



HAL
open science

Preserving Consistency in Multi-Issue Liquid Democracy

Rachael Colley, Umberto Grandi

► **To cite this version:**

Rachael Colley, Umberto Grandi. Preserving Consistency in Multi-Issue Liquid Democracy. 31st International Joint Conference on Artificial Intelligence (IJCAI 2022), International Joint Conferences on Artificial Intelligence (IJCAI), Jul 2022, Vienna, Austria. pp.201-207, 10.24963/ijcai.2022/29 . hal-03858789

HAL Id: hal-03858789

<https://hal.science/hal-03858789>

Submitted on 17 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Preserving Consistency in Multi-Issue Liquid Democracy

Rachael Colley, Umberto Grandi

IRIT, University of Toulouse, France

{rachael.colley, umberto.grandi}@irit.fr

Abstract

Liquid democracy bridges the gap between direct and representative democracy by allowing agents to vote directly on an issue or delegate to a trusted voter. Yet, when applied to votes on multiple interconnected issues, liquid democracy can lead agents to submit inconsistent votes. Two approaches are possible to maintain consistency: either modify the voters' ballots by ignoring problematic delegations, or resolve all delegations and make changes to the final votes of the agents. We show that rules based on minimising such changes are NP-complete. We propose instead to elicit and apply the agents' priorities over the delegated issues, designing and analysing two algorithms that find consistent votes from the agents' delegations in polynomial time.

1 Introduction

Delegative voting, most notably the setting called liquid democracy in which delegations are transitive, is currently under scrutiny by researchers in artificial intelligence. On the one hand, as a collective intelligence application in which its ability to track truth messages from noisy signals is evaluated [Caragiannis and Micha, 2019; Becker *et al.*, 2021; Dhillon *et al.*, 2021; Halpern *et al.*, 2021; Kahng *et al.*, 2021; Zhang and Grossi, 2022], and on the other hand, as an algorithmically rich application of interactive democracy [Boldi *et al.*, 2011; Kotsialou and Riley, 2020; Brill *et al.*, 2022; Markakis and Papasotiropoulos, 2021].

Liquid democracy is mostly studied in its simplest form: a binary decision has to be taken by a group of voters, who are allowed to express a direct vote (eventually including abstentions) or delegate their vote to another voter. However, its main potential lies in the fact that virtually any collective decision mechanism can be “liquidised”—to borrow a term from Brill and Talmon [2018], i.e., to include transitive delegations: for example, the liquid version of knapsack voting (related to participatory budgeting) allows a voter to delegate their assessment of different projects to different voters [Jain *et al.*, 2021]. Another example is pairwise liquid democracy [Brill and Talmon, 2018], in which a voter can delegate relative comparisons over two competing alternatives to (possibly different) voters in preference aggregation.

However, applying delegative voting to more complex multi-issue collective decisions runs into the problem of maintaining voters' consistency (e.g., in the two settings previously mentioned, the projects approved by the agent should not exceed a given budget, and the agent's preferences should be transitive) while at the same time utilising voters' delegations in the input as much as possible. Optimisation-based approaches proposed in previous work all run into high computational costs, and approximation algorithms may be hard to defend in a social choice setting where fairness criteria and explanatory power of the outcome are of primary importance.

Our Contribution. We propose to elicit a voter's priorities over the issues on which they are delegating and feed them to tractable rules that maintain ballot consistency. Such an elicitation can easily be performed by, for example, using the order in which voters enter their delegations in the voting platform. We first consider two rules that minimise either the number of ignored delegations in the voters' ballots, as proposed in previous work by Brill and Talmon [2018] and Jain *et al.* [2021], or minimise the number of changes to the agents' final votes once all delegations have been resolved. After showing that both rules are NP-complete, we propose two polynomial algorithms inspired by the two minimisation rules, that apply the agents' priorities to output consistent final votes. Our results show that although these principled algorithms cannot be considered as approximations of the minimisation rules, one of our algorithms respects the agents' priorities over their delegations more than its minimisation counterpart. Furthermore, when focusing on settings with budget constraints, we show that one of our minimisation rules outperforms the other rules when considering their global effects on the acceptance rates of projects.

Related Work. The seminal work of Christoff and Grossi [2017] already considered a “liquidised” version of binary aggregation on interdependent issues. This paper builds on this, generalising and unifying the approach of Jain *et al.* [2021] and Brill and Talmon [2018], where the former pertains to approved projects respecting a budget and the latter to accepted pairwise comparisons being transitive. More precisely, we study liquid democracy over multiple binary issues where the final opinions are bounded to satisfy a constraint. Another model considering a liquid version of preference aggregation is that of Harding [2021],

however, this work guides agents in choosing a proxy that is consistent with their views. The algorithmic literature on liquid democracy has mostly focused on eliciting rankings over potential delegates to solve the problem of delegation cycles [Kotsialou and Riley, 2020; Brill *et al.*, 2022; Dey *et al.*, 2021] or to study game-theoretical solution concepts where liquid democracy is interpreted as a voting game [Escoffier *et al.*, 2019; Armstrong and Larson, 2021; Zhang and Grossi, 2021]. As already observed by Christoff and Grossi [2017] liquid democracy relates strongly to opinion diffusion, where a delegation can be interpreted as an influence link. Constraints in opinion diffusion have been considered by Friedkin *et al.* [2016] for real-valued beliefs and by Botan *et al.* [2019] for majoritarian updates on binary issues.

Paper Structure. The remainder of the paper is organised as follows. Section 2 presents the basic definitions of the model. Section 3 presents two rules that minimise the number of delegations that have to be deleted to find consistent votes, showing that they are intractable. In Section 4 we introduce the notion of priorities over the issues as well as two rules combining delegation deletion and the agents’ priorities, showing that they terminate with consistent votes in polynomial time (Section 4.1). In Section 4.2 we show that the tractable rules are not efficient approximations of their minimisation counterparts. In Section 4.3 we compare the four rules on how well they respect the priorities. Section 5 focuses on knapsack constraints and in Section 6 we conclude.

2 The Model

We model a group of agents $\mathcal{N} = \{1, \dots, n\}$ making a collective decision on a set of issues $\mathcal{I} = \{1, \dots, m\}$, where the domain of each issue $j \in \mathcal{I}$ is $\mathcal{D}(j)$. Without loss of generality we will consider binary issues, thus $\mathcal{D}(j) = \{0, 1\}$ for all $j \in \mathcal{I}$.¹ Each agent $i \in \mathcal{N}$ submits a ballot $B_i \in \Pi_{j \in \mathcal{I}}(\mathcal{D}(j) \cup \mathcal{N} \setminus \{i\})$, which specifies for each issue $j \in \mathcal{I}$ if the agent gives a direct vote in $\mathcal{D}(j)$ or a delegation to any other agent in $\mathcal{N} \setminus \{i\}$. A profile of ballots, one for each agent in \mathcal{N} , is denoted by \mathbf{B} , and the vote of agent i on issue j in profile \mathbf{B} is denoted as B_{ij} . We let $\hat{\mathbf{B}}$ be an $n \times m$ matrix containing only the direct votes of the agents in profile \mathbf{B} , where $\hat{B}_{ij} = B_{ij}$ if $B_{ij} \in \{0, 1\}$ and $\hat{B}_{ij} = \Delta$, otherwise. We let $\mathbf{X} \in \{0, 1\}^{n \times m}$ denote a profile of votes.

Example 1. Consider two agents $\mathcal{N} = \{C, D\}$, two issues $\mathcal{I} = \{j, k\}$, and a profile \mathbf{B} composed of ballots $B_C = (1, D)$ and $B_D = (C, 1)$, where C delegates to D on issue k , and D delegates to C on issue j . The direct votes of the agents are $\hat{\mathbf{B}} = ((1, \Delta), (\Delta, 1))$.

We assume that there are no delegation cycles on any given issue. To clarify this, we associate a labelled delegation graph $G_{\mathbf{B}} = (V, E)$ with each profile \mathbf{B} , where $V = \mathcal{N}$ and edge $(C, D, j) \in E$ if C delegates to D on issue j in \mathbf{B} (where j is the label of the edge). We assume that $G_{\mathbf{B}}$ is acyclic for any issue, i.e., for each $(C, D, j) \in E$ there is no j -path from

¹Non-binary issues can be expressed in the binary setting by letting each alternative become a binary issue where the constraint permits only one alternative to be chosen.

C returning to C . This allows us to focus on preserving consistency without making assumptions on how delegation cycles are broken, in a similar vein as the previous work [Brill and Talmon, 2018; Jain *et al.*, 2021]. Thanks to the assumption of acyclicity, we can define a *canonical profile of votes* $\mathbf{X}^{\mathbf{B}} \in \{0, 1\}^{n \times m}$ for every \mathbf{B} , where all delegating voters are assigned the vote of the direct voter at the end of the delegation chain (sometimes known as a *guru* or an *ultimate delegate*). Given a delegation graph $G_{\mathbf{B}}$, for each $i \in \mathcal{N}$ and $j \in \mathcal{I}$, we find the endpoint $k \in \mathcal{N}$ of the longest outgoing j -path from i and let $X_{ij}^{\mathbf{B}} = \hat{B}_{kj}$.

We assume that the issues in \mathcal{I} are interconnected, i.e., that the votes of the agents have to respect a given constraint or set of constraints Γ . For instance, in knapsack voting agents have to respect a given budget in the set of approved projects, or in preference aggregation the approved comparisons are assumed to be transitive. If $X_i \in \{0, 1\}^m$, we denote with $X_i \models \Gamma$ the fact that agent i ’s vote satisfies the constraint Γ . If \mathbf{X} is a profile of votes, we write $\mathbf{X} \models \Gamma$ when $X_i \models \Gamma$ for all $i \in \mathcal{N}$. The set of all consistent votes is denoted by $\mathcal{X}_{\Gamma} = \{X \mid X \models \Gamma\}$, assuming that Γ is not a contradiction, i.e. $\mathcal{X}_{\Gamma} \neq \emptyset$. We allow for any representation of Γ , with the only restriction being that checking if a partial vote can be completed to a consistent one should be feasible in polynomial time. For instance, budget constraints, logical formulas in complete-DNF, and the transitivity of preferences would all be acceptable constraints. Constraints expressed as arbitrary logical formulas do not, as the problem of completing partial evaluations is equivalent to SAT. Formally, if $\bar{X} \in \{0, 1, \Delta\}^m$ is a partial vote, we write $\bar{X} \approx \Gamma$ if there exists a complete vote $X \in \{0, 1\}^m$ such that $\bar{X} \subseteq X$ and $X \models \Gamma$.² We assume that every agent’s (partial) direct votes can be completed, i.e., for all $i \in \mathcal{N}$ we assume $\hat{B}_i \approx \Gamma$. In line with previous work, we say that a profile of ballots \mathbf{B} is *consistent* if $\mathbf{X}^{\mathbf{B}} \models \Gamma$.³ From our assumptions that delegations are acyclic and the membership problem of Γ is tractable, we obtain that:

Remark 1. To check if a profile \mathbf{B} is consistent is polynomial.

As profiles may not be consistent, we define consistent delegation rules that take a profile of ballots \mathbf{B} and return a profile of consistent votes, preserving the voters’ direct votes:

Definition 1. Given a constraint Γ , a consistent delegation rule \mathcal{F} takes a profile of ballots \mathbf{B} and returns a profile of votes such that $\mathcal{F}(\mathbf{B}) \models \Gamma$ and $\hat{\mathbf{B}} \subseteq \mathcal{F}(\mathbf{B})$.

Example 2. Consider the same setting as Example 1, with the constraint that j and k cannot both be accepted. Thus, Γ can be represented as $\neg j \vee \neg k$ (a logical formula in complete DNF) and $\mathcal{X}_{\Gamma} = \{(0, 0), (0, 1), (1, 0)\}$. Following the agents’ delegations in \mathbf{B} , the canonical profile of votes is $\mathbf{X}^{\mathbf{B}} = ((1, 1), (1, 1))$. Thus, \mathbf{B} is not consistent. Observe that any consistent delegation rule would give the outcome $((1, 0), (0, 1))$ (where the direct votes are not changed).

²Here we slightly abuse the subset notation, for a partial assignment $\bar{X} \in \{0, 1, \Delta\}^m$ and assignment $X \in \{0, 1\}^m$, we say $\bar{X} \subseteq X$ when X is found by only changing Δ s in \bar{X} to 0 or 1.

³Brill and Talmon [2018] also define a weaker notion of consistency. Both are equivalent to our definition on acyclic profiles.

3 Minimal Changes to Ballots and Votes

In this section we introduce two consistent delegation rules that preserve vote consistency by making minimal changes, showing that associated decision problems are NP-complete.

The first approach is to modify the profile of ballots \mathbf{B} by ignoring a minimal number of delegations that conflict with the constraint, replacing them with a direct vote. This rule has been defined in previous work on knapsack voting [Jain *et al.*, 2021] and preference aggregation [Brill and Talmon, 2018]. Given a profile \mathbf{B} , the *minimal delegation change rule* MDC finds all consistent profiles \mathbf{B}' that replace a minimal number of delegations with direct votes and whose canonical profile of votes $\mathbf{X}^{\mathbf{B}'}$ is consistent with Γ :

$$\text{MDC}(\mathbf{B}) = \{ \mathbf{X}^{\mathbf{B}'} \mid \mathbf{B}' \in \arg \max_{\{ \mathbf{B}' \mid \hat{\mathbf{B}} \subseteq \mathbf{X}^{\mathbf{B}'} \ \& \ \mathbf{X}^{\mathbf{B}'} \models \Gamma \}} \sum_{i \in \mathcal{N}} | \mathbf{B}_i \cap \mathbf{B}'_i | \}$$

We now show that the following decision problem associated with MDC is NP-complete.

MINIMALDELCHANGE. Given a profile \mathbf{B} , constraint Γ and integer $k \geq 0$, is there a consistent profile \mathbf{B}' found by changing at most k delegations of \mathbf{B} to direct votes?

Proposition 1. *MINIMALDELCHANGE is NP-complete.*

Proof. A sub-problem of MINIMALDELCHANGE is the consistent knapsack voting problem CKV from Jain *et al.* [2021] (they also consider acyclic delegation graphs, as CKV removes cycles by replacing every delegation in a cycle with a direct vote against the issue). CKV is an NP-complete problem, as shown by Jain *et al.* [2021] (Theorem 1), implying that MINIMALDELCHANGE is NP-hard. To see that MINIMALDELCHANGE is in NP, consider a certificate that lists agents and issues, indicating delegations to be changed to direct votes to obtain \mathbf{B}' from \mathbf{B} . First we check that the list has at most k entries. Then we check whether \mathbf{B}' is consistent, which can be done in polynomial time by Remark 1. \square

The second approach, loosely inspired by related literature on judgment aggregation (see, e.g., Lang *et al.* [2011]), makes minimal changes to the canonical profile of votes $\mathbf{X}^{\mathbf{B}}$ directly, but only on issues where the voter expressed a delegation. The *minimal vote change rule* MVC is defined as follows:

$$\text{MVC}(\mathbf{B}) = \arg \max_{\{ \mathbf{X} \mid \hat{\mathbf{B}} \subseteq \mathbf{X} \ \& \ \mathbf{X} \models \Gamma \}} \sum_{i \in \mathcal{N}} | X_i \cap X_i^{\mathbf{B}} |$$

In line with Proposition 1, we now show that the following decision problem associated with MVC is NP-complete.

MINIMALVOTECHANGE. Given a profile \mathbf{B} , constraint Γ , and integer $k \geq 0$, is there an $\mathbf{X} \models \Gamma$ such that $\hat{\mathbf{B}} \subseteq \mathbf{X}$ and \mathbf{X} is found by changing at most k votes in $\mathbf{X}^{\mathbf{B}}$?

Proposition 2. *MINIMALVOTECHANGE is NP-complete.*

Proof. For membership in NP, we take a certificate that lists pairs of agents and issues to be changed in $\mathbf{X}^{\mathbf{B}}$ to define \mathbf{X}' . We then check in polynomial time that the list has k or less entries, and that $X'_i \models \Gamma$ for each $i \in \mathcal{N}$.

To show NP-hardness, we reduce from the NP-complete problem independent set INDSET [Karp, 1972]. The input of INDSET is a graph $G = (V, E)$ and integer $t \geq 0$. It asks if there is a $V' \subseteq V$ with $|V'| \geq t$ such that no edge in E has both ends in V' . Given an instance of INDSET , consider a set of agents $\mathcal{N} = \{A_v \mid \text{for all } v \in V\} \cup \{A\}$, issues $\mathcal{I} = \{I_v \mid v \in V\}$, and bound $k = |V| - t$. Next, our constraint only allows for independent sets: $\Gamma_{\text{IND}} = \{ (I_u \rightarrow \neg I_v) \wedge (I_v \rightarrow \neg I_u) \mid (u, v) \in E \}$. First, observe that we can check in polynomial time if a partial assignment can be completed to one satisfying Γ_{IND} , as we only need to check that the vertices accepted in the partial assignment are an independent set. Finally, consider a profile of ballots \mathbf{B} such that agent A delegates to agent A_v on issue I_v for all $v \in V$, and each A_v only approves of I_v , rejecting all other issues.

Assume there is an independent set $V' \subseteq V$ with $|V'| \geq t$. Given that $X_A^{\mathbf{B}} = (1, \dots, 1)$, consider X_A containing votes for the issues I_v with $v \in V'$ and against the remaining issues. The resulting profile of votes is consistent: for each $v \in V$, X_{A_v} is consistent as only one issue is accepted, and $X_A \models \Gamma_{\text{IND}}$ as X_A reflects an independent set. Thus, we also have a solution to our problem as the number of votes changed is $|V| - |V'| \leq |V| - t = k$. Next assume that there is no independent set of size at least t , and suppose $t' < t$ is the size of the largest independent set of G . Observe that in order to make $X_A^{\mathbf{B}}$ consistent, at least $|V| - t'$ votes need to be reverted. As $|V| - t' > |V| - t = k$, there is no solution to our problem either. Hence, MINIMALVOTECHANGE is NP-hard, thus NP-complete. \square

4 Eliciting and Applying Priorities over Issues

In order to provide tractable and principled rules to preserve vote consistency, we propose to elicit from the voters their priorities over the issues they choose to delegate on—this could be, e.g., the order in which the voter expressed their delegations. In doing so we discard a fixed parameter tractability analysis of MDC and MVC , since Jain *et al.* [2021] already provided extensive negative results for this approach.⁴

We assume that voters specify a total ordering over the issues on which they express a delegation. To simplify the presentation, we assume that these orderings are on the whole set of issues \mathcal{I} (the two assumptions are equivalent since consistent delegation rules never change direct votes). We denote this order by \prec_i for each agent $i \in \mathcal{N}$, and we write $\prec_i(k)$ for the issue with the k^{th} highest priority in \prec_i .

In parallel with the minimisation rules presented in Section 3, we present two approaches based on either delegation changes or vote changes. First, the *priority delegation changing rule* (PDC) described in Algorithm 1 iteratively alters an agent's delegation to a direct vote if its addition is inconsistent. Second, the *priority vote changing rule* (PVC) described in Algorithm 2 iteratively adds consistent votes from $\mathbf{X}^{\mathbf{B}}$ following the order of priorities. We now define some notation

⁴Jain *et al.* [2021, Theorem 1] show that their version of MDC is tractable only under restrictive conditions such as when voters delegate on at most one issue. Moreover, their problem is NP-complete even when restricting many parameters to be small constants.

Algorithm 1 Priority delegation changing (PDC)

```

1: Input:  $\mathbf{B}$ ,  $\prec_i$  for all  $i \in \mathcal{N}$ 
2:  $t = 0$ 
3: Find  $\mathbf{X}^0 = \hat{\mathbf{B}}$ 
4: while  $\mathbf{X}^t \notin \{0, 1\}^{n \times m}$  do
5:    $\mathbf{X}^{t+1} := \mathbf{X}^t$ 
6:   for  $i \in \mathcal{N}$  and  $k \in [1, m]$  do
7:      $j := \prec_i(k)$ 
8:     if  $\text{vote}(\mathbf{X}^t, i, j) = \Delta$  and  $\text{vote}(\mathbf{X}^t, B_{ij}, j) \in \{0, 1\}$  then
9:       if  $(\text{bal}(\mathbf{X}^{t+1}, i)_{-j}, \text{vote}(\mathbf{X}^t, B_{ij}, j)) \approx \Gamma$ 
10:      then  $\text{vote}(\mathbf{X}^{t+1}, i, j) := \text{vote}(\mathbf{X}^t, B_{ij}, j)$ 
11:      else  $\text{vote}(\mathbf{X}^{t+1}, i, j) := 1 - \text{vote}(\mathbf{X}^t, B_{ij}, j)$ 
12:    $\mathbf{X} := \mathbf{X}^t$ 
    
```

used in the algorithms. Recall that $\hat{\mathbf{B}}$ denotes an $n \times m$ matrix containing only the direct votes of \mathbf{B} . We let $\text{bal}(\mathbf{X}, i)$ denote the vector of votes recorded for agent i in matrix \mathbf{X} and $\text{vote}(\mathbf{X}, i, j)$ denote the vote recorded in \mathbf{X} for agent i on issue j . Furthermore, (X_{-j}, x_j) denotes the vector X with the entry for issue j appended with a new entry x_j .

Algorithm 1 inputs the profile \mathbf{B} and the agents' priorities over the issues \prec_i , and then sets the counter t to 0. On line 4 the algorithm starts a `while`-loop that continues until every agent has a vote in $\{0, 1\}$ for every issue. Each iteration of the `while`-loop checks whether an update can be made for each agent following their priority order over the issues. In the current \mathbf{X}^t , if i does not have a vote for j and their delegate does (on line 8), we check if adding their delegate's vote is consistent, if so we update X_i^{t+1} with it (line 9), otherwise we add the opposite (line 10). Note that the update to \mathbf{X}^{t+1} is simultaneous as it uses information from \mathbf{X}^t . Hence, the order of agents in the `for`-loop does not affect the outcome.

Algorithm 2 describes PVC, it inputs the profile \mathbf{B} and the agents' priorities. It then finds $\hat{\mathbf{B}}$ and the canonical profile of votes \mathbf{X}^B . For each agent $i \in \mathcal{N}$ it checks in the priority order \prec_i (line 4) if they have a vote recorded in \mathbf{X}^0 (line 6). If the addition of X_{ij}^B is consistent then we update X_i^0 with it (line 7), if not we update the opposite (line 8).

Example 3. Consider agents $\mathcal{N} = \{C, D, E, F\}$ and issues $\mathcal{I} = \{j, k\}$ where at most one issue can be accepted, i.e., $\Gamma = \neg j \vee \neg k$. Consider profile \mathbf{B} in which agents C and D vote directly, $B_C = (1, 0)$ and $B_D = (0, 1)$, whereas the remaining agents delegate as such: $B_E = (C, D)$ and $B_F = (E, E)$. Agent E has the priority $j \succeq_E k$ while F has the priority $k \succeq_F j$. PDC first considers delegations of the highest priority issue. The top priority issue for E is j and their delegate has a vote for j in \mathbf{X}^0 . As E currently has no votes in \mathbf{X}^0 we let $X_{Ej}^1 = 1$. Then the delegation on the second priority issue k is not consistent, thus, $X_{Ek}^1 = 0$. As in \mathbf{X}^0 , E does not have any votes, F 's votes can only be added in the second iteration. Thus, $X_F^2 = (1, 0)$ and $\text{PDC}(\mathbf{B}) = ((1, 0), (0, 1), (1, 0), (1, 0))$. PVC first computes the canonical profile of votes $\mathbf{X}^B = ((1, 0), (0, 1), (1, 1), (1, 1))$, where X_E^B and X_F^B are inconsistent. For similar reasons as when

Algorithm 2 Priority vote changing (PVC)

```

1: Input:  $\mathbf{B}$ ,  $\prec_i$  for all  $i \in \mathcal{N}$ 
2: Find  $\mathbf{X}^0 = \hat{\mathbf{B}}$ 
3: Find  $\mathbf{X}^B$ 
4: for  $i \in \mathcal{N}$  and  $k \in [1, m]$  do
5:    $j := \prec_i(k)$ 
6:   if  $\text{vote}(\mathbf{X}^0, i, j) = \Delta$  then
7:     if  $(\text{bal}(\mathbf{X}^0, i)_{-j}, \text{vote}(\mathbf{X}^0, X_{ij}^B, j)) \approx \Gamma$  then
8:        $\text{vote}(\mathbf{X}^0, i, j) := \text{vote}(\mathbf{X}^0, X_{ij}^B, j)$ 
9:     else  $\text{vote}(\mathbf{X}^0, i, j) := 1 - \text{vote}(\mathbf{X}^0, X_{ij}^B, j)$ 
10:   $\mathbf{X} := \mathbf{X}^0$ 
    
```

using PDC, PVC then changes E 's vote to $X_E^0 = (1, 0)$. However, for F , it first tries $X_{Fk}^0 = 1$ in accordance with F 's priorities, which is consistent, entailing that $X_{Fj}^0 = 0$. Thus, $\text{PVC}(\mathbf{B}) = ((1, 0), (0, 1), (1, 0), (0, 1))$.

4.1 Complexity of PDC and PVC

In the absence of constraints it is easy to see that all four rules output the canonical profile of votes associated to the profile.

Remark 2. When $\Gamma = \top$, PDC, PVC, MDC and MVC give the same outcome, namely \mathbf{X}^B .

We next show that PDC and PVC always terminate and output a consistent profile of votes.

Proposition 3. For any acyclic profile \mathbf{B} and constraint Γ , PDC and PVC terminate, $\text{PDC}(\mathbf{B}) \models \Gamma$ and $\text{PVC}(\mathbf{B}) \models \Gamma$.

Proof. We start from PDC. Due to the `while`-loop on line 4, if Algorithm 1 terminates, it does so with $\mathbf{X} \in \{0, 1\}^{n \times m}$. We assume for a contradiction that Algorithm 1 does not terminate, and therefore, $X_{ij}^t = \Delta$ for all $t > T$ after some iteration $T \in \mathbb{N}$ and for some $i \in \mathcal{N}$ and $j \in \mathcal{I}$. Note that $B_{ij} \notin \{0, 1\}$ as the algorithm does not change any direct votes to Δ . Therefore, B_{ij} is a delegation, and since votes are propagated following delegations we can infer that also $X_{B_{ij}j}^t = \Delta$. By the acyclicity of \mathbf{B} , the delegation chain on issue j starting at agent i must end at a direct voter for j , whose vote will then eventually reach voter i , against our assumption. Next, we show that every agent's votes are consistent with Γ . Recall that the initial direct votes are consistent, i.e., $\hat{B}_i \approx \Gamma$ for all $i \in \mathcal{N}$. Each time a vote is added, we check if its addition allows for a consistent completion and if not, the opposite is added. As $\mathcal{X}_\Gamma \neq \emptyset$, the process is guaranteed to give a consistent completion of votes.

Next we show the same for PVC. Algorithm 2 first finds \mathbf{X}^B . It then inspects the cells of \mathbf{X}^0 with a `for`-loop on line 4, cycling through every agent and issue (in order of their priorities), after which it will terminate. In line 6, it inspects every entry of \mathbf{X}^0 without a vote $X_{ij}^0 = \Delta$ and then updates it to a vote in $\{0, 1\}$ from line 7 to line 8. Thus, the algorithm always returns $\mathbf{X} \in \{0, 1\}^{n \times m}$. We next show that the agents' votes are consistent. The algorithm starts with $\hat{\mathbf{B}}$, which by assumption is such that $\hat{B}_i \approx \Gamma$ for all $i \in \mathcal{N}$. It then iteratively adds votes from \mathbf{X}^B if they can be completed

to a consistent set of votes, and the opposite otherwise. This will always lead to votes $X_i^0 \models \Gamma$. Therefore, PVC always terminates with $\mathbf{X}^{n \times m}$ and $X_i \models \Gamma$ for all $i \in \mathcal{N}$. \square

We show that both rules run in polynomial time. Recall that n is the number of voters, m the number of issues, and let ℓ be the time to check if a partial vote has a Γ -consistent completion (we assumed that ℓ is polynomial in n and m).

Proposition 4. *PDC terminates in $\mathcal{O}(nm(n + \ell))$ time.*

Proof. In each iteration of while-loop on line 4, Algorithm 1 first checks if $\mathbf{X}^t \in \{0, 1\}^{n \times m}$ in $\mathcal{O}(nm)$ time. Since each issue has at least one direct voter, in the first iteration of the while-loop the for-loop on line 6 checks at most $m(n - 1)$ delegations, to see if their delegate has a vote in \mathbf{X}^0 , each taking constant time. As the profile is acyclic, at least one vote for each issue is added in each iteration. Thus, in iteration t , the for-loop checks at most $m(n - t)$ delegations. Therefore, Algorithm 1 requires at most $\sum_{t=1}^n nm + (n - t)m \in \mathcal{O}(mn^2)$ time to check the while-loop condition and if a voter's delegate has a direct vote. Furthermore, once for each delegation, we check if the addition of their delegate's vote has a Γ -consistent completion, in total taking at most $m(n - 1)\ell$ steps. Hence, Algorithm 1 terminates in $\mathcal{O}(nm(n + \ell))$ time. \square

Proposition 5. *PVC terminates in $\mathcal{O}(nm(n + \ell))$ time.*

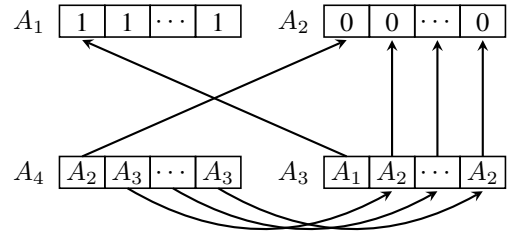
Proof. Algorithm 2 first finds \mathbf{X}^B in $\mathcal{O}(n^2m)$ time, where for each issue and each agent it takes at most $\mathcal{O}(n)$ time to unravel the delegation chain until a direct voter is found. For each delegation in \mathbf{B} , the for-loop on line 4 will check, following the order given by the priorities, if the addition of the vote found in \mathbf{X}^B along with the votes already in \mathbf{X}^0 can be completed to comply with Γ . As there are at most $m(n - 1)$ delegations and each completion check takes ℓ steps, this can be done in $\mathcal{O}(nm\ell)$ time. Summing the two figures we obtain that Algorithm 2 gives an outcome in $\mathcal{O}(nm(n + \ell))$ time. \square

4.2 Approximation Bounds

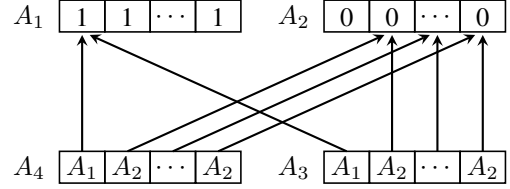
Next we assess whether PDC and PVC are good polynomial approximations of MDC and MVC, giving a negative response. This is unsurprising, as PDC and PVC aim to respect priorities rather than minimise changes. We abuse notation by defining $\text{flip}(\mathcal{F}(\mathbf{B}))$ to count the number of changes that \mathcal{F} makes when finding a consistent outcome on profile \mathbf{B} . This corresponds to delegations in \mathbf{B} being ignored by PDC and MDC, or direct votes being reverted in \mathbf{X}^B by PVC and MVC.

Proposition 6. *For each $n \geq 3$ and $m \geq 2$ there is a profile \mathbf{B} such that $\text{flip}(\text{PDC}(\mathbf{B})) = \mathcal{O}(nm) \times \text{flip}(\text{MDC}(\mathbf{B}))$.*

Proof. Consider $\mathcal{N} = \{A_1, \dots, A_n\}$ and $\mathcal{I} = \{i_1, \dots, i_m\}$, and $\mathcal{X}_\Gamma = \{(0, \dots, 0), (1, \dots, 1)\}$. Consider profile \mathbf{B} where $B_{A_1} = (1, \dots, 1)$, $B_{A_2} = (0, \dots, 0)$, and for the remaining agents let $B_{A_t} = (A_{t-2}, A_{t-1}, \dots, A_{t-1})$ for $t \in [3, n]$, each with priorities $i_x \succ_{A_t} i_y$ if and only if $x < y$. As replacing the delegation of A_3 on issue i_1 to a direct vote of 0 gives a consistent profile, $\text{flip}(\text{MDC}(\mathbf{B})) = 1$. However, PDC returns the vote $X_{A_t} = (1, \dots, 1)$ when t is odd and $X_{A_t} =$



(a) Profile from Proposition 6



(b) Profile from Proposition 7

Figure 1: The profiles from (a) Proposition 6 and (b) Proposition 7 when $\mathcal{N} = \{A_1, A_2, A_3, A_4\}$ and $|\mathcal{I}| = m$.

$(0, \dots, 0)$ when t is even. Thus, for the $n - 2$ delegating agents, A_t with $t \geq 2$, all of their delegations are flipped for the issues in $\mathcal{I} \setminus \{i_1\}$. Thus, $\text{flip}(\text{PDC}(\mathbf{B})) = (n - 2)(m - 1)$, and $\text{flip}(\text{PDC}(\mathbf{B})) = \mathcal{O}(nm) \times \text{flip}(\text{MDC}(\mathbf{B}))$. \square

Proposition 7. *For each $n \geq 3$ and $m \geq 2$ there is a profile \mathbf{B} such that $\text{flip}(\text{PVC}(\mathbf{B})) = \mathcal{O}(m) \times \text{flip}(\text{MVC}(\mathbf{B}))$.*

Proof. Consider $\mathcal{N} = \{A_1, \dots, A_n\}$ and $\mathcal{I} = \{i_1, \dots, i_m\}$, and $\mathcal{X}_\Gamma = \{(0, \dots, 0), (1, \dots, 1)\}$. The ballots of \mathbf{B} are as follows: $B_{A_1} = (1, \dots, 1)$, $B_{A_2} = (0, \dots, 0)$, and the remaining agents have the ballot $B_{A_t} = (A_1, A_2, \dots, A_2)$ for $t \in [3, n]$, with the priorities $i_x \succ_{A_t} i_y$ for $x < y$. The canonical profile \mathbf{X}^B is inconsistent since $X_{A_t} = (1, 0, \dots, 0)$ for $t > 2$. MVC alters the delegated vote of such voters on issue i_1 , hence $\text{flip}(\text{MVC}(\mathbf{B})) = n - 2$. PVC instead reverts all delegated votes except for the one on issue i_1 , hence $\text{flip}(\text{PVC}(\mathbf{B})) = (n - 2) \times (m - 1)$. Therefore, $\text{flip}(\text{PVC}(\mathbf{B})) = \mathcal{O}(m) \times \text{flip}(\text{MVC}(\mathbf{B}))$. \square

4.3 Comparing Rules on Priorities

In this section we compare the four consistent delegation rules with respect to the agents' priorities over the issues. We use the top^i function which identifies agent i 's highest priority issue from a subset of issues according to \prec_i . We first define the sets of issues on which a rule \mathcal{F} does not respect agent i 's delegations, either with respect to their delegate (*del*):

$$\text{del}_{\mathcal{F}}^i(\mathbf{B}) = \{j \mid j \in \mathcal{I}, B_{ij} \notin \{0, 1\} \text{ and } \mathcal{F}(\mathbf{B})_{ij} \neq \mathcal{F}(\mathbf{B})_{B_{ij}j}\},$$

or with respect to the canonical profile of vote (*dir*):

$$\text{dir}_{\mathcal{F}}^i(\mathbf{B}) = \{j \mid j \in \mathcal{I} \text{ and } X_{ij}^B \neq \mathcal{F}(\mathbf{B})_{ij}\}.$$

The set $\text{del}_{\mathcal{F}}^i(\mathbf{B})$ identifies the issues on which the votes of agent i in $\mathcal{F}(\mathbf{B})$ differ from their delegate's vote, whereas $\text{dir}_{\mathcal{F}}^i(\mathbf{B})$ identifies the issues on which the votes of agent i in the outcome $\mathcal{F}(\mathbf{B})$ differ to their votes in \mathbf{X}^B . Note

that there is no logical connection between these sets. Intuitively, agents want lower priority issues to be in these sets, as they reflect the delegations that have been ignored in finding consistent votes. For a profile \mathbf{B} , rules \mathcal{F} and \mathcal{F}' , and measure $c \in \{\text{del}, \text{dir}\}$, we say that an agent i *top-prefers* $c_{\mathcal{F}}^i(\mathbf{B})$ to $c_{\mathcal{F}'}^i(\mathbf{B})$ (denoted by $c_{\mathcal{F}}^i(\mathbf{B}) \succeq_i^{\text{top}} c_{\mathcal{F}'}^i(\mathbf{B})$) if $\text{top}^i(c_{\mathcal{F}}^i(\mathbf{B})) \preceq_i \text{top}^i(c_{\mathcal{F}'}^i(\mathbf{B}))$, where $\text{top}^i(c)$ gives the highest element of c with respect to \prec_i . Thus, top-preferring $c_{\mathcal{F}}^i(\mathbf{B})$ to $c_{\mathcal{F}'}^i(\mathbf{B})$ entails that the top issue whose delegation is ignored using \mathcal{F} has a lower priority than the top issue using \mathcal{F}' . Let $c_{\mathcal{F}}^{\mathbf{B}} = (c_{\mathcal{F}}^1(\mathbf{B}), \dots, c_{\mathcal{F}}^n(\mathbf{B}))$, and $c_{\mathcal{F}'}^{\mathbf{B}} \preceq_{\mathcal{N}}^{\text{top}} c_{\mathcal{F}}^{\mathbf{B}}$ if and only if $c_{\mathcal{F}'}^i(\mathbf{B}) \preceq_i^{\text{top}} c_{\mathcal{F}}^i(\mathbf{B})$ for every $i \in \mathcal{N}$.

The following proposition shows that PVC respects issues with higher priorities than its minimisation counterpart MVC.

Proposition 8. $\text{dir}_{\text{MVC}}^{\mathbf{B}} \preceq_{\mathcal{N}}^{\text{top}} \text{dir}_{\text{PVC}}^{\mathbf{B}}$ for any profile \mathbf{B} .

Proof. We assume for a contradiction that there exists a profile \mathbf{B} such that $\text{dir}_{\text{MVC}}^{\mathbf{B}} \not\preceq_{\mathcal{N}}^{\text{top}} \text{dir}_{\text{PVC}}^{\mathbf{B}}$. Therefore, there is an agent $i \in \mathcal{N}$ such that $\text{top}^i(\text{dir}_{\text{MVC}}^i(\mathbf{B})) \prec_i \text{top}^i(\text{dir}_{\text{PVC}}^i(\mathbf{B}))$. Suppose that $\text{top}^i(\text{dir}_{\text{MVC}}^i(\mathbf{B}))$ is i 's k^{th} priority and that $\text{top}^i(\text{dir}_{\text{PVC}}^i(\mathbf{B}))$ is i 's m^{th} priority (thus, $m < k$). MVC outputs a consistent vote including i 's first $(k-1)$ top-priority issues. Thus, the partial truth assignment \bar{X}_i including i 's direct votes and accepting the delegations on the top $(k-1)$ priority issues is such that $\bar{X}_i \approx \Gamma$. PVC on the other hand only accepts the delegations on i 's top $(m-1)$ priorities and rejects the delegation on the m^{th} issue. Yet if $\bar{X}_i \approx \Gamma$, then any $\bar{X}' \subseteq \bar{X}_i$ is such that $\bar{X}' \approx \Gamma$, including the partial assignment accepting i 's direct votes and their delegations on the first m priority issues. We have reached a contradiction as PVC rejected a partial assignment that could be completed. \square

We now show that the analogous result is not true for PDC.

Proposition 9. There exists profiles \mathbf{B} and \mathbf{B}' such that $\text{del}_{\text{MDC}}^{\mathbf{B}} \preceq_{\mathcal{N}}^{\text{top}} \text{del}_{\text{PDC}}^{\mathbf{B}}$ and $\text{del}_{\text{PDC}}^{\mathbf{B}'} \preceq_{\mathcal{N}}^{\text{top}} \text{del}_{\text{MDC}}^{\mathbf{B}'}$.

Proof. First we give \mathbf{B} such that $\text{del}_{\text{MDC}}^{\mathbf{B}} \preceq_{\mathcal{N}}^{\text{top}} \text{del}_{\text{PDC}}^{\mathbf{B}}$. We have $\mathcal{N} = \{C, D, E, F\}$ and issues $\mathcal{I} = \{i, j, k\}$ with the set of consistent votes being $\mathcal{X}_{\Gamma} = \{(1, 0, 0), (0, 1, 1)\}$. We let $B_C = (1, 0, 0)$, $B_D = (0, 1, 1)$, $B_E = (0, 1, 1)$ and agent F delegates as such: $B_F = (C, D, E)$ with the following priorities over the issues $i \succ_F j \succ_F k$. PDC first adds F 's delegation on i giving $X_F = (1, \Delta, \Delta)$, and then attempts to add the delegations on j and k but rejects both giving $X_F = (1, 0, 0)$. Thus, $\text{del}_{\text{PDC}}^F(\mathbf{B}) = \{j, k\}$. MVC returns $X_F' = (0, 1, 1)$, where only the delegation on issue i is changed, thus $\text{del}_{\text{MDC}}^F(\mathbf{B}) = \{i\}$. All agents $T \in \mathcal{N} \setminus \{F\}$ are direct voters, hence $\text{del}_{\text{MDC}}^T(\mathbf{B}) = \text{del}_{\text{PDC}}^T(\mathbf{B}) = \emptyset$. As $\text{top}^F(\text{del}_{\text{MDC}}^F(\mathbf{B})) \prec_F \text{top}^F(\text{del}_{\text{PDC}}^F(\mathbf{B}))$, we can conclude that $\text{del}_{\text{MDC}}^{\mathbf{B}} \preceq_{\mathcal{N}}^{\text{top}} \text{del}_{\text{PDC}}^{\mathbf{B}}$.

Now let \mathcal{N} , \mathcal{I} , and \mathcal{X}_{Γ} be as in the first part of the proof, and let profile \mathbf{B}' be composed of $B_C = (0, 1, 1)$, $B_D = (1, 0, 0)$, $B_E = (C, C, 1)$, and $B_F = (E, E, D)$, with E and F having the priority $i \succ j \succ k$. The first iteration of Algorithm 1 gives $X_E = (0, 1, 1)$ and $X_F = (\Delta, \Delta, 0)$.

In the second iteration we have $X_F = (1, 0, 0)$, as the delegations to E are inconsistent. Therefore, $\text{del}_{\text{PDC}}^C(\mathbf{B}') = \text{del}_{\text{PDC}}^D(\mathbf{B}') = \text{del}_{\text{PDC}}^E(\mathbf{B}') = \emptyset$, whereas, $\text{del}_{\text{PDC}}^F(\mathbf{B}') = \{i, j\}$. MVC however lets $X_F = (0, 1, 1)$ where $\text{del}_{\text{MDC}}^F(\mathbf{B}') = \{k\}$. As $\text{top}^F(\text{del}_{\text{MDC}}^F(\mathbf{B}')) \prec_F \text{top}^F(\text{del}_{\text{PDC}}^F(\mathbf{B}'))$, we have that $\text{del}_{\text{PDC}}^{\mathbf{B}'} \preceq_{\mathcal{N}}^{\text{top}} \text{del}_{\text{MDC}}^{\mathbf{B}'}$. \square

5 Knapsack Constraints

In this section we focus on knapsack or budget constraints, i.e., sets of feasible votes respecting a budget limit $L \in \mathbb{N}$ of the form $\Gamma_L = \{X \mid \sum_{j \in \mathcal{I}} x_j \leq L\}$. They have the property that turning a vote from acceptance to rejection in a consistent ballot preserves consistency. This property is exploited by Jain *et al.* [2021] in their algorithm to remove cycles and solve inconsistent delegations. However, this can lead to rejecting a large number of issues and potentially leaving funds unassigned. In this section we evaluate this factor for the four rules we proposed. Let $\text{count}(\mathbf{X}) = \sum_{i \in \mathcal{N}, j \in \mathcal{I}} x_{ij}$ be the number of acceptances in the profile of votes \mathbf{X} .

Remark 3. Given a profile \mathbf{B} and budget constraint Γ_L , we have that $\text{count}(\mathbf{X}^{\mathbf{B}}) \geq \text{count}(\mathcal{F}(\mathbf{B}))$ for $\mathcal{F} \in \{\text{MDC}, \text{MVC}, \text{PDC}, \text{PVC}\}$.

Proposition 10. For any profile \mathbf{B} , $\text{count}(\text{MVC}(\mathbf{B})) \geq \text{count}(\mathcal{F}(\mathbf{B}))$ for $\mathcal{F} \in \{\text{MDC}, \text{PVC}, \text{PDC}\}$.

Proof. \mathbf{B} is an arbitrary profile and Γ_L a budget constraint. MVC(\mathbf{B}) outputs all consistent profiles of votes obtained from $\mathbf{X}^{\mathbf{B}}$ with a minimal number of 1s changed to 0s. Our rules do not change 0s to 1s, as rejections do not use any of the budget. Note that if $\mathbf{X}, \mathbf{X}' \in \text{MVC}(\mathbf{B})$ then $\text{count}(\mathbf{X}) = \text{count}(\mathbf{X}')$. Thus, any other consistent delegation rule must approve at most the same number of issues as MVC(\mathbf{B}). \square

An analogous result showing that PVC accepts more issues than MDC or PDC does not hold, as can be shown by counterexample. To conclude, the approach of minimal delegation changes proposed in the literature might not be appropriate for budgeting applications, where it is outperformed by minimal vote changes in terms of number of projects accepted.

6 Conclusion

Existing approaches in the literature looked at changes to the ballots of voters to maintain consistency, in knapsack voting and in preference aggregation. We generalise this by studying liquid democracy on multiple interconnected issues, putting forward two novel ideas: first, we propose two rules that start by resolving delegations and then make changes to the final votes and, second, we design polynomial algorithms to maintain vote consistency by eliciting agents' priorities.

The main open problem is whether PDC can be improved, since Proposition 9 shows that it does not use priorities in the best way. We conjecture that this cannot be done in polynomial time, but a result showing this impossibility is yet to be shown. Our result in Section 5 shows that specific constraints require specific treatment, and thus the choice of a consistent delegation rule is not trivial. Hence a prominent direction for future research is to specify our general setting on specific classes of constraints.

Acknowledgments

The authors acknowledge the support of the ANR JCJC project SCONE (ANR 18-CE23-0009-01).

References

- [Armstrong and Larson, 2021] Ben Armstrong and Kate Larson. On the limited applicability of liquid democracy. In *Appears at the 3rd Games, Agents, and Incentives Workshop (GAIW 2021). Held as part of the Workshops at the 20th International Conference on Autonomous Agents and Multiagent Systems*, 2021.
- [Becker *et al.*, 2021] Ruben Becker, Gianlorenzo D’Angelo, Esmael Delfaraz, and Hugo Gilbert. Unveiling the truth in liquid democracy with misinformed voters. In *International Conference on Algorithmic Decision Theory (ADT)*, 2021.
- [Boldi *et al.*, 2011] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. Viscous democracy for social networks. *Communications of the ACM*, 2011.
- [Botan *et al.*, 2019] Sirin Botan, Umberto Grandi, and Laurent Perrussel. Multi-issue opinion diffusion under constraints. In *Proceedings of the 18th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2019.
- [Brill and Talmon, 2018] Markus Brill and Nimrod Talmon. Pairwise liquid democracy. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, 2018.
- [Brill *et al.*, 2022] Markus Brill, Théo Delemazure, Anne-Marie George, Martin Lackner, and Ulrike Schmidt-Kraepelin. Liquid democracy with ranked delegations. *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, 2022.
- [Caragiannis and Micha, 2019] Ioannis Caragiannis and Evi Micha. A contribution to the critique of liquid democracy. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.
- [Christoff and Grossi, 2017] Zoé Christoff and Davide Grossi. Binary voting with delegable proxy: An analysis of liquid democracy. In *Proceedings Sixteenth Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2017.
- [Dey *et al.*, 2021] Palash Dey, Arnab Maiti, and Amartya Sharma. On parameterized complexity of liquid democracy. In *Algorithms and Discrete Applied Mathematics - 7th International Conference (CALDAM)*, 2021.
- [Dhillon *et al.*, 2021] Amrita Dhillon, Grammateia Kotsialou, and Dimitris Xefteris. Information aggregation with delegation of votes. *Center for Open Science*, 2021.
- [Escoffier *et al.*, 2019] Bruno Escoffier, Hugo Gilbert, and Adèle Pass-Lanneau. The convergence of iterative delegations in liquid democracy in a social network. In *Algorithmic Game Theory - 12th International Symposium (SAGT)*, 2019.
- [Friedkin *et al.*, 2016] Noah E. Friedkin, Anton V. Proskurnikov, Roberto Tempo, and Sergey E. Parsegov. Network science on belief system dynamics under logic constraints. *Science*, 354(6310):321–326, 2016.
- [Halpern *et al.*, 2021] Daniel Halpern, Joseph Y Halpern, Ali Jadbabaie, Elchanan Mossel, Ariel D Procaccia, and Manon Revel. In defense of fluid democracy. *arXiv preprint arXiv:2107.11868*, 2021.
- [Harding, 2021] Jacqueline Harding. Proxy selection in transitive proxy voting. *Social Choice and Welfare*, pages 1–31, 2021.
- [Jain *et al.*, 2021] Pallavi Jain, Krzysztof Sornat, and Nimrod Talmon. Preserving consistency for liquid knapsack voting. In *20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2021.
- [Kahng *et al.*, 2021] Anson Kahng, Simon Mackenzie, and Ariel Procaccia. Liquid democracy: An algorithmic perspective. *Journal of Artificial Intelligence Research*, 70:1223–1252, 2021.
- [Karp, 1972] Richard M. Karp. Reducibility among combinatorial problems. In *Proceedings of a symposium on the Complexity of Computer Computations*, pages 85–103. Springer, 1972.
- [Kotsialou and Riley, 2020] Grammateia Kotsialou and Luke Riley. Incentivising participation in liquid democracy with breadth-first delegation. *AAMAS*, 2020.
- [Lang *et al.*, 2011] Jérôme Lang, Gabriella Pigozzi, Marija Slavkovic, and Leendert van der Torre. Judgment aggregation rules based on minimization. In *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, 2011.
- [Markakis and Papatotiropoulos, 2021] Evangelos Markakis and Georgios Papatotiropoulos. An approval-based model for single-step liquid democracy. In *Algorithmic Game Theory - 14th International Symposium (SAGT)*, 2021.
- [Zhang and Grossi, 2021] Yuzhe Zhang and Davide Grossi. Power in liquid democracy. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, 2021.
- [Zhang and Grossi, 2022] Yuzhe Zhang and Davide Grossi. Tracking truth by weighting proxies in liquid democracy. *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2022.