter. In each subsequent time step, we add a node for agent i and flip a biased coin to determine her delegation behavior. With probability d, she delegates to other agents. Else, she votes independently. If i does not delegate, her node has no outgoing edges. Otherwise, add edges to k many i.i.d. selected, previously inserted nodes, where the probability of choosing node j is proportional to $(\textit{indegree}(j) + 1)^{\gamma}$. Note that this model might generate multiple edges between the same pair of nodes, and that all sinks are voters. Figure 1 shows example graphs for different settings of γ.



(a) $k = 2, d = 0.5, \gamma = 0$ (b) $k = 2, d = 0.5, \gamma = 1$

Fig. 1. Example graphs generated by the preferential delegation model.

In the case of $\gamma = 0$, which we term *uniform delegation*, a delegator is equally likely to attach to any previously inserted node. Already in this case, a "rich-get-richer" phenomenon can be observed, i.e., voters at the end of large networks of potential delegations will likely see their network grow even more. Indeed, a larger network of delegations is more likely to attract new delegators. In traditional liquid democracy, where $k = 1$ and all potential delegations will be realized, this explains the emergence of super-voters with excessive weight observed by Kling et al. [12]. We aim to show that for $k \geq 2$, the resolution of potential delegations can strongly outweigh these effects. In this, we profit from an effect known as the "power of two choices" in load balancing described by Azar et al. [1].

For $\gamma > 0$, the "rich-get-richer" phenomenon additionally appears at the degrees of nodes. Since the number of received potential delegations is a proxy for an agent's competence and visibility, new agents are more likely to attach to agents with high indegree. In total, this is likely to further strengthen the inherent inequality between voters. For increasing γ, the graph becomes increasingly flat, as a few super-voters receive nearly all delegations. This matches observations from the LiquidFeedback dataset [12] that "the delegation network is slowly becoming less like a friendship network, and more like a bipartite networks of super-voters connected to normal voters." The special case of $\gamma = 1$ corresponds to preferential attachment as described by Barabási and Albert [2].

The most significant difference we expect to see between graphs generated by the preferential delegation model and real delegation graphs is the assumption that agents always delegate to more senior agents. In particular, this causes generated graphs to be acyclic, which need not be the case in practice. It does seem plausible that the majority of delegations goes to agents with more experience on the platform. Even if this assumption should not hold, there is a second interpretation of our process if we assume—as do Kahng et al. [11]—that agents can be ranked by competence and only delegate to more competent agents. Then, we can think of the agents as being inserted in decreasing order of competence. When a delegator chooses more competent agents to delegate to, her choice would still be biased towards agents with high indegree, which is a proxy for popularity.

In our theoretical results, we focus on the cases of $k = 1$ and $k = 2$, and assume $\gamma = 0$ to make the analysis tractable. The parameter d can be chosen freely between 0 and 1. Note that our upper bound for $k = 2$ directly translates into an upper bound for larger k, since the resolution mechanism always has the option of ignoring all outgoing edges except for the two first. Therefore, to understand the effect of multiple delegation options, we can restrict our attention to $k = 2$. This crucially relies on $\gamma = 0$, where potential delegations do not influence the probabilities of choosing future potential delegations. Based on related results by Malyshkin and Paquette [14], it seems unlikely that increasing k beyond 2 will reduce the maximum weight by more than a constant factor.

3.2 Lower Bounds for Single Delegation ($k = 1$, $\gamma = 0$)

As mentioned above, we begin with a lower bound on the maximum weight for the case of uniform delegation and a single delegation option per delegator. Here and in the following, we say that a sequence $(\mathcal{E}_t)_t$ of events happens *with high probability* if $\mathbb{P}[\mathcal{E}_t] \to 1$ for $t \to \infty$. Our lower bound, whose proof is relegated to the full version [8], is the following:

Theorem 3. *In the preferential delegation model with $k = 1$, $\gamma = 0$, and $d \in (0, 1)$, with high probability, the maximum weight of any voter at time t is in $\Omega(t^{\beta})$, where $\beta > 0$ is a constant that depends only on d.*

3.3 Upper Bound for Double Delegation ($k = 2$, $\gamma = 0$)

Analyzing cases with $k > 1$ is considerably more challenging. One obstacle is that we do not expect to be able to incorporate optimal resolution of potential delegations into our analysis, because the computational problem is hard even when $k = 2$ (see Theorem 2). Therefore, we give a pessimistic estimate of optimal resolution via a greedy delegation mechanism, which we can reason about alongside the stochastic process. Clearly, if this stochastic process can guarantee an upper bound on the maximum weight with high probability, this bound must also hold if delegations are optimally resolved to minimize maximum weight.

In more detail, whenever a new delegator is inserted into the graph, the greedy mechanism immediately selects one of the delegation options. As a result, at any point during the construction of the graph, the algorithm can measure the weight of the voters. Suppose that a new delegator suggests two delegation options, to agents a and b. By following already resolved delegations, the mechanism obtains voters a^* and b^* such that a transitively delegates to a^* and b to b^*. The greedy mechanism then chooses the delegation whose voter currently has lower weight, resolving ties arbitrarily.

This situation is reminiscent of a phenomenon known as the "power of choice." In its most isolated form, it has been studied in the *balls-and-bins* model, for example by Azar et al. [1]. In this model, n balls are to be placed in n bins. In the classical setting, each ball is sequentially placed into a bin chosen uniformly at random. With high probability, the fullest bin will contain $\Theta(\log n / \log \log n)$ balls at the end of the process. In the choice setting, two bins are independently and uniformly selected for every ball, and the ball is placed into the emptier one. Surprisingly, this leads to an exponential improvement, where the fullest bin will contain at most $\Theta(\log \log n)$ balls with high probability.

We show that, at least for $\gamma = 0$ in our setting, this effect outweighs the "rich-get-richer" dynamic described earlier:

Theorem 4. *In the preferential delegation model with* $k = 2$*,* $\gamma = 0$*, and* $d \in (0, 1)$*, the maximum weight of any voter at time* t *is* $\log_2 \log t + \Theta(1)$ *with high probability.*

Due to space constraints, we defer the proof to the full version [8]. In our proof we build on work by Malyshkin and Paquette [14], who study the maximum *degree* in a graph generated by preferential attachment with the power of choice. In addition, we incorporate ideas by Haslegrave and Jordan [10].

4 Simulations

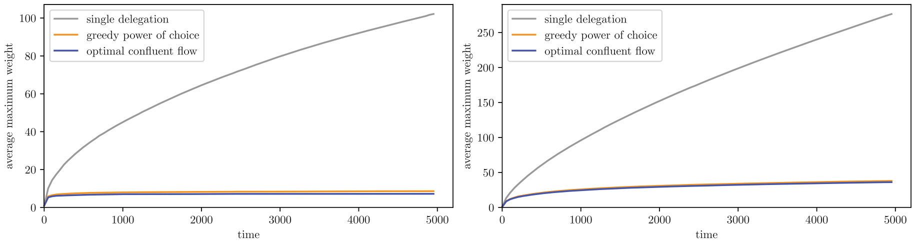
In this section, we present our simulation results, which support the two main messages of this paper: that allowing multiple delegation options significantly reduces the maximum weight, and that it is computationally feasible to resolve delegations in a way that is close to optimal.

Our simulations were performed on a MacBook Pro (2017) on MacOS 10.12.6 with a 3.1 GHz Intel Core i5 and 16 GB of RAM. All running times were measured with at most one process per processor core. Our simulation software is written in Python 3.6 using Gurobi 8.0.1 to solve MILPs. All of our simulation code is open-source and available at https://github.com/pgoelz/fluid.

4.1 Multiple vs. Single Delegations

For the special case of $\gamma = 0$, we have established a doubly exponential, asymptotic separation between single delegation ($k = 1$) and two delegation options per delegator ($k = 2$). While the strength of the separation suggests that some of this improvement will carry over to the real world, we still have to examine via simulation whether improvements are visible for realistic numbers of agents and other values of γ.

To this end, we empirically evaluate two different mechanisms for resolving delegations. First, we optimally resolve delegations by solving an MILP for confluent flow. Our second mechanism is the greedy "power of choice" algorithm used in the theoretical analysis and introduced in Sect. 3.3.



(a) $\gamma = 0$, $d = 0.5$ (b) $\gamma = 1$, $d = 0.5$

Fig. 2. Maximum weight averaged over 100 simulations of length 5 000 time steps each. Maximum weight has been computed every 50 time steps.

In Fig. 2, we compare the maximum weight produced by a single-delegation process to the optimal maximum weight in a double-delegation process, for different values of γ. Corresponding figures for different values of d and γ can be found in the full version [8].

These simulations show that our asymptotic findings translate into considerable differences even for small numbers of agents, across different values of d. Moreover, these differences remain nearly as pronounced for values of γ up to 1, which corresponds to classical preferential attachment. This suggests that our mechanism can outweigh the social tendency towards concentration of votes; however, evidence from real-world elections is needed to settle this question. Lastly, we would like to point out the similarity between the graphs for the optimal maximum weight and the result of the greedy algorithm, which indicates that a large part of the separation can be attributed to the power of choice.

If we increase γ to large values, the separation between single and double delegation disappears. As we show in the full version [8], there are even combinations of $\gamma > 1$ and d such that the curve for single delegation falls below the ones for double delegation. In these settings, since a large fraction of delegators give two identical delegation options, any resolution mechanism has virtually no leverage. In the double delegation setting, indegrees grow faster, which makes the delegations concentrate toward a single voter more quickly than in classical liquid democracy, leading to a wildly unrealistic concentration of weight. Thus, it seems that large values of γ do not actually describe our scenario of multiple delegations.

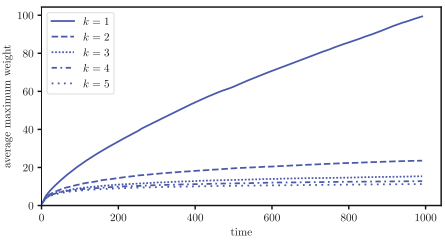


Fig. 3. Optimal maximum weight for different k averaged over 100 simulations, computed every 10 steps. $\gamma = 1$, $d = 0.5$.

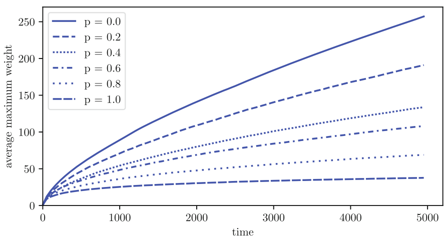


Fig. 4. Optimal maximum weight averaged over 100 simulations. Voters give two delegations with probability p; else one. $\gamma = 1$, $d = 0.5$.

As we have seen, switching from single delegation to double delegation greatly improves the maximum weight in plausible scenarios. It is natural to wonder whether increasing k beyond 2 will yield similar improvements. As Fig. 3 shows, however, the returns of increasing k quickly diminish, which is common to many incarnations of the power of choice [1].

4.2 Evaluating Mechanisms

Already the case of $k = 2$ appears to have great potential; but how easily can we tap it?

We have observed that, on average, the greedy "power of choice" mechanism comes surprisingly close to the optimal solution. However, this greedy mechanism depends on seeing the order in which our random process inserts agents and on the fact that all generated graphs are acyclic, which need not be true in practice. If the graphs were acyclic, we could simply first sort the agents topologically and then present the agents to the greedy mechanism in reverse order. On arbitrary active graphs, we instead proceed through the strongly connected components in reversed topological order, breaking cycles and performing the greedy step over the agents in the component. To avoid giving the greedy algorithm an unfair advantage, we use this generalized greedy mechanism throughout this section. Thus, we compare the generalized greedy mechanism, the optimal solution, the $(1 + \log |V|)$-approximation algorithm[3] and a random mechanism that materializes a uniformly chosen option per delegator.

On a high level, we find that both the generalized greedy algorithm and the approximation algorithm perform comparably to the optimal confluent flow solution. Across a variety of values of d and γ examined in the full paper [8], all three mechanisms seem similarly effective in exploiting the advantages of double delegation.

The similar success of these three mechanisms might indicate that our probabilistic model for $k = 2$ generates delegation networks that have low maximum weights for arbitrary resolutions. However, this is not the case: The random mechanism does quite poorly already on small instances, and the gap between random and the other mechanisms only grows further as t increases, as indicated by Fig. 5. In general, the graph for random delegations looks more similar to single delegation than to the other mechanisms on double delegation. Indeed, for $\gamma = 0$, random delegation is equivalent to the process with $k = 1$, and, for higher values of γ, it performs even slightly worse since the unused delegation options make the graph more centralized (see the full paper [8]). Thus, if simplicity is a primary desideratum, we recommend using the generalized greedy algorithm.

As Fig. 6 and additional measurements detailed in the full version [8] demonstrate, all three other mechanisms, including the optimal solution, easily scale to input sizes as large as the largest implementations of liquid democracy to date.

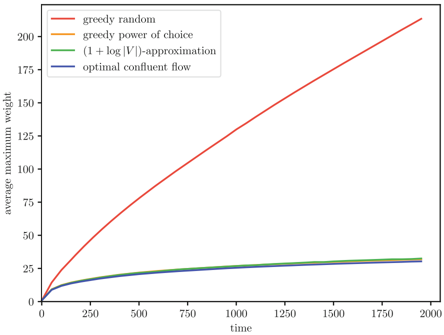


Fig. 5. Maximum weight per algorithm for $d = 0.5$, $\gamma = 1$, $k = 2$, averaged over 100 simulations.

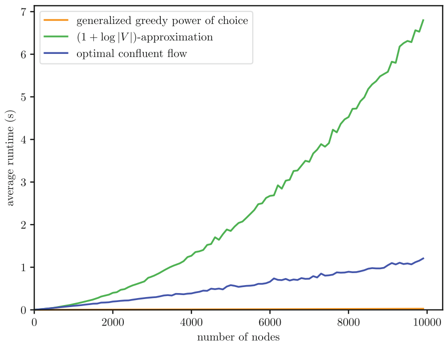


Fig. 6. Running time of mechanisms on graphs for $d = 0.5$, $\gamma = 1$, averaged over 20 simulations.

[3] For one of their subprocedures, instead of directly optimizing a convex program, Chen et al. [5] reduce this problem to finding a lexicographically optimal maximum flow in $\mathcal{O}(n^5)$. We choose to directly optimize the convex problem in Gurobi, hoping that this will increase efficiency in practice.

5 Discussion

The approach we have presented and analyzed revolves around the idea of allowing agents to specify multiple delegation options, and selecting one such option per delegator. A natural variant of this approach corresponds to splittable—instead of confluent—flow. In this variant, the mechanism would not have to commit to a single outgoing edge per delegator. Instead, a delegator's weight could be split into arbitrary fractions between her potential delegates. Indeed, such a variant would be computationally less expensive, and the maximum voting weight can be no higher than in our setting. However, we view our concept of delegation as more intuitive and transparent: Whereas, in the splittable setting, a delegator's vote can disperse among a large number of agents, our mechanism assigns just one representative to each delegator. As hinted at in the introduction, this is needed to preserve the high level of accountability guaranteed by classical liquid democracy. We find that this fundamental shortcoming of splittable delegations is not counterbalanced by a marked decrease in maximum weight. Indeed, representative empirical results given in the full version [8] show that the maximum weight trace is almost identical under splittable and confluent delegations. Furthermore, note that in the preferential delegation model with $k = 1$, splittable delegations do not make a difference, so the lower bound given in Theorem 3 goes through. And, when $k \geq 2$, the upper bound of Theorem 4 directly applies to the splittable setting. Therefore, our main technical results in Sect. 3 are just as relevant to splittable delegations.

To demonstrate the benefits of multiple delegations as clearly as possible, we assumed that every agent provides two possible delegations. In practice, of course, we expect to see agents who want to delegate but only trust a single person to a sufficient degree. This does not mean that delegators should be required to specify multiple delegations. For instance, if this was the case, delegators might be incentivized to pad their delegations with very popular agents who are unlikely to receive their votes. Instead, we encourage voters to specify multiple delegations on a voluntary basis, and we hope that enough voters participate to make a significant impact. Fortunately, as demonstrated in Fig. 4, much of the benefits of multiple delegation options persist even if only a fraction of delegators specify two delegations.

Without doubt, a centralized mechanism for resolving delegations wields considerable power. Even though we only use this power for our specific goal of minimizing the maximum weight, agents unfamiliar with the employed algorithm might suspect it of favoring specific outcomes. To mitigate these concerns, we propose to divide the voting process into two stages. In the first, agents either specify their delegation options or register their intent to vote. Since the votes themselves have not yet been collected, the algorithm can resolve delegations without seeming partial. In the second stage, voters vote using the generated delegation graph, just as in classic liquid democracy, which allows for transparent decisions on an arbitrary number of issues. Additionally, we also allow delegators to change their mind and vote themselves if they are dissatisfied with how delegations were resolved. This gives each agent the final say on their share of votes, and can only further reduce the maximum weight achieved by our mechanism. We believe that this process, along with education about the mechanism's goals and design, can win enough trust for real-world deployment.

Beyond our specific extension, one can consider a variety of different approaches that push the current boundaries of liquid democracy. For example, in a recent position paper, Brill [4] raises the idea of allowing delegators to specify a ranked list of potential representatives. His proposal is made in the context of alleviating delegation cycles, whereas our focus is on avoiding excessive concentration of weight. But, on a high level, both proposals envision centralized mechanisms that have access to richer inputs from agents. Making and evaluating such proposals now is important, because, at this early stage in the evolution of liquid democracy, scientists can still play a key role in shaping this exciting paradigm.

Acknowledgments. We are grateful to Miklos Z. Racz for very helpful pointers to analyses of preferential attachment models.

References

1. Azar, Y., Broder, A.Z., Karlin, A.R., Upfal, E.: Balanced allocations. In: Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC), pp. 593–602 (1994)
2. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science **286**, 509–512 (1999)
3. Behrens, J., Kistner, A., Nitsche, A., Swierczek, B.: The principles of LiquidFeedback. Interaktive Demokratie (2014)
4. Brill, M.: Interactive democracy. In: Proceedings of the 17th International Conference on Autonomous Agents and Multi-agent Systems (AAMAS) (2018)
5. Chen, J., Kleinberg, R.D., Lovász, L., Rajaraman, R., Sundaram, R., Vetta, A.: (Almost) tight bounds and existence theorems for single-commodity confluent flows. J. ACM **54**(4) (2007). Article 16
6. Chen, J., Rajaraman, R., Sundaram, R.: Meet and merge: Approximation algorithms for confluent flows. J. Comput. Syst. Sci. **72**(3), 468–489 (2006)
7. Christoff, Z., Grossi, D.: Binary voting with delegable proxy: An analysis of liquid democracy. In: Proceedings of the 16th Conference on Theoretical Aspects of Rationality and Knowledge (TARK), pp. 134–150 (2017)
8. Gölz, P., Kahng, A., Mackenzie, S., Procaccia, A.D.: The fluid mechanics of liquid democracy (2018). https://arxiv.org/pdf/1808.01906.pdf
9. Green-Armytage, J.: Direct voting and proxy voting. Const. Polit. Econ. **26**(2), 190–220 (2015)
10. Haslegrave, J., Jordan, J.: Preferential attachment with choice. Random Struct. Algorithms **48**(4), 751–766 (2016)
11. Kahng, A., Mackenzie, S., Procaccia, A.D.: Liquid democracy: An algorithmic perspective. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI) (2018)
12. Kling, C.C., Kunegis, J., Hartmann, H., Strohmaier, M., Staab, S.: Voting behaviour and power in online democracy. In: Proceedings of the 9th International AAAI Conference on Web and Social Media (ICWSM), pp. 208–217 (2015)

13. Kumar, R., Novak, J., Tomkins, A.: Structure and evolution of online social networks. In: Proceedings of the 12th International Conference on Knowledge Discovery and Data Mining (KDD), pp. 611–617 (2006)
14. Malyshkin, Y., Paquette, E.: The power of choice over preferential attachment. Lat. Am. J. Probab. Math. Stat. **12**(2), 903–915 (2015)
15. Miller, J.C.: A program for direct and proxy voting in the legislative process. Public Choice **7**(1), 107–113 (1969)
16. Newman, M.E.J.: Clustering and preferential attachment in growing networks. Phys. Rev. E **64**(2), 1–13 (2001)