Power and truth in liquid democracy

Zhang, Yuzhe

DOI:
10.33612/diss.821820523

Publication date:
2023

The research described in this dissertation has been carried out at the Faculty of Science and Engineering, University of Groningen, the Netherlands.

Cover design by Yifei Chen and Yuzhe Zhang

university of
groningen

Power and Truth in Liquid Democracy

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. J.M.A. Scherpen
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Monday 13 November, 2023 at 12:45 hours

by

Yuzhe Zhang

born on December 22, 1992
in Zigong, China

1 INTRODUCTION

Liquid democracy is a new approach to collective decision making. As an innovative form of representative democracy, it combines the features of direct democracy and traditional representative democracy.

In recent years, the representative quality of existing institutions of representative democracy has come under scrutiny. For example, in 2022, the right to have an abortion was overturned in the U.S. by the Supreme Court, while this decision appeared to be disapproved by the majority of the citizens [18]. Against this backdrop, researchers have started to focus on innovations, among which liquid democracy is one. Though having its roots already in work on representation from the 19th century [27], it is in the last few years that liquid democracy has been attracting increasing attention from the public as well as from researchers in both political and computer science.

In this chapter, we provide an overview of this recent line of research in liquid democracy. This overview provides the context of the dissertation.

1.1 BACKGROUND AND HISTORY

Liquid democracy is a form of proxy voting [1, 21, 39, 58, 67]. In proxy voting, instead of casting a ballot, a voter may delegate her voting right to a proxy, who will represent her in the collective decision-making process. Therefore, the set of voters is naturally divided into two exclusive subsets: active voters, who act as proxies and actually cast ballots, and inactive voters, each of whom delegates to an active voter. Proxies vote with weight equal to the number of delegations obtained from the voters. Liquid democracy extends proxy voting by allowing proxies to also delegate. That is, a proxy may pass the votes she has accrued further to yet another, thereby giving rise to so-called transitive delegations. The voters who retain their votes cast their ballots, which now carry the weight given by the number of delegations they accrued. Therefore, liquid democracy is a form of proxy voting with transitive delegations.

MAIN FEATURES OF LIQUID DEMOCRACY Conceptually, a voting system equipped with the following four components is considered in the political science literature to be a liquid democracy system [12].

1. *Direct democratic component*: Each voter can directly use their vote for any issue.
2. *Flexible delegation component*: Each voter can delegate her vote to a proxy on (1) a singular issue, (2) all issues in one or more political areas, or (3) all issues in all political areas.
3. *Meta-delegation component*: Each voter who receives delegation can in turn decide to (1) cast her ballots which now carry the weight given by the number of delegations she accrued, or (2) delegate further to yet another proxy.
4. *Instant recall component*: Each voter can terminate her delegation at any time.

Example 1. *A group of 5 persons, Elina, Sarah, Jan, Renske and Ineke, are to decide which restaurant to go to for dinner. Elina suggests to go to a Chinese restaurant while Sarah would like a French one. Jan believes that Elina always has a good food taste, therefore, he delegates to Elina to make the decision on behalf of him. Renske and Ineke are good friends, and Ineke asks Renske to decide for her because Ineke is indifferent on both options.*

Subsequently, Renske first decides to delegate to Sarah, and thus Sarah collects two more votes and the majority (Sarah, Renske and Ineke) would support to go to the French restaurant (Figure 1.1, left). But when Ineke knows she would also follow the decision of Sarah, she decides to retrieve her delegation (to Sarah through Renske) because she had bad experiences before when she followed Sarah to choose restaurants. Hence, Ineke finally delegates her vote to Elina (Figure 1.1, right). The result is then reversed, such that they would go to the Chinese restaurant since the majority (Elina, Jan and Ineke) votes for it.

Example 1 intuitively shows how a liquid democracy system works: Each of the 5 persons in the example can choose to vote directly or to delegate to another to make the decision. The delegations are further transferable, for instance, initially, Ineke delegates to Renske and Renske further delegates to Sarah (Figure 1.1 left). During the process, each person can also change her delegation (e.g., Ineke) by terminating the delegation or changing her proxy. If the group would also make decisions on other issues, people may choose their proxies differently for each issue.

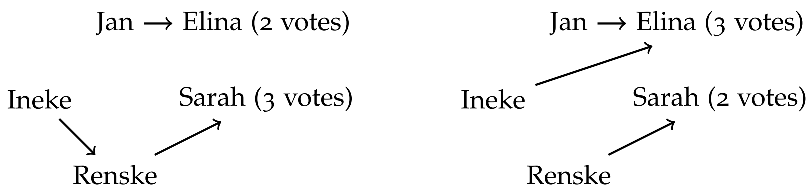


Figure 1.1: The change of the delegation structure in Example 1.

HISTORICAL NOTES Liquid democracy has been implemented in decision-support tools like LiquidFeedback[1] [9]. With the help of such online tools, several grassroots campaigns and local parties have used liquid democracy in their internal decision making, e.g., Piratenpartei[2] and LiquidFriesland[3] in Germany, Demoex[4] in Sweden. Behrens [8] provides a comprehensive overview of the historical development of liquid democracy upon which we base the brief historical overview of liquid democracy that follows.

Liquid democracy originated from the concept of transitive votes, which was first proposed by Dodgson [27][5]. The author proposed to allow voters to transfer their votes to representatives when selecting members of a house of representatives. Dodgson [27] proposed to endow representatives with the following three rights:

- directly use accrued delegations;
- further transfer received delegations to other representatives;
- leave votes unused.

Almost a century later, Tullock [66] discussed such a system in which each voter can choose herself or another voter as a representative as a form of representative democracy, and observed that it combines the characteristics of direct democracy and representative democracy. Tullock [66] also first suggested that such a democratic system can only become feasible by utilizing digital technology.

Two years later, Miller [58] extended the proposal of Tullock [66] by assuming that voters may choose different proxies for different issues, and they may also retract previous delegations at any time. However, the system proposed by Miller [58] is still closer to proxy voting, since he did not explicitly state that delegations could be further delegable.

In 1995, Lanphier [52] proposed to add the feature that each delegator can, at any time, change her proxy, or override the proxy's vote by retrieving her delegation and using the vote directly. Like Miller [58], Lanphier [52] did not explicitly consider delegations to be further delegable.

In the early 2000s, Ford [34] and Green-Armytage [38] were the first to explicitly propose the concept of transitive delegations, i.e., that proxies can further delegate the delegations they accrued to other voters, thereby explicitly identifying a key feature of the definition of liquid democracy currently in use.

1 `https://liquidfeedback.org/`
2 `https://www.piratenpartei.de/`
3 `https://www.liquidfriesland.de/`
4 `http://demoex.se/en/`
5 Charles Lutwidge Dodgson, also known by his pen name Lewis Carroll, was the author of the famous children's book *Alice in Wonderland*.

1.2 RECENT RESEARCH IN LIQUID DEMOCRACY

In this section, we summarize the main on-going research lines in liquid democracy. We base our overview on [62].

1.2.1 Truth Tracking

One of the main lines of research on liquid democracy concerns its performance on collective truth tracking, i.e., the ability of a group of voters to identify (for instance, via majority rule) the ”correct” alternative in a set of two (or more) alternatives, which is one of the standard settings in the study of wisdom of the crowd phenomena [24]. The typical example is jury decision making, where a jury needs to decide whether a defendant is guilty or not. In these settings, a tradition on "jury theorems" [24] has shown that if jurors are sufficiently competent and sufficiently independent, crowds are "wise" in the sense of becoming infallible if the group becomes large enough.

Research has shown that liquid democracy may undermine the wisdom of the crowd by correlating voters' errors via delegation. Kahng *et al.* [46] focus on liquid democracy's truth-tracking performance under certain classes of delegation mechanisms, i.e., mechanisms that describe how voters delegate. They mainly address two truth-tracking related properties: Do No Harm (DNH), i.e., a liquid democracy mechanism is not worse than direct voting with high probability, and Positive Gain (PG), i.e., a liquid democracy mechanism can outperform direct voting with high probability. The authors draw relatively pessimistic conclusions in terms of these two properties, especially when voters decide delegation strategies based only on the information of their neighbors in a social network. However, the truth-tracking performance can be improved by adding to delegation mechanisms a centralized component which restricts the number of delegations each voter receives. We will present this model with more details later in Chapter 2.

Motivated by the centralized component in [46], Gölz *et al.* [37] study the problem which selects for each voter at most one proxy given a set of potential proxies, so to minimize the maximum of agents' accrued number of delegations. They show that it is NP-hard to solve this problem. However, it becomes feasible when using some delegation mechanisms with a mixed delegation model.

Caragiannis & Micha [16] extend the model of [46] by assuming that mis-informed agents may delegate to even more mis-informed ones, and further strengthen the pessimistic conclusions of [46]. They then show that it is computationally intractable to find the delegation structure that maximizes truth-tracking performance when voters' delegations are restricted by a social network. Becker *et al.* [7] provide further evidence to support the results of [16], however, Becker *et al.* [7] also show that under specific conditions, polynomial time algorithms exist to find some optimal delegation structures.

While the above-mentioned literature mainly sketches a pessimistic perspective on liquid democracy's truth-tracking performance, Halpern *et al.* [41] provide a more optimistic picture. They study natural delegation mechanisms for liquid democracy, where they assume that each agent may choose to directly vote or to delegate with some probability, according to their expertise on the issue. If the agent chooses to delegate, she chooses her proxy by a stochastic strategy based on the expertise of her potential proxies. The authors prove that, with high probability, the DNH and PG properties proposed by Kahng *et al.* [46] are satisfied under specific conditions to characterize crowd wisdom in liquid democracy of [16].

Butterworth & Booth [15] then further provide philosophical arguments to address this optimistic perspective with practical considerations. They specify that crux settings in [46] and [16] do not happen frequently in the real world, i.e., delegations are usually motivated by trust and reputation but not by seeking for a better decision. Moreover, they also point out that examples in [16] are extreme cases and it is not necessary to maximize the truth-tracking performance if the collective decision making is guaranteed to be correct in many cases.

Alouf-Heffetz *et al.* [2] study the truth-tracking performance of liquid democracy empirically by simulating voters' delegating behaviour in generated social networks as well as real-world social networks sampled from Facebook users. Due to the computational hardness of finding the best delegation profile with respect to truth tracking [16], Alouf-Heffetz *et al.* [2] use a simulated annealing (SA) algorithm to approximate the best delegation profile in a given social network. They show that when a high ratio of voters delegate, even delegation mechanisms where voters can only noisily delegate to a better neighbor can perform similarly to the SA algorithm. Their simulation results support the claim that delegation can considerably improve voters truth-tracking performance, especially when the variance of voters' individual competences is high. Interestingly, their results do not reflect significant influence of network structure on liquid democracy's truth-tracking performance, except that liquid democracy's truth-tracking performance is weakened in extremely sparse social networks.

1.2.2 Ballot Consistency

Liquid democracy can also be applied to preference aggregation settings when there are more than two alternatives. For example, when ballots are ordinal rankings of alternatives, the flexible delegation component of liquid democracy may allow voters to delegate different pairwise comparisons to different voters. This flexibility can in turn introduce inconsistency into the liquid democracy system, by violating the transitive property of ballots.

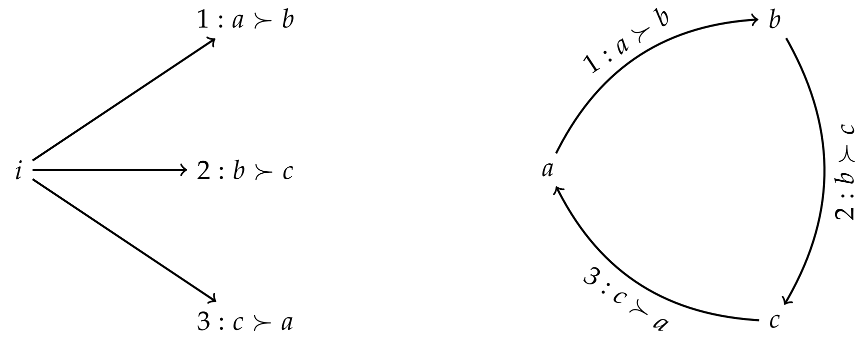


Figure 1.2: Inconsistency introduced in Example 2. Left: Voter i delegates the pairwise comparisons to three different voters. Right: The aggregated ballot of voter i becomes a cyclic preference (inconsistency introduced).

Example 2. *Consider a voting with three alternatives $\{a, b, c\}$. Voter i delegates the decisions on the pairwise comparisons of these three alternatives to three different voters, say the comparison between a and b to voter 1, the comparison between b and c to voter 2, and the comparison between a and c to voter 3. Then inconsistency occurs, for example, if voter 1 submits $a \succ b$, voter 2 submits $b \succ c$, while voter 3 submits $c \succ a$. This makes the ballot of voter i a cyclic ranking of a, b and c, as shown in Figure 1.2.*

Brill & Talmon [14] formalize the inconsistency problem as pairwise delegative systems, and show that it is computationally intractable to detect an inconsistent ballot. They further propose several methods to cope with the inconsistency problem, e.g., modifying delegations, or using voting rules to aggregate ranked delegations. Inconsistency problems in liquid democracy have also been studied in other voting settings, such as participatory budgeting [45] and judgement aggregation [20, 23].

Christoff & Grossi [20] propose to resolve inconsistencies by treating liquid democracy as a process of ballot copying rather than of vote delegation. Individuals would then refrain from copying ballots when these give rise to inconsistencies.

Colley & Grandi [23] propose two polynomial time algorithms to resolve inconsistent ballot profiles by allowing voters to submit ranked delegations (which are also proposed in the aforementioned [14]) over issues, such that, in order to meet the constraints, the algorithms can modify voters' delegations/votes according to their priorities.

1.2.3 Game-theoretical Analysis

From a different viewpoint, Bloembergen *et al.* [11] formalize liquid democracy in a game-theoretic model and define so-called delegation games. They assume that, under a costly binary voting setting, each agent benefits from only one of two alternatives but the agent does not necessarily know which one. To correctly identify the best alternative for them, agents incur a cost. They can then choose to vote directly incurring that cost, or to delegate to a proxy at no cost, hoping that the proxies can cast a more accurate ballot for them. Bloembergen *et al.* [11] then study the existence of Nash equilibria in this type of interactions, i.e., a delegation profile such that no agent can obtain a better outcome by deviating from it, in general and for several subclasses of delegation games. We will present this model in more details later (in Chapter 2).

Escoffier *et al.* [31] extend the above game-theoretic model. They assume that each agent has an ordinal preference over all the other agents. This preference denotes to whom the agent prefers to delegate. They study the existence of Nash equilibria in such preference profiles. Escoffier *et al.* [31] show that generally, Nash equilibria cannot be guaranteed to exist, and it is intractable to verify the existence of a Nash equilibrium given a preference profile. However, they prove that if the social network is a tree, it costs polynomial time to verify the existence of Nash equilibria and whether a Nash equilibrium satisfies certain properties, such as minimizing the dissatisfaction of voters and minimizing the maximum voting power of a voter who does not delegate.

1.2.4 Delegation Cycle

Delegation cycles, in which agents delegate back to themselves through other proxies, is a much debated issue in liquid democracy since it introduces a lack of representation for the agents in the cycle and those delegating to them.

Example 3 (Example 1 continued)**.** *We continue to consider the original delegation structure (left in Figure 1.1) in Example 1. If Sarah changes her mind to delegate to Ineke, a delegation cycle is formed: Ineke delegates to Renske, Renske delegates to Sarah, and Sarah further delegates back to Ineke. This results in that none of Ineke, Renske and Sarah votes (loss of representation).*

Christoff & Grossi [20] treat cycles precisely as a loss of representation in that they induce abstentions by all agents involved in delegation cycles and those delegating to them. They propose to solve delegation cycles by overruling them with default values agents would be requested to cast even when delegating.

Several other researches approach this problem by allowing agents to point to multiple other agents as their potential proxies and provide rankings over them.

Given this information, so-called delegation mechanisms are then used to select one proxy for each agent to avoid forming cycles in the final delegation structure.

To determine better delegation for each agent, Kavitha *et al.* [48] study the branching problem which aims at finding a subgraph where each node has at most one in-degree (therefore the tail node is the proxy) for a given directed graph. They show that given a preference over in-degrees for each node, it is polynomial-time computable to verify the existence of a popular branching, i.e., no majority of nodes prefer another branching, and find it.

Kotsialou & Riley [50] study two classes of such delegation mechanisms: Depth-first mechanisms (DFMs) and Breadth-first mechanisms (BFMs). DFMs assign delegations to agents that are as preferred by them as possible, while BFMs prioritize shorter delegation chains (i.e., the transitive chains from delegators to the agents who eventually use the delegations). The authors show that BFMs can resolve delegation cycles.

Brill *et al.* [13] further specify more general classes of delegation mechanisms, of which DFMs and BFMs are special cases. Then, the authors investigate these mechanism classes by verifying the axioms that the mechanisms satisfy. Lastly, Brill *et al.* [13] also develop an axiomatic characterization for DFMs and BFMs, respectively.

1.2.5 Data-driven Analysis

Kling *et al.* [49] is, to date, the only study of liquid democracy based on real-world data. They investigate delegating behaviour by analysing long-term data from the platform LiquidFeedback, which was in use by the German Pirate Party during 2009-2013. The authors are mainly interested in the voting power distribution resulting from delegations, especially the emergence of super-voters (i.e., voters accruing high levels of voting power).

They show that super-voters do appear frequently. However, most of those super-voters vote in accordance with the majority. That is, although super-voters exist, they usually do not manipulate the system by abusing their voting power. To measure the power accrued by voters, Kling *et al.* [49] develop variants of voting power indices, such as the Banzhaf index [5] and the Shapley-Shubik index [64], inspired by their empirical findings. We will discuss these indices in Chapter 2.

1.3 RESEARCH CONTRIBUTIONS

Taking inspiration from some of the above lines of research, in this dissertation, we address three questions:

1. How can we measure voting power in liquid democracy?

2. Can we improve liquid democracy's truth-tracking performance if we allow agents to split their delegations across several proxies?

3. How do agents behave in delegation games when they care about power or are allowed to split delegations?

We turn now to a more detailed presentation of these questions.

1.3.1 How Can We Measure Voting Power in Liquid Democracy?

Voting power is a controversial issue in liquid democracy, as the system may suffer from manipulation if a small subset of agents accrue a large number of delegations due to the flexibility endowed by the system. As mentioned above, Kling *et al.* [49] propose to adapt existing voting power indices, the Banzhaf power index [5, 63] and the Shapley-Shubik power index [64], to liquid democracy in order to measure the voting power of the agents that do not delegate.

However, their analysis does not capture the instant recall component (recall Section 1.1) of liquid democracy. Since agents are allowed to change their delegations during the process leading to a vote, delegators may still exert significant influence on the voting results.

The first contribution of this dissertation is to develop a novel voting power index, which is a strict generalization of the Banzhaf power index, for liquid democracy. By using this power index, we can capture the influence of both delegators and voters who eventually cast ballots. We provide an axiomatic characterization of this power index, and study its properties.

1.3.2 Can We Improve Liquid Democracy's Truth-tracking Performance If We Allow Agents To Split Their Delegations Across Several Proxies?

As reported above, whether liquid democracy helps or hinders the wisdom of the crowd is an open issue in the literature.

Shapley & Grofman [65] prove that the wisdom of the crowd is maximal if voting weight can be re-distributed among agents in a way dependent on their individual accuracies. Inspired by their work, we study liquid democracy as a mechanism to redistribute voting weight through delegations in a way that approaches the optimal voting weight distribution shown in [65].

In order to achieve such an aim, we assume that agents may divide their voting weight and delegate it to multiple proxies. Based on this assumption, we propose two ways to interpret the agents' apportioning of weights, and design centralized delegation mechanisms, which output the optimal voting weight distribution in specific classes of social networks.

1.3.3 How Do Agents Behave in Delegation Games When They Care About Power or Are Allowed To Split Delegations?

We incorporate the proposals answering questions in Section 1.3.1 and Section 1.3.2 into the game-theoretic model of [11], and investigate their influence on agents' delegation strategies from a game theory point of view.

In the delegation games defined by Bloembergen *et al.* [11], agents aim at delegating to more accurate agents, i.e., agents with higher probability to make the correct decision, so as to rely on them to vote correctly. We investigate two types of games based on that work:

1. *Delegation games sensitive to power.* In these games, we assume that, while agents try to delegate to more accurate agents, they also have incentives to retain voting power in the liquid democracy system. This extra motivation considerably changes agents' delegating behavior, and we obtain considerably different results from those in [11]. We analyze the existence of Nash equilibria in these delegation games and find that Nash equilibria are not guaranteed to exist in general, but they can be shown to exist in several special classes of delegation games.

2. *Delegation games with weighted delegations.* In these delegation games, we assume that agents can partition their voting weight and delegate it to multiple proxies. Since agents are allowed to take weighted delegation strategies, equilibria are always guaranteed to exist [59]. We look at the delegation structures in equilibria under this new assumption, as well as the quality of such equilibria in terms of truth-tracking accuracy.

Finally, for specific classes of the game-theoretic models at points 1 and 2 above, we study how agents behave in random social networks, by computer simulations.

1. *Experiments on delegation games sensitive to power.* We investigate agents' delegation behavior by manipulating their incentives to retain voting power in the delegation games. To model agents' behavior, we utilize standard equilibrium computation dynamics (such as better response dynamics [59]) by adding the power-sensitive motivation into the model of [11]. We find that, in general, delegations are less frequent when the connectivity of the social network is sparse or agents' incentives to retain voting power are strong.

2. *Experiments on delegation games with weighted delegations.* We use the similar equilibrium computation dynamics as above to simulate agents' behavior in the weighted delegation game model generalizing [11]. We observe that in the dynamics, agents may distribute their weighted delegations over proxies differently: from uniform distributions to concentrations on high-accuracy

proxies. Subsequently a variety of measures, including one concerning truth-tracking performance, are computed to compare the performance of weighted delegations. We find that the truth-tracking performance of liquid democracy is improved when the distribution of delegations is less concentrated.

1.4 OUTLINE OF THE DISSERTATION

This dissertation consists of three parts: Part I Background (Chapter 1 and Chapter 2), Part II (Chapter 3 and Chapter 4), and Part III (Chapter 5 and Chapter 6). In Chapter 2, we introduce the thesis background and details on the existing research on liquid democracy, upon which we base our research. In Chapter 3, we present our voting power index for liquid democracy and study its axiomatic properties. Then, in Chapter 4, the proposed voting power index is incorporated into delegation games, and both theoretical and empirical results on the resulting game-theoretic model are presented.

In Chapter 5, we present our theory of weighted delegations, where agents can partition their voting weights and delegate to multiple proxies. Two ways of interpreting weighted delegations are provided, and we show that optimal delegations in terms of truth tracking become possible unlike in the standard case. Then, in Chapter 6, we conduct a similar game-theoretic analysis as in Chapter 4 on delegation games with weighted delegations.

Finally, Chapter 7 concludes the dissertation by summarizing its findings and outlining an agenda for future research.

The structure of the dissertation is shown in Figure 1.3.

1.5 SOURCES OF CHAPTERS

The contents in this dissertation are based on the following papers:

1. Yuzhe Zhang and Davide Grossi. *Power in liquid democracy.* In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 6, pp. 5822-5830. 2021.

2. Yuzhe Zhang and Davide Grossi. *Power in Liquid Democracy: Theory and Computational Experiments*, to be submitted (extended improved version of 1).

3. Yuzhe Zhang and Davide Grossi. *Tracking Truth by Weighting Proxies in Liquid Democracy.* the 8th International Workshop on Computational Social Choice, 2021.

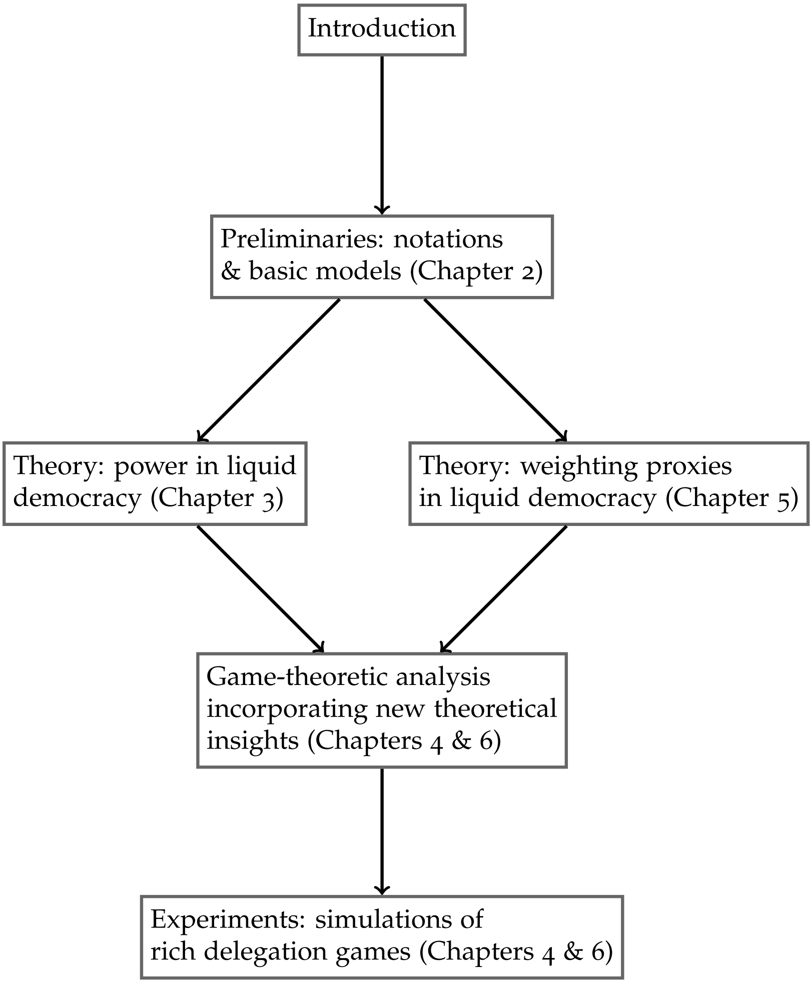


Figure 1.3: Dissertation structure and main contributions: different theories of liquid democracy are developed in Part II and Part III, however, similar game-theoretic analysis and empirical study are conducted in both parts.

4. Yuzhe Zhang and Davide Grossi. *Tracking Truth by Weighting Proxies in Liquid Democracy.* In Proceedings of the 21st International Conference on Autonomous Agents and Multi-agent Systems, pp. 1482-1490. 2022.

5. Yuzhe Zhang and Davide Grossi, *Weighted Delegations in Search for the Truth: Liquid Democracy by Weighting Proxies*, to be submitted (extended and improved version of 3 and 4).

The material presented in Chapters 3 and 4 is collected in paper 2, which is an extended version of paper 1.

The material presented in Chapters 5 and 6 is collected in publication 5, which is an extended version of papers 3 and 4.

Papers published during my Ph.D. trajectory but not included in this dissertation are:

- Takamasa Suzuki, Akihisa Tamura, Kentaro Yahiro, Makoto Yokoo, and Yuzhe Zhang. *Strategyproof Allocation Mechanisms with Endowments and M-convex Distributional Constraints.* Artificial Intelligence 315 (2023): 103825.

- Kentaro Yahiro, Yuzhe Zhang, Nathanaël Barrot, and Makoto Yokoo. *Strategyproof and fair matching mechanism for ratio constraints.* Autonomous Agents and Multi-Agent Systems 34, no. 1 (2020): 1-29.

- Anisse Ismaili, Naoto Hamada, Yuzhe Zhang, Takamasa Suzuki, and Makoto Yokoo. *Weighted matching markets with budget constraints.* Journal of Artificial Intelligence Research 65 (2019): 393-421.

2 PRELIMINARIES

In this chapter, we first introduce the basic model of liquid democracy, based on which we develop all theoretical and simulation results in this dissertation. Then we introduce the theory of power indices and wisdom of the crowd, which are the basis of our proposed theory. Finally, we review two lines of research that form the background of our contribution to liquid democracy: one assumes that agents' ultimate goal is to delegate to agents with good expertise; while in the other, the focus is on the overall accuracy of group decisions.

2.1 A LIQUID DEMOCRACY MODEL IN BINARY VOTING

Our model is based on the binary voting setting for truth-tracking [24, 30, 40]. The setting has already been applied to the study of liquid democracy by Bloembergen *et al.* [11], Caragiannis & Micha [16], and Kahng *et al.* [46].

In binary voting, a finite set of agents $N = \{1, 2, \ldots, n\}$, initialized with a *voting weight profile* $\mathbf{w} = (w(1), \ldots, w(n)) \in \mathbb{R}^N_{\geq 0}$,[1] has to vote on whether to accept (1) or reject (0) an issue. The vote is supposed to track the correct state of the world—that is whether it is "best" to accept or reject the issue. Let θ denote this correct state, which we call the *truth*. Then we assume that both alternatives have equal prior of being the truth, i.e., $\Pr(\theta = 1) = \Pr(\theta = 0) = 0.5$.

Let $\mathbf{v} = (v_1, \ldots, v_n)$, where $v_i \in \{0, 1\}$, denote the *ballot profile*, which records the ballot cast by each agent $i \in N$. Then an agent's ability to make the correct choice (i.e., the agent's error model) is represented by the agent's *accuracy* $q_i = \Pr(v_i = x \mid \theta = x)$, where $x \in \{0, 1\}$, and we call the vector $\mathbf{q} = (q_1, \ldots, q_n) \in \mathbb{R}^N$ the *accuracy profile*. We assume that $q_i \in (0.5, 1]$, i.e., each agent is strictly more accurate than a random decision ($q = 0.5$). Agents' accuracies are also assumed to be independent, that is, for any pair of agents $i, j \in N$, the probability that they both cast the correct ballot is $\Pr(v_i = 1, v_j = 1) = q_i q_j$.

In this model, we divide the mechanism of liquid democracy into two phases: the delegation phase and the voting phase. In the delegation phase, all agents decide to whom to delegate their vote (*delegation strategy*), and then in the voting phase, all agents who do not delegate vote with weight corresponding to their accrued delegations.

---

1 We normally call the setting one-person-one-vote when $w(i) = 1$ for all $i \in N$.

DELEGATION PROFILES The choices made in the delegation phase determine what we call a delegation profile. When agent $i \in N$ delegates to agent $j \in N$ we write $d_i = j$. Then $\mathbf{d} = (d_1, d_2, \ldots, d_n)$ is called a *delegation profile* (or simply *profile*) and is a vector describing each agent's delegation strategy. Equivalently, delegation profiles can be usefully thought of as maps $\mathbf{d} : N \to N$, where $\mathbf{d}(i) = d_i$. When $d_i = i$, agent i votes on her own behalf. We call such an agent a *guru*. On the other hand, any agent who is not a guru is called a *delegator*. For profile $\mathbf{d}$, and $C \subseteq N$, let $Gu(C, \mathbf{d})$ denote all gurus in C in profile $\mathbf{d}$, i.e., $Gu(C, \mathbf{d}) = \{i \in C \mid d_i = i\}$. We simply write $Gu(\mathbf{d})$ instead of $Gu(N, \mathbf{d})$ to denote all gurus in N. A delegation profile in which all agents are gurus (i.e., for all $i \in N$, $d_i = i$) is said to be *trivial*.

CHAINS AND CYCLES Any profile $\mathbf{d}$ can also be represented by a directed graph $\langle N, E(\mathbf{d})\rangle$. An edge from agent i to j ($i \to j$) exists whenever $d_i = j$, i.e., $E(\mathbf{d}) = \{(i,j) \in N \times N \mid \mathbf{d}(i) = j\}$. Given a profile $\mathbf{d}$, we call such a directed graph the *delegation graph* of profile $\mathbf{d}$. Consider then a profile $\mathbf{d}$ where a path exists from i_1 to i_k, i.e., $i_1 \to i_2 \to \cdots \to i_k$. We call such paths *delegation chains*. When such a chain from agent i_1 to guru i_k exists, every agent on this delegation chain (indirectly) delegates to i_k, and we denote i_1's guru by $d^*_{i_1} = \mathbf{d}^*(i_1) = i_k$. In delegation graphs, the gurus are linked to themselves via a *loop*, i.e., a delegation chain with length of 1 and coincident head and tail. Note that we sometimes do not draw these loops in delegation graphs when the context is clear. Additionally, the set of agents between any pair of agents on a delegation chain are called the *intermediaries* between the two agents. For example, suppose the above delegation chain occurs in profile $\mathbf{d}$. Then the set of intermediaries between i_1 and i_k is $\{i_2, \ldots, i_{k-1}\}$, and it is denoted by $\delta_{\mathbf{d}}(i_1, i_k)$. Between agents i_1 and i_k, the number of intermediaries plus the terminal i_k, is called the *delegation distance* from i_1 to i_k and is denoted by $\Delta_{\mathbf{d}}(i,j) = |\delta_{\mathbf{d}}(i_1, i_k) \cup \{i_k\}|$.

A *delegation cycle* is a delegation chain with a length more than 1 where the first and last agents coincide, e.g., $i_1 = i_k$ in the above delegation chain. In such a case, no agent in the chain is linked to a guru unless the chain is a one-element loop from i_1 to herself. Therefore neither does any agent linked via a delegation chain to an agent in a delegation cycle have a guru. For $C \subseteq N$, we write $De(C, \mathbf{d}) = \{j \in N \mid d^*_j \in Gu(C, \mathbf{d})\}$ to denote the set of delegators that directly or indirectly delegate to a guru in C, and all gurus in C. If $C = \{i\}$ we simply write $De(i, \mathbf{d})$ for the set of agents whose guru is i.

The weight accrued by an agent via delegations in $\mathbf{d}$ is:

$$w(i, \mathbf{d}) = \begin{cases} \sum_{j \in De(i,\mathbf{d})} w(i) & \text{if } i \in Gu(\mathbf{d}) \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

As gurus are the only ones voting in $\mathbf{d}$, they accrue the weight transferred by delegators through delegation chains. Observe that because of delegation cycles, $\sum_{i \in N} w(i, \mathbf{d})$ may be smaller than n. That is, voting weight is lost if some agents are caught in delegation cycles. This is in line with the intuition that agents not linked to a guru fail to relay their votes to the voting mechanism.

UNDERLYING NETWORK Finally, for the delegation phase, we assume delegations to be constrained by a network represented by an undirected graph $R = \langle N, E \rangle$, where N is the set of agents, and $E \subseteq N \times N$. For $i \in N$, $R(i)$ denotes the *neighborhood* of i, i.e., $R(i) = \{i\} \cup \{j \in N \mid (i,j) \in E\}$. Agents are able to delegate only to agents in their neighborhoods. We write $R'(i)$ to denote $R(i) \backslash \{i\}$ and $\mathcal{R}^N$ to denote all networks with vertices N.

An alternative representation of networks is based on directed graph, which are a generalization of undirected graph. We can also denote a directed graph by a tuple $R = \langle N, E \rangle$, where each agent is a node in the network and for any $i, j \in N$ ($i \neq j$), $(i, j) \in E$ if there is an *directed* edge from i to j. Then j is called a *neighbor* of agent i. We also use $R'(i)$ to denote all neighbors of agent i, i.e., $R'(i) = \{j \in N \mid (i,j) \in E\}$ and $R(i) = R'(i) \cup \{i\}$. Note that we will be working with this more general network representation in Part II.

In the rest of this dissertation, we use a few specific classes of networks. A *complete network* of the undirected graph class is a network $R = \langle N, E \rangle$ where all pairs of nodes are linked. In the directed graph class, this requires that between each pair of nodes, there exist two edges with different directions. A *connected network* of the undirected graph class is a network R in which for each pair of nodes, we can always find an undirected path linking them. However, in the directed graph class, in a connected network, from each node, we can always find a directed path to each of the other nodes. In the experiments of this dissertation (i.e., in Section 4.3 and Section 6.2), we generate random networks as the underlying networks. A random network is generated by specifying a probability p, such that for any agent $i \in N$, she is linked with any other agent $j \in N \setminus \{i\}$ with probability p.

VOTING WITH ACCRUED WEIGHT Once delegations are settled, liquid democracy results in weighted voting where only gurus ($Gu(\mathbf{d})$) vote, in the following voting phase, with the sum of weights they accrued from direct or indirect delegations in profile $\mathbf{d}$, i.e., $w(i, \mathbf{d})$ for all $i \in Gu(\mathbf{d})$ by Equation 2.1.

Then we assume that the weighted majority rule is utilized to determine the result. That is, the issue is accepted if

$$\sum_{i \in Gu(\mathbf{d})} w(i, \mathbf{d}) v_i > \frac{n}{2}, \tag{2.2}$$

otherwise the issue is rejected. Notice that Equation 2.2 decides whether to accept or reject an issue deterministically, and the decision is biased towards rejecting the issue. Sometimes, we also assume that ties are randomly broken when $\sum_{i \in Gu(\mathbf{d})} w(i, \mathbf{d}) v_i = \frac{n}{2}$, i.e., in case of a tie, the issue is rejected or accepted uniformly at random. We use both random and deterministic versions in the dissertation depending on the context. Note also that delegation cycles make it harder to accept an issue, since they lead to loss of voting weight. Notice that when $\mathbf{d}$ is trivial, i.e., each agent is a guru, the voting phase reduces *weighted voting*, where each agent votes with their initialized voting weight.

We use the following example to briefly illustrate the above concepts.

Example 4. *Consider a set of agents $N = \{1, 2, 3, 4, 5, 6, 7\}$ with initialized weight $w(i) = 1$ for all $i \in N$ and assume the delegation profile $\mathbf{d}$ in Figure 2.1. Observe that the only gurus in the profile are $Gu(\mathbf{d}) = \{1, 7\}$, and agents 3 and 2 delegate to 1 on a delegation chain $3 \to 2 \to 1$. Therefore $\mathbf{d}^*(3) = \mathbf{d}^*(2) = 1$ and 2 is the only intermediary between agent 1 and 3, i.e., $\delta_{\mathbf{d}}(1, 3) = \{2\}$.*

In the middle component of the delegation graph, agents 4 and 5 form a delegation cycle. Hence they have no guru to represent them. Since agent 6 delegates to agent 5, who is in the delegation cycle, agent 6 is also caught in the cycle and does not have a guru.

Then in the voting phase, gurus 1 and 7 vote, with weight $w(1, \mathbf{d}) = 3$ and $w(7, \mathbf{d}) = 1$. Therefore in this weighted voting, the issue can be accepted if and only if both 1 and 7 vote for accepting it since $n = 7$ and $w(1, \mathbf{d}) + w(7, \mathbf{d}) = 4 > \frac{n}{2}$.

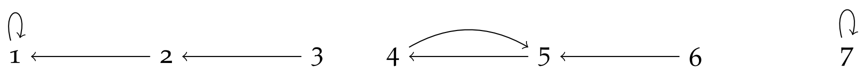


Figure 2.1: Delegation graph of Example 4

2.2 MEASURES OF VOTING POWER

From a voting perspective, gurus are the only agents who retain voting power after the delegation phase. To study voting power in our model (Part II), we introduce here two well-known power indices, the Banzhaf index [5, 63] and Shapley-Shubik power index [64], which measure agents' (different) ability to influence the final voting result.

To introduce these power indices, we first introduce the notion of *simple games* [19].

SIMPLE GAMES A simple game is a tuple $\mathcal{G} = \langle N, f \rangle$, where f is the characteristic function $f : 2^N \to \{0, 1\}$. For all $C \subseteq N$, if $f(C) = 1$ the coalition C is called *winning*, otherwise it is called *losing*. An agent i ($i \in C$) is called a *swing agent* for coalition C if $f(C) - f(C \backslash \{i\}) = 1$. Normally, such characteristic functions are assumed to be *monotone*, i.e., for all $C_1, C_2 \subseteq N$ such that $C_1 \subseteq C_2$, $f(C_1) \leq f(C_2)$.

BANZHAF POWER INDEX Given a simple game $\mathcal{G}$, for all $i \in N$, i's Banzhaf index is

$$\mathsf{B}_i(\mathcal{G}) = \frac{1}{2^{n-1}} \sum_{C \subseteq N \backslash \{i\}} (f(C \cup \{i\}) - f(C)), \tag{2.3}$$

where $n = |N|$. Intuitively i's Banzhaf index is the probability that i is a swing agent for a uniformly random coalition.

SHAPLEY-SHUBIK POWER INDEX Instead of considering a random coalition in N, the Shapley-Shubik index measures an agent's probability to swing in a random permutation sequence of N. That is, agent i's Shapley-Shubik index in simple game $\mathcal{G}$ is given as

$$\mathsf{SS}_i(\mathcal{G}) = \sum_{C \subseteq N \backslash \{i\}} \frac{c!(n-c-1)!}{n!} (f(C \cup \{i\}) - f(C)), \tag{2.4}$$

where $c = |C|$. Observe that the Shapley-Shubik index computes the proportion of permutation sequences in which i swings for the coalition consisting of herself and all her predecessors.

Example 5. *Consider agents* $N = \{1,2,3,4\}$ *and initial voting weight* $\mathbf{w} = (3,4,2,2)$*, so for example, agent* 1 *can cast 3 votes for an alternative. Assume that the winner is decided by weighted majority, i.e., alternative* 1 *wins if and only if* $\sum_{i \in N} v_i w(i) \geq 6$*, otherwise it loses.*

We first compute the Banzhaf index of agent 3 *as an example. She is swing for coalitions* $C_0 = \{2,3\}$ *and* $C_1 = \{1,3,4\}$*, therefore,* $\mathsf{B}_3(\mathcal{G}) = \frac{2}{2^3} = \frac{1}{4}$.

To compute $\mathsf{SS}_3(\mathcal{G})$*, we check whether* 3 *is swing for all permutation sequences of* N*. For instance in sequence* $(1,2,3,4)$*, we count agents' weight in a round-robin order:*

1. *The coalition consisting of the first agent is* $\{1\}$*, and* $w(3) = 3$*, indicating* $\{1\}$ *is losing;*
2. *The coalition consisting of the first and second agents is* $\{1,2\}$*, and* $w(1) + w(2) = 7$*, therefore* $\{1,4\}$ *is already winning.*

Hence we conclude that agent 3 *is not a swing agent for this sequence.*

However, for sequence $(2,3,1,4)$*, since* $\{2\}$ *is a losing coalition, while* $\{2,3\}$ *is winning, agent* 3 *swings for this sequence. The set of sequences for which agent* 3 *swings is then* $\{(2,3,1,4),(2,3,4,1),(1,4,3,2),(4,1,3,2)\}$*, by which we obtain that the Shapley-Shubik index of agent* 3 *is* $\mathsf{SS}_3(\mathcal{G}) = \frac{4}{4!} = \frac{1}{6}$.

2.3 WISDOM OF THE CROWD

In the truth-tracking perspective on binary voting, we are concerned about a group of agents' ability to reveal the truth using majority voting. That is, what is the probability that at least half of the agents choose the correct alternative? Note that in this dissertation we assume that agents are initially pairwise independent, i.e., for any pair of agents $i, j \in N$, the probability that they both choose the true state is $\Pr(v_i = v_j = \theta) = q_i q_j$. Then, based on the assumption of independence, the concept of *group accuracy* is formulated as follows.

Definition 1 (Group accuracy). *Given a set of agents N and an accuracy profile* $\mathbf{q}$*, we call* group accuracy*, denoted as* q_N*, the probability that the (weighted majority) winner is* θ*, i.e.,*

$$q_N = \sum_{C \in W} \prod_{i \in C} q_i \prod_{i \in N \setminus C} (1 - q_i), \tag{2.5}$$

where W is the set of winning coalitions*. A coalition* $C \subseteq N$ *is winning if its weight is larger than the weight of its complement coalition* $N \setminus C$*, or if its weight is equal to its complement coalition and it is selected uniformly at random.*

That is, when deciding whether a coalition is a winning coalition, we break ties ($\sum_{i \in C} w(i) = \sum_{i \in N \setminus C} w(i)$) uniformly at random. We also say that a specific coalition C's accuracy is $\prod_{i \in C} q_i \prod_{i \in N \setminus C}(1 - q_i)$, where $C \subseteq N$.

CONDORCET THEOREM The Condorcet theorem [24] is the simplest formulation of the wisdom of the crowd. The theorem assumes that each agent has the same individual accuracy. It then states that when each agent is well-informed, i.e., with individual accuracy that is higher than 0.5, a group of agents who have the same accuracy always make a better decision than each individual, and the group accuracy monotonically increases with the group size. However, when each agent is mis-informed, i.e., with individual accuracy lower than 0.5, a group of agents who have the same accuracy always make a worse decision than each individual, and the group accuracy monotonically decreases with the group size. Note that the Condorcet theorem also assumes that $|N|$ is odd to avoid ties.

Theorem 1 (Condorcet jury theorem [24]). *Let* $|N|$ *be odd and* $w(i) = 1$ *for all* $i \in N$*. Then:*

1. *if* $q_i = q \in (0.5, 1]$*, we have* $q_N \geq q$*, and* q_N *monotonically increases in N and* $\lim_{|N| \to \infty} q_N = 1$.
2. *if* $q_i = q \in [0, 0.5)$*, we have* $q_N \leq q$*, and* q_N *monotonically decreases in N and* $\lim_{|N| \to \infty} q_N = 0$.

CONDORCET THEOREM FOR HETEROGENEOUS AGENTS A natural generalization of Theorem 1 is then to consider the group accuracy in a heterogeneous setting, where each agent has a different individual accuracy.

Feld & Grofman [32] provided several results on the group accuracy of the combined group when adding one agent to an existing set.

Theorem 2 ([32]). *Given a set of agents N, where $|N| = n$ is odd, and $w(i) = i$ and $q_i \in (0.5, 1]$ for all $i \in N$, and a single agent $j \notin N$, with $q_j \in (0.5, 1]$, for the combined set $N' = N \cup \{j\}$, we have $q_{N'} \geq q_N$ if either of the following conditions holds:*

1. *The* average *individual accuracy, $\frac{1}{n}\sum_{i \in N} q_i$, of N is lower than q_j.*
2. $q_j > q_N$.

Berend & Paroush [10] developed a general result for the asymptotic part of Theorem 2.

Theorem 3 ([10]). $\lim_{n\to\infty} q_N = 1$ *holds if and only if at least one of the following conditions holds:*

1. $\lim_{n\to\infty} \frac{\sum_{i=1}^{n} q_i - n/2}{\sqrt{\sum_{i=1}^{n} q_i(1-q_i)}} \to \infty$.
2. *For every sufficiently large n, $|\{i \in N \mid q_i = 1\}| \geq \frac{n}{2}$.*

CONDORCET THEOREM FOR WEIGHTED VOTING Based on the above setting of heterogeneous agents, Shapley & Grofman [65] further extended the Condorcet theorem to investigate weighted voting, instead of the one-person-one-vote setting in the above theorems of group accuracy. They provided an optimal way of assigning voting weight among agents based on the agents' individual accuracies. We also provide the proof for the following theorem, which is relevant for our results in Part III.

Theorem 4 ([65]). *For a set of agents N such that $q_i \in (0.5, 1]$ for all $i \in N$, q_N is maximal if for each agent $i \in N$, $w(i) \propto \log(\frac{q_i}{1-q_i})$, i.e., the weight of agent i is proportional to $\log(\frac{q_i}{1-q_i})$.*

Proof. First by this weight allocation, we have that for each agent $i \in N$, the voting weight $w(i) \propto w\log(\frac{q_i}{1-q_i})$, where w is a constant.

Then for any winning coalition $N_1 \subseteq N$, we have that

$$\sum_{i \in N_1} w(i) > \sum_{i \in N \setminus N_1} w(i), \tag{2.6}$$

which can be rewritten as

$$\sum_{i \in N_1} w\log(\frac{q_i}{1-q_i}) > \sum_{i \in N \setminus N_1} w\log(\frac{q_i}{1-q_i}). \tag{2.7}$$

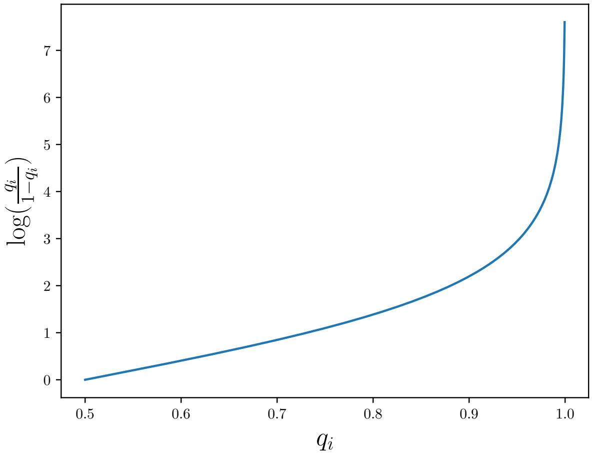


Figure 2.2: Plot of the optimal voting weight, i.e., $\log(\frac{q_i}{1-q_i})$, as a function of q_i.

Then we further obtain

$$\prod_{i \in N_1} \frac{q_i}{1-q_i} > \prod_{i \in N \setminus N_1} \frac{q_i}{1-q_i}, \tag{2.8}$$

and then from Equation 2.8, we have

$$\prod_{i \in N_1} q_i \prod_{i \in N \setminus N_1} (1-q_i) > \prod_{i \in N_1} (1-q_i) \prod_{i \in N \setminus N_1} q_i. \tag{2.9}$$

Equation 2.9 indicates that under the voting weight allocation rule, each winning coalition's accuracy is always higher than that of the coalition's complement set (which is therefore a losing coalition). Therefore we conclude that q_N is maximal. □

Intuitively, this weight allocation rule guarantees that for each winning coalition, the group accuracy is higher when the coalition chooses the true state other than when its complement chooses the true state. The optimal voting weight $\log(\frac{q_i}{1-q_i})$ is plotted in Figure 2.2 as a function of q_i.

Note that this weight allocation is not the unique one to achieve this maximal group accuracy. If any coalition satisfying Equation 2.6 is a winning coalition, the same maximal group accuracy can be obtained. The intuition is shown in Example 6.

Example 6 ([65])**.** *Consider five agents with accuracies* $(0.9, 0.9, 0.6, 0.6, 0.6)$*. Then by the weight allocation rule in Theorem 4, we obtain the normalized weight allocation*

$(0.392, 0.392, 0.072, 0.072, 0.072)$. *However, if the inequality system in Equation 2.10 is satisfied, the set of winning coalitions is consistent, i.e., corresponding to the same group accuracy.*

$$\begin{aligned}
w(1) + w(2) &> w(3) + w(4) + w(5) \\
w(1) + w(3) + w(4) &> w(2) + w(5) \\
w(1) + w(3) + w(5) &> w(2) + w(4) \\
w(1) + w(4) + w(5) &> w(2) + w(3) \\
w(2) + w(3) + w(4) &> w(1) + w(5) \\
w(2) + w(3) + w(5) &> w(1) + w(4) \\
w(2) + w(4) + w(5) &> w(1) + w(3)
\end{aligned} \tag{2.10}$$

For example, weight allocations $(0.47, 0.47, 0.02, 0.02, 0.02)$ *and* $(0.26, 0.26, 0.16, 0.16, 0.16)$ *can both achieve the maximal group accuracy.*

This extension of the Condorcet theorem is the starting point of our contribution in Part III, since liquid democracy can be viewed as a mechanism which re-assigns voting weights among the agents. That is, in Part III, we aim at (approximately) achieving the voting weight distribution proposed in Theorem 4, by liquid democracy, in order to achieve the maximal group accuracy.

Since in liquid democracy, the weight re-allocation process is realised via delegation, we re-define the group accuracy for liquid democracy by introducing an additional input, the delegation profile $\mathbf{d}$.

Definition 2 (Group accuracy under delegation)**.** *Given a group of agents N, an individual accuracy profile* $\mathbf{q}$ *and a delegation profile* $\mathbf{d}$*, the* group accuracy *under* $\mathbf{d}$ *is:*

$$q_{N,\mathbf{d}} = \sum_{C \in W} \prod_{i \in C} q_i \prod_{i \in N \setminus C} (1 - q_i), \tag{2.11}$$

where $W = \{C \subseteq N \mid \sum_{i \in C} w(i, \mathbf{d}) > \sum_{i \in N \setminus C} w(i, \mathbf{d})\}$ is the set of winning coalitions*. Ties are broken uniformly at random.*

We simply call "group accuracy under delegation" *group accuracy* in the rest of the dissertation.

In the following two sections of this chapter, we review the two main recent lines of research in liquid democracy upon which we build our contributions.

The first one focuses on agents' motivation to delegate and assumes that agents seek to delegate to experts on whom the delegators can rely to make better decisions. This research line takes a decentralized perspective. The second one focuses on the effects of liquid democracy on group accuracy. It assumes that agents decide their delegations through mechanisms, and studies the group accuracies achieved by various mechanisms.

In Section 2.4, we first introduce the game-theoretical model by [11], in which agents aim to delegate to more accurate agents, and the resulting equilibrium analysis. Thereafter, in Section 2.5, the algorithmic analysis on group accuracy optimization problem is introduced, and it provides more insight into understanding the challenge imposed on this optimization problem.

2.4 RATIONAL DELEGATIONS

Bloembergen *et al.* [11] apply game theory [59] to study agents' rational delegation strategy in liquid democracy. Other than in the truth-tracking model (Section 2.1), this model belongs to a more general class, which we call *type tracking*, where each agent benefits from one and only one of the two alternatives in the binary voting setting, and the preferred alternative of an agent is called her "type". In this model, each agent belongs to a type probabilistically, and is also initialized with an accuracy: agents can only imperfectly recognize their types. The truth-tracking model above is a special case of the type-tracking model where all agents have the same type.

The focus of [11] is the decision-making problem that the voters face: choosing for direct voting, which determines a cost in terms of *effort* invested to acquire information of the issue being voted for; or choosing to delegate to another voter so as to avoid the cost.

2.4.1 From Truth Tracking To Type Tracking

In the model by Bloembergen *et al.* [11], agents seek to delegate to gurus, who tend to make better choices for them. Instead of truth tracking in binary votings, each agent $i \in N$ is initialized with a *type* $\tau(i) \in \{0, 1\}$, such that i benefits from the candidate consistent with her type. Agents' types are *probabilistic*, i.e., $\tau(i)$ is a random variable drawn from a distribution $\mathbb{P}$. Then for any pair of agents $i, j \in N$, let $p_{i,j}$ denote the likelihood that they are of the same type, i.e., $p_{i,j} = \mathbb{P}(\tau(i) = \tau(j)) = \mathbb{P}(\tau(i) = 1)\mathbb{P}(\tau(j) = 1) + (1 - \mathbb{P}(\tau(i) = 1))(1 - \mathbb{P}(\tau(j) = 1))$. Note that the truth-tracking model is a special case, where all agents are of type θ with probability 1.

Then an agent's accuracy is expressed as the probability that the agent votes according to her type. That is $q_i = \Pr(v_i = x \mid \tau(i) = x)$, where $x \in \{0, 1\}$.

Each agent i can then choose between: (i) voting on her own behalf with accuracy q_i and pay effort $e_i \in [0, 0.5)$, or (ii) delegating to a neighbor in the underlying network, and inherit the accuracy of the neighbor's guru. Based on a delegation profile $\mathbf{d}$, we can compute the individual accuracy of agent i as:

$$q_i^*(\mathbf{d}) = q_{d_i^*} p_{i,d_i^*} + (1 - q_{d_i^*})(1 - p_{i,d_i^*}), \tag{2.12}$$

if i is not caught in a delegation cycle, otherwise 0.5, i.e., as accurate as a random decision. Intuitively, if i delegates to guru d_i^*, the probability that her vote is used according to her type equals the probability that her guru votes correctly when they are of the same type, plus the probability that the guru makes a wrong decision when they have different types.

2.4.1.1 *Delegation Games*

Based on these concepts, we define a *delegation game* as follows.

Definition 3. *A delegation game is a tuple $G = \langle N, \mathbb{P}, R, \Sigma_i, u_i \rangle$, with $i \in N$, where N is the set of agents, $\mathbb{P}$ is the probability distribution on the types $\{0, 1\}$, R is the undirected graph representing the underlying network, $\Sigma_i \in N$ is the strategy space of agent i (i.e., her neighborhood $R(i)$ in the underlying network), and the utility function u_i of i is:*

$$u_i(\mathbf{d}) = \begin{cases} q_i^*(\mathbf{d}) & \text{if } d_i \neq i \\ q_i - e_i & \text{if } d_i = i \end{cases}. \tag{2.13}$$

Observe that, in the above definition, i's utility equals the accuracy inherited from her guru if i is a delegator, otherwise it equals her own accuracy minus the effort, e_i, spent in order to vote with accuracy q_i. Note that $q_i - e_i$ is assumed to be in range $(0.5, 1]$, otherwise the agent would prefer to choose randomly without effort. Additionally, agents caught in a delegation cycle are assumed to obtain a utility of 0.5, i.e., they make random decisions, due to their loss of representation. In such cases, agents fail to express their opinions (see also [20]).

Among the general delegation games, some special classes can be specified. We call those delegation games with *deterministic types* where all agents have a type with probability 1. Moreover, when all agents have the same type, it is called a delegation game with *homogeneous agents*, i.e., the same setting as the truth-tracking model of Section 2.1. When, for all agents, the effort to directly use their vote is 0, we call it a delegation game with *effortless voting*.

Bloembergen *et al.* [11] studied the existence of (pure strategy) *Nash equilibria* (NE), which, in some subclasses of the delegation games, may be achieved by *best response* dynamics.

Definition 4 (Nash Equilibria)**.** *Let G be a delegation game. A delegation profile $\mathbf{d}$ is a Nash equilibrium of G if for any agent $i \in N$, no profile $\mathbf{d}'$ ($\mathbf{d}' \neq \mathbf{d}$) exists such that for all $j \in N \setminus \{i\}$, $\mathbf{d}'(j)$=$\mathbf{d}(j)$, and $u_i(\mathbf{d}') > u_i(\mathbf{d})$.*

Intuitively, in a Nash equilibrium, no agent has incentive to unilaterally change her delegation strategy.

Definition 5 (Best Response)**.** *Given a delegation game G and a delegation profile $\mathbf{d}$, for any agent $i \in N$, delegation strategy $d_i' \in \Sigma_i$ is called a best response of agent i to* *profile* $\mathbf{d}$ *if* $d'_i \in \arg\max_{d''_i \in \Sigma_i} u_i((\mathbf{d}_{-i}, d''_i))$, *where* $(\mathbf{d}_{-i}, d''_i)$ *is the profile in which all agents take the delegation strategy in* $\mathbf{d}$, *except that* i *takes* d''_i.

In other words, given a delegation profile, an agent's best response is a delegation strategy which maximizes the agent's utility. We use the following example to intuitively illustrate the above concepts.

Example 7 (Homogeneous two-agent delegation game [11]). *Consider two agents* $N = \{1, 2\}$ *in a complete underlying network* $R = \langle N, E = N^2 \rangle$, *with deterministic homogeneous type, e.g.,* $\Pr(\tau(1) = \tau(2) = 1) = 1$. *Then the delegation game can be described by the payoff matrix in Table 2.1.*

	vote	delegate (to 1)
vote	$q_1 - e_1, q_2 - e_2$	$q_1 - e_1, q_1$
delegate (to 2)	$q_2, q_2 - e_2$	$0.5, 0.5$

Table 2.1: The delegation game with homogeneous agents in Example 7

A situation can arise such that for some $i, j \in N$ $(i \neq j)$, $q_j > q_i - e_i$. *That is, each of the two agents prefers to delegate. However, by the above assumption that both* $q_1 - e_1$ *and* $q_2 - e_2$ *are always weakly larger than* 0.5, *agents need coordination to avoid acting with the same strategy, i.e., both vote or both delegate (to form a delegation cycle). Notice that there are two NE in this example, i.e.,* $(d_1 = 2, d_2 = 2)$ *and* $(d_1 = 1, d_2 = 1)$, *since no single agent can improve by unilaterally changing her own strategy.*

Whether Nash equilibria exist in delegation games is an open question. However, Bloembergen *et al.* [11] show that Nash equilibria exist in two special classes of delegation games: delegation games with deterministic types and delegation games with effortless voting.

DETERMINISTIC TYPE DELEGATION GAMES This class restricts agents' type to be deterministic, that is, each agent i has type 0 or 1 with probability of 1 $(\mathbb{P}(\tau(i)) = 1)$. Then the existence of a pure NE can always be guaranteed in this class.

Theorem 5 ([11]). *Delegation games with deterministic type agents always have a pure strategy NE.*

Intuitively, to converge to a NE, for each type of agents, we only consider a sub-network of R, which consists of all agents of that type and connected by edges in R. Then in each connected component, all agents delegate to the one with the highest accuracy, and this delegation structure forms a NE, since each delegator inherits the maximal accuracy so that she has no incentive to deviate while, also, the gurus cannot deviate, otherwise a delegation cycle would be formed. Note that, in this structure, the optimal strategy of the highest-accuracy agent in each connected component is to be a guru even though voting might cost her high effort, otherwise a delegation cycle would be formed if she delegates.

This result immediately establishes the existence of NE in the truth-tracking setting of Section 2.1, where all agents are of the same deterministic type. This is the class of delegation games we will focus on in this dissertation.

EFFORTLESS VOTING This situation corresponds to $e_i = 0$ for all $i \in N$. Here the existence of a pure strategy NE can also be guaranteed.

Theorem 6. *Delegation games with effortless voting always have a pure strategy NE.*

Different from the above, the proof is by showing convergence of a process of best response: agents iteratively choose their best response to the delegation profile at that moment. It is shown that each agent's best response does not do harm to the other agents' utility, therefore the process converges to a stable state corresponding to a NE [11].

PRICE OF ANARCHY Later in Chapter 6, we will also be investigating the quality of NE by using the *price of anarchy* [51]. To provide the general definition, we will be using the *social welfare* of the delegation games. Let $\mathbb{D}^p$ be the set of all pure profiles. $\hat{u} : \mathbb{D}^p \to \mathbb{R}$ is the social welfare which takes a pure profile and outputs a real number. An example of social welfare is the utilitarian welfare, which equals the sum of each agent's individual utility: $\sum_{i \in N} u_i(\mathbf{d})$. The price of anarchy based on pure strategy is defined as follows.

Definition 6 (Pure Price of anarchy)**.** *Given a delegation game G with utility function $\hat{u}$, the price of anarchy of G is as follows:*

$$\mathsf{PoA}^{pure} = \frac{\max_{\mathbf{d} \in \mathbb{D}^p} \hat{u}(\mathbf{d})}{\min_{\mathbf{d} \in \mathcal{E}^p(G)} \hat{u}(\mathbf{d})}, \tag{2.14}$$

where $\mathcal{E}^p(G)$ is the set of pure strategy Nash equilibria of delegation game G.

Intuitively, the PoA^{pure} of a delegation game reflects the game's efficiency. That is, it measures ratio of social welfare lost in order to achieve the worst Nash equilibrium for selfish agents.

In Chapter 6, we will be working with forms of mixed PoA, where agents are allowed to delegate weightedly.

2.5 LIQUID DEMOCRACY V.S. DIRECT DEMOCRACY

Instead of pursuing to inherit high individual accuracy, in the second research line relevant for us, agents are motivated to delegate in order to better reveal the true state under the setting of truth-tracking binary votings, i.e., to achieve high group accuracy. By the Condorcet theorem and its extensions, it is known that direct voting, especially when most agents are well-informed (e.g., with individual accuracies higher than 0.5), performs well in terms of truth tracking. Literature in liquid democracy has asked the natural question: can liquid democracy perform even better than direct voting?

2.5.1 Delegation Mechanisms

Kahng *et al.* [46] provide a relatively pessimistic answer. They defined two classes of delegation mechanisms (that is, processes constructing delegation profiles): local and non-local, where in a local mechanism, agents' delegation strategies only depend on their neighbors, while global information is needed to determine delegations in a non-local mechanism. Observe that these mechanisms are not game-theoretic. Kahng *et al.* [46] mainly study whether two properties are satisfied: the *positive gain* (*PG*), i.e., the delegation mechanism can outperform direct voting in terms of group accuracy (Definition 1), and *do not harm* (*DNH*), i.e., the loss of group accuracy on the delegation mechanism from the direct voting is asymptotically bounded to zero. They showed that no local mechanism is *PG* and *DNH*, but non-local mechanisms that satisfy *PG* and *DNH* do exist.

DELEGATION MECHANISMS In this model, given a set of agents N, a *delegation mechanism* $\mathbf{M}$ is a function which takes the underlying network which is represented as a directed graph $R = \langle N, E \rangle$ and an accuracy profile $\mathbf{q}$ and outputs a probability distribution on a set of delegation profiles. Mechanism $\mathbf{M}$ is said to *deterministically* output a delegation profile if the output distribution is degenerated on this delegation profile. Then we define the accuracy of the delegation mechanism as follows.

Definition 7 (Mechanism Accuracy)**.** *Given the underlying network R and an accuracy profile $\mathbf{q}$, the accuracy of delegation mechanism $\mathbf{M}$, $q_{\mathbf{M}}(R, \mathbf{q})$, is the probability that the decision made by the following 4 steps is correct:*

1. *Apply $\mathbf{M}$ to R and $\mathbf{q}$.*
2. *Sample a delegation profile from the above output distribution.*
3. *In the profile, each guru votes with her accrued weight computed by Equation 2.1 and individual accuracy.*
4. *A decision is made by weighted majority rule (Equation 2.2) with a random tie breaker.*

Notice that the group accuracy under delegation (Definition 2) is a special case of the mechanism accuracy, where in $q_{N,\mathbf{d}}$, the profile can be seen as a mechanism which outputs a profile $\mathbf{d}$ deterministically.

Then any mechanism is either *local* or *non-local*. Given an underlying network R and an accuracy profile $\mathbf{q}$, agent i *approves* her neighbor $j \in R'(i)$ if $q_j > q_i + \alpha$ for some parameter $\alpha \in [0,1)$. Intuitively agent i approves agent j if j's accuracy exceeds i's to a strictly greater extent than α. Let $A_{R,\mathbf{q}}(i)$ denote all i's approval neighbors, i.e., $A_{R,\mathbf{q}}(i) = \{j \in R'(i) \mid i \text{ approves } j\}$. Then by a local mechanism, i's delegation strategy only depends on $A_{R,\mathbf{q}}(i)$, for example, a local mechanism is that each agent uniformly at random chooses an approval neighbor to delegate.

GAIN Let Di denote the direct voting mechanism, i.e., $Di(R, \mathbf{q})$ is the trivial delegation profile, under which the decision is made by simple majority in the one-person-one-vote setting. The *gain* of mechanism $\mathbf{M}$ is then defined as:

$$gain(\mathbf{M}, R, \mathbf{q}) = q_{\mathbf{M}}(R, \mathbf{q}) - q_{Di}(R, \mathbf{q}). \tag{2.15}$$

Intuitively, the gain of mechanism $\mathbf{M}$ on network R measures the extent to which mechanism $\mathbf{M}$ outperforms Di in terms of mechanism accuracy.

Then the desired axioms are formally defined as:

- A mechanism $\mathbf{M}$ satisfies *positive gain* if there exist $\gamma > 0, n_0 \in \mathbb{N}$, such that for all $n \geq n_0$, there exists a network R_n (with $\mathbf{q}$) on n vertices such that $gain(\mathbf{M}, R_n, \mathbf{q}) \geq \gamma$.
- A mechanism $\mathbf{M}$ satisfies *do not harm* if for all $\epsilon > 0$, there exists $n_1 \in \mathbb{N}$ such that for all networks R_n (with $\mathbf{q}$) on $n \geq n_1$ vertices, $gain(\mathbf{M}, R_n, \mathbf{q}) \geq -\epsilon$.

Kahng *et al.* [46] verify that liquid democracy can easily lead to undesirable situations from the viewpoint of truth tracking, as illustrated in Example 8.

Example 8 ([46])**.** *Consider a network $R = \langle N, E\rangle$, such that $E = \{(i,1) \mid i \in N \setminus \{1\}\}$, individual accuracy $q_1 = \frac{4}{5}$ and $q_i = \frac{2}{3}$ for all $i \in N \setminus \{1\}$, and $\alpha = \frac{1}{10}$. That is, the network is a star, where each agent in $N \setminus \{1\}$ has and only has one neighbor 1. Then a local mechanism $\mathbf{M}$, by which any agent delegates to the highest-accuracy approval neighbor, outputs delegation profile $\mathbf{d}$, in which $d_1 = 1$ and $d_i = 1$ for all $i \in N \setminus \{1\}$. Hence the group accuracy degenerates to the individual accuracy of agent 1, which is $4/5$. However, by Di, as $|N| \to \infty$, the accuracy $q_{Di(R,\mathbf{q})} \to 1$ due to the Condorcet theorem, and the $gain(\mathbf{M}, R, \mathbf{q})$ approaches $-\frac{1}{5}$.*

They further prove that no "smart" local mechanism exists.

Theorem 7 ([46])**.** *For any $\alpha_0 \in [0,1)$, there is no local mechanism that satisfies the PG and DNH properties.*

Theorem 7 indicates that, to achieve better performance on truth tracking, a centralized mechanism is needed to coordinate agents' delegation strategies. Hence Kahng *et al.* [46] propose a non-local mechanism called *GreedyCap* (Algorithm 1), in which a cap on accrued delegation weight is imposed on each agent.

Algorithm 1 GreedyCap

INPUT: underlying network: R, accuracy profile $\mathbf{q}$, cap $C : \mathbb{N} \to \mathbb{N}$

$N' \leftarrow N$
while $N' \neq \emptyset$ **do**
  let $i \in argmax_{j \in N'} |A^{-1}_{R,\mathbf{q}}(j) \cap N'|$
  $J \leftarrow A_{R,\mathbf{q}}(i) \cap N'$
  if $|J| \leq C(n) - 1$ **then**
    $J' \leftarrow J$
  else
    let $J' \subseteq J$ such that $|J'| = C(n) - 1$
  end if
  vertices in J' delegate to i
  $N' \leftarrow N' \setminus (\{i\} \cup \{J'\})$
end while

In words, this algorithm takes as input the underlying network, the accuracy profile and the cap $C : \mathbb{N} \to \mathbb{N}$, which is a function taking the number of agents $n = |N|$ and outputting the cap. In each iteration, the algorithm selects an agent with the most approvals as a guru, and lets at most $C(n)$ neighbors delegate to the agent. Those delegators do not further change their delegation strategies, therefore the output delegation profile only contains one-hop delegation chains.

It can be observed that GreedyCap is PG. However, it violates DNH as shown in Example 9.

Example 9 ([46]). *Assume that $\alpha < 1/3$. For any odd $n = 2k + 1$, consider the underlying network $R = \langle N, E = \{(1, 2)\}\rangle$, that is, there is only one edge, from agent 1 to 2. The individual accuracies are: $q_1 = 1/3$, $q_2 = 2/3$, there are k agents with accuracy 1, and the other $k - 1$ agents with accuracy 0. Even with cap $C(n) = 2$, by GreedyCap, agent 1 would delegate to agent 2. Then the true state will be selected if and only if agent 2 makes the correct choice, i.e., with probability $2/3$. However, by direct voting, it is sufficient that at least one of agents 1 and 2 vote for the true state, i.e., with probability $7/9$. Hence the loss of GreedyCap is $1/9$, which indicates that it violates DNH.*

Kahng *et al.* [46] show that if all agents' accuracies are bounded away from 0 or 1, i.e., they do not always make a correct/wrong decision, GreedyCap can guarantee both properties.

Theorem 8 (Theorem 2, [46]). *Assume that there exists $\beta \in (0, 1/2)$, such that $q_i \in [\beta, 1 - \beta]$ for all $i \in N$. Then for any $\alpha \in (0, 1 - 2\beta)$, GreedyCap with cap $C : \mathbb{N} \to \mathbb{N}$, such that $C(n) \in \omega(1)$ (loose upper bound of $C(n)$) and $C(n) \in o(\sqrt{\log n})$ (loose lower bound of $C(n)$) satisfies the PG and DNH properties.*

Intuitively, the GreedyCap algorithm limits the accrual of delegations. In doing so, it limits the effects of delegations on voters' independence.

2.5.2 Optimal Delegation Profiles Are Hard To Find

Caragiannis & Micha [16] provide further reasons for objecting to local mechanisms.

They first fix a local mechanism, called α-delegation in their work, to a dichotomous scheme: a misinformed agent i, i.e., with individual accuracy less than 0.5, would approve a neighbor j if $q_j < q_i - \alpha$ since she, in practice, might consider a high-accuracy agent as misinformed. Observe that this goes against the assumption in the Condorcet theorem (Theorem 1). Then Caragiannis & Micha [16] show that a local mechanism $\mathbf{M}$ utilizing such α-delegation is considerably worse than two extreme mechanisms: *Di* (direct voting) and *FD* (Full Delegation, i.e., all agents delegate to one guru).

Theorem 9 ([16]). *Let* $\mathbf{M}$ *be a local delegation mechanism,* $\alpha \geq 0$ *and* $\delta > 0$. *There exists a pair* $(R, \mathbf{q})$ *such that* $\max\{q_{Di}(R, \mathbf{q}), q_{FD}(R, \mathbf{q})\} - q_{\mathbf{M}}(R, \mathbf{q}) \geq \frac{1}{2} - \alpha - \delta$.

While local delegation mechanisms appear undesirable, centralized coordination cannot easily direct delegations to achieve high performance in terms of group accuracy either. The following example shows that, counter to intuitions, delegating to low-accuracy agents may even result in better group accuracy.

Example 10 ([16]). *Consider a case with 5 agents, with individual accuracies and underlying network shown in Fig. 2.3. Note that this underlying network is represented*

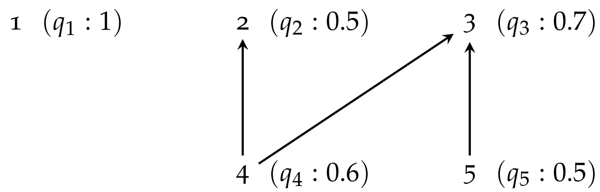


Figure 2.3: The underlying network and individual accuracy.

as a directed graph, i.e., the tail agent can delegate to the head agent but not vice versa. We list 6 mechanisms that output all possible delegation profiles, as well as the group accuracy of the output delegation profiles.

Di: Since agent 1 *always chooses the correct alternative, the true state is selected with the probability that at least 2 among* $\{2, 3, 4, 5\}$ *make the correct choice. This yields the group accuracy of* $q_{Di}(R, \mathbf{q}) = 0.795$.

M$_1$: *Agent* 4 *delegates to agent* 2. *Then the true state wins when agent* 2 *votes for it or agent* 2 *votes for the wrong alternative and both agents* 3 *and* 5 *vote for the truth, by which we obtain* $q_{\mathbf{M}_1}(R, \mathbf{q}) = 0.675$.

M$_2$: *Agent* 4 *delegates to agent* 3. *Now the true state is selected when agent* 3 *votes for it or agent* 3 *votes for the wrong alternative and both agents* 2 *and* 5 *vote for the true state. Then the group accuracy is* $q_{\mathbf{M}_2}(R, \mathbf{q}) = 0.775$.

M$_3$: *Agent* 5 *delegates to agent* 3. *Similarly, when agent* 3 *votes for the true state or agent* 3 *votes for the wrong alternative and both agents* 2 *and* 4 *vote for the true state, the set of agents can successfully select the true state, and this gives* $q_{\mathbf{M}_3}(R, \mathbf{q}) = 0.79$.

M$_4$: *Agents* 4 *and* 5 *delegate to agent* 3. *Now the true state can be selected if and only if agent* 3 *votes for it. Therefore we obtain* $q_{\mathbf{M}_4}(R, \mathbf{q}) = q_3 = 0.7$.

M$_5$: *Agent* 4 *delegates to agent* 2 *and agent* 5 *delegates to agent* 3. *By this delegation profile, when at least one of agents* 2 *and* 3 *vote for the true state, it will win. Hence* $q_{\mathbf{M}_5}(R, \mathbf{q}) = 0.85$.

This example conveys the intuition that liquid democracy can achieve better group accuracy by introducing *appropriate* correlation, sometimes even by delegating to non-optimal neighbors. However this intuition also reflects the high complexity of the optimization problem: given the underlying network and individual accuracy, can we coordinate agents' delegations to maximize the group accuracy? Caragiannis & Micha [16] show that it is NP-hard in terms of the size of the input network to solve this question.

2.6 OUTLINE OF CONTRIBUTION

Based on these preliminaries, the dissertation proceeds in two parts. In Part II, we first propose a novel power index to measure the influence of agents, including delegators, on the final voting result. Then we incorporate this power index into the game-theoretical model of [11]. In doing so, we study delegation games where agents do not only consider to delegate to accurate agents, but also want to retain power in the system.

In Part III, inspired by Theorem 4, we seek methods which allow agents to partition their voting weight and redistribute it continuously among the network, in order to achieve better group accuracy and counter the negative results in [46] and [16]. We also consider these weighted delegation methods in the more decentralized game-theoretical model of [11], and provide insights into the influence brought by weighted delegation.

3 POWER IN LIQUID DEMOCRACY

This chapter develops a theory of power for liquid democracy systems. Understanding voting power is crucial in liquid democracy since, due to the flexibility of liquid democracy, a small set of agents might accrue too much voting weight, i.e., become super-voters. A straightforward application of the Banzhaf index (see Section 2.2) allows us to quantify the voting power of gurus. However, note also the fact that agents can terminate the delegation of their votes at any time due to the *instant call component*, which is one of the four fundamental components in the definition of liquid democracy (recall Section 1.1 and [12]). Delegators can also be pivotal: they might change the voting result by altering their delegations.

CHAPTER CONTRIBUTION In this chapter, we aim at capturing the *instant call component* in the measure of voting power for liquid democracy. We define a power index able to measure the influence of both gurus and delegators on voting. We then axiomatically characterize the power index we define and further study several intuitive properties of the index. The part presents and extends material from [68].

3.1 A POWER INDEX FOR LIQUID DEMOCRACY

To measure agents' voting power in liquid democracy, we generalize the well-known Banzhaf power index (Equation 2.3) in the context of liquid democracy. Specifically, given a delegation profile, we study the influence imposed by agents, including both gurus and delegators, on the voting phase. Our method could be fairly easily adapted to obtain similar generalization of other power indices such as Shapley-Shubik power index (recall Equation 2.4).

3.1.1 Liquid Democracy Election (LDE)

In this chapter we will be working with the general setting introduced in Section 2.1: each agent is initialized with a (possibly different) non-negative weight $w(i) \in \mathbb{R}_{>0}$ for all $i \in N$; and the voting result is decided by the *quota rule* defined as follows. Let $w = \sum_{i \in N} w(i)$, then the quota rule $\mathcal{Q}$ is formally defined as: given a quota $\beta \in (\frac{w}{2}, w]$ and a ballot profile $\mathbf{v}$, the issue voted for is accepted if and only if the total weight of agents who vote *accept* (e.g., alternative 1) matches or exceeds the quota (see Chalkiadakis *et al.* [19]), i.e.,

$$\mathcal{Q}(\mathbf{v}) = \begin{cases} 1, & \text{if } \sum_{i \in N} w(i) v_i \geq \beta \\ 0, & \text{otherwise} \end{cases}. \tag{3.1}$$

Notice that the quota β is assumed to be strictly larger than half of the total weight. Otherwise the quota rule would not be resolute: the case is possible that both alternatives meet the quota. Notice also that the rule uses a deterministic tie-breaking and is biased towards one of the two alternatives.

It is worth mentioning that the above quota rule uses the absolute quota rule, i.e., β is an absolute threshold between $\sum_{i \in N} w(i)/2$ and $\sum_{i \in N} w(i)$. However, another form of quota is the relative quota, i.e., the issue is accepted if the ratio between the weight of agents who vote for 1 and the "entire weight" exceeds a threshold in interval $[0.5, 1]$. Using a relative quota is a potential extension of our work, since there are two alternatives of the "entire weight" in LD: (1) the weight of all agents $\sum_{i \in N} w(i)$, or (2) the weight of all gurus $\sum_{i \in De(N, \mathbf{d})} w(i)$ given a profile $\mathbf{d}$. Notice that the latter alternative is dynamic with respect to $\mathbf{d}$ because of abstention, where the corresponding agents' guru is 0 (see following description).

Note that in Section 4.2.2, we also study the LDEs with a quota less or equal to $\frac{w}{2}$, which lays out of the range in the above definition.

Example 11. *Some specific quota choices yield frequently used voting rules. For instance, when* $\beta = \frac{\sum_{i \in N} w(i)}{2} + \epsilon$ *(where* ϵ *is a sufficiently small positive real number), Equation 3.1 defines the deterministic version of the weighted majority rule shown in Equation 2.2. That is, alternative* 1 *wins if more than half of the total voting weight supports it.*

When $\beta = \sum_{i \in N} w(i)$*, Equation 3.1 defines the rule, where the issue is accepted if and only if all agents accept it.*

Definition 8 (Liquid Democracy Election)**.** *We call a* liquid democracy election *(LDE) the tuple* $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$*, where* N *is the set of agents with weight profile* $\mathbf{w} = (w(1), \ldots, w(n))$*,* $\mathbf{d}$ *is a delegation profile, and* β *is a quota.*

Note that in a given delegation profile $\mathbf{d}$, any guru $i \in Gu(\mathbf{d})$ now accrues voting weight $w(i, \mathbf{d}) = \sum_{j \in De(i, \mathbf{d})} w(j)$. That is, guru i accrues the voting weight of all agents who delegate to i through delegation chains as introduced in Equation 2.1. Recall also that $w(j)$ is the voting weight of agent j. Then, in LDEs, we modify the quota rule of Equation 3.1 by replacing the individual weight $w(i)$ by the accrued weight through delegations $w(i, \mathbf{d})$ for all gurus $i \in Gu(N)$. Let then $\mathbb{V}$ denote the set of all LDEs. Clearly, LDEs with trivial profiles are instances of standard weighted voting (Equation 2.2).

In this chapter, we also admit the possibility for an agent to abstain by delegating to a nul agent 0. When a delegation chain ends up with the 0 agent, e.g., $i_1 \rightarrow i_2 \rightarrow \cdots \rightarrow 0$, all agents on the chain abstain, in other words, their weight will not be counted in the voting phase. This feature will be of technical use for the characterization of the power index we are going to introduce.

We need to introduce some further terminology. Given the chain $i_1 \rightarrow i_2 \rightarrow i_3 \rightarrow \cdots \rightarrow i_{k-1} \rightarrow i_k$, if $\{i_1, i_3, \dots, i_k\} \subseteq C \subseteq N$ but $i_2 \notin C$, then i_1 is not able to delegate to i_k within C as she has no access to intermediary i_3 in such subset. For $C \subseteq N$ we write $\mathbf{d}^*_C(i_3) = i_k$ to denote that i_k is the guru of i_3 and the chain from i_3 to i_k contains only elements of C, i.e., for $i_3, i_k \in C$, $\mathbf{d}^*_C(i_3) = i_k$ if $\mathbf{d}^*(i_3) = i_k$ and $\delta_{\mathbf{d}}(i_3, i_k) \subseteq C$. Then we write

$$De^*(C, \mathbf{d}) = \{j \in N \mid \exists k \in C, \mathbf{d}^*_C(j) = k\} \tag{3.2}$$

for the set of agents that directly or indirectly delegate to some agent in C through intermediaries contained in C. Intuitively, this captures the support accrued by gurus in C via agents in C. Notice that $De(N, \mathbf{d}) = De^*(N, \mathbf{d})$.

3.1.2 Delegative Banzhaf Index

Once delegations are settled, liquid democracy results in weighted voting where only the gurus vote with weight equal to the sum of weights they accrued from direct or indirect delegations. From a voting perspective, gurus are therefore the only agents who retain voting power after the delegation phase. However, this neglects the *instant call component* of liquid democracy and the power that delegators actually have within liquid democracy by being able to control a large number of votes. Let us give a simple example. A guru i obtaining m direct delegations is intuitively more 'powerful' than a guru obtaining m delegations via an intermediary j, who is in turn recipient of $m - 1$ direct delegations. Most of i's power then depends on j (see also Example 13 below).

In this section we generalize the standard Banzhaf index (Equation 2.3) to the delegable proxy voting setting. The Banzhaf index has already been used to study the power of gurus in liquid democracy by Kling *et al.* [49].

There is one obvious way in which an LDE V induces a simple game (see the definition of simple games in Section 2.2): it is the simple game capturing the weighted voting occurring among gurus once delegations have been fixed, i.e., $\mathcal{G}_V = \langle N, f_V \rangle$ where, for any $C \subseteq N$:

$$f_V(C) = 1 \text{ iff } \sum_{i \in Gu(C, \mathbf{d})} w(i, \mathbf{d}) \geq \beta. \tag{3.3}$$

That is, a coalition wins whenever all gurus in it together accrue enough weight to meet the quota. In such a game, only gurus may have positive power: $i \in Gu(\mathbf{d})$

$$1 \longleftarrow 2 \qquad 3 \qquad 4 \longleftarrow 5 \longleftarrow 6 \longleftarrow 7$$

Figure 3.1: The delegation graph in Example 12

if $\mathsf{B}_i(\mathcal{G}_V) > 0$, because $\mathcal{G}_V$ is silent about the influence that delegators have in determining the winning coalitions.

The influence of delegators can be captured by a different simple game defined as follows.

Definition 9 (Delegative Simple Games). *A* delegative simple game *induced from LDE $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ is a tuple $\mathcal{G}'_V = \langle N, f'_V \rangle$ where N is the set of agents and f'_V is the characteristic function, such that for any $C \subseteq N$:*

$$f'_V(C) = 1 \textit{ iff } \sum_{i \in De^*(C, \mathbf{d})} w(i) \geq \beta. \tag{3.4}$$

That is, a coalition C is winning whenever the sum of weights accrued by the gurus in C *from agents in* C (recall Equation 3.2), meets the quota. According to this way of constructing the simple game, an agent's weight is accrued in a coalition C if the agent, her guru, and all intermediaries between them are contained in C. We refer to $\mathcal{G}'_V$ as the *delegative simple game* of LDE V. Clearly, if $\mathbf{d}$ is trivial, all agents are gurus and therefore $\mathcal{G}_V = \mathcal{G}_{V'}$.

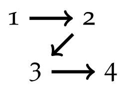
Example 12. *Consider an LDE V consisting of 7 agents, each of whom is initialized with voting weight $w_i = 1$, the quota is $\beta = 4$, and the delegation profile $\mathbf{d}$ is denoted in Figure 3.1. We consider coalition $C = \{3, 4, 6, 7\}$. When computing the coalition's weight under the simple game definition introduced in Section 2.2, i.e., $\mathcal{G}_V$, the weight of coalition C is the sum of the accrued weights by gurus 3 and 4 in delegation profile $\mathbf{d}$, according to Equation 2.1. Then the weight is 5, and C is a winning coalition in simple game $\mathcal{G}_V$.*

However, when we consider the delegative simple game $\mathcal{G}'_V$, agents 6 and 7 do not delegate to guru 4 since the delegation chain is broken (agent 5 is not in C). Therefore in $\mathcal{G}'_V$, guru 4 only accrues weight of 1 in coalition C. Hence C's weight is 2 in $\mathcal{G}'_V$ and it is a losing coalition.

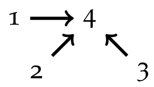
Definition 10 (Delegative Banzhaf Index). *Given an LDE $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$, we define the delegative Banzhaf index (*DB*) of an agent i in LDE V simply as the Banzhaf index of i in the delegative simple game of V:*

$$\mathsf{DB}_i(V) = \mathsf{B}_i(\mathcal{G}'_V). \tag{3.5}$$

Observe that in LDEs V where the delegation profile is trivial, and therefore games $\mathcal{G}(V)$ and $\mathcal{G}'(V)$ coincide, the Banzhaf index and the delegative Banzhaf index coincide.



(a) $\mathbf{d}_1$



(b) $\mathbf{d}_2$

Figure 3.2: The delegation profiles in Example 13

Example 13. *Consider two LDEs, $V_1 = \langle N, \mathbf{w}, \mathbf{d}_1, \beta \rangle$ and $V_2 = \langle N, \mathbf{w}, \mathbf{d}_2, \beta \rangle$, where $N = \{1, 2, 3, 4\}$, $w(i) = 1$ for all $i \in N$, $\beta = 3$, and $\mathbf{d}_1$ and $\mathbf{d}_2$ are represented in Figure 3.2a and Figure 3.2b, respectively. As follows, we list the computation of agents' DBs in both LDEs.*

- V_1:
  - *Agent 1: Agent 1 cannot swing for any coalition, since the only coalition counting her weight is $\{1, 2, 3, 4\}$ but $\{2, 3, 4\}$ is already a winning coalition. Therefore, we have $\mathsf{DB}_1(V_1) = 0$.*
  - *Agent 2: Agent 2 swings for coalitions $\{2, 3, 4\}$ and $\{1, 2, 3, 4\}$, e.g., for coalition $C = \{2, 3, 4\}$, $f'_{V_1}(C) = 1$ due to $\sum_{i \in C} w(i) = 3 \geq \beta$, but $f'_{V_1}(C \setminus \{2\}) = 0$. Therefore, we have $\mathsf{DB}_2(V_1) = 2/2^{4-1} = 1/4$.*
  - *Agent 3: Agent 3 swings for coalitions $\{2, 3, 4\}$ and $\{1, 2, 3, 4\}$, which results in that $\mathsf{DB}_3(V_1) = 1/4$.*
  - *Agent 4: Agent 4 also swings for the same coalitions as above, i.e., coalitions $\{2, 3, 4\}$ and $\{1, 2, 3, 4\}$. Hence, we have $\mathsf{DB}_4(V_1) = 1/4$.*
- V_2:
  - *Agent 1: Agent 1 is a direct delegator who delegates to guru 4. Observe that agent 1 swings for coalitions $\{1, 2, 4\}$ and $\{1, 3, 4\}$. Therefore, we have $\mathsf{DB}_1(V_2) = 2/2^{4-1} = 1/4$.*
  - *Agent 2 and Agent 3: Notice that, the same as agent 1, both agents 2 and 3 are also direct delegators who delegate to guru 4. Therefore, agents 1, 2 and 3 are in the "symmetric" positions in $\mathbf{d}_2$, and they have the same power, i.e., $\mathsf{DB}_2(V_2) = \mathsf{DB}_3(V_2) = \mathsf{DB}_1(V_2) = 1/4$.*
  - *Agent 4: Agent 4 is a swing agent for coalitions $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$ and $\{1, 2, 3, 4\}$, and hence we have $\mathsf{DB}_4(V_2) = 4/2^{4-1} = 1/2$.*

Observe that the delegative Banzhaf index depends on the structure of the delegation profile. For instance, in both V_1 and V_2, agent 1 is a delegator with no incoming delegations, but $\mathsf{DB}_1(V_1) = 0$ while $\mathsf{DB}_1(V_2) = 1/4$. In V_1, agent 1 is "far" from the guru and her delegation does not matter for meeting the quota. In V_2, however, agent 1 delegates directly to the guru. Similarly, in both LDEs agent 4 collects 4 votes. However, $\mathsf{DB}_4(V_1) = 1/4$ but $\mathsf{DB}_4(V_2) = 1/2$ since in V_1, the delegation chain pointing to 4 *is long, so that agent 4 depends on 3 for 3/4 of her weight. In V_1, even though both agents 2 and 3 delegate to agent 4, that is, agents 2 and 3 rely on agent 4 to make the decision, the three agents have the same power (i.e.,* $\mathsf{DB}_2(V_1) = \mathsf{DB}_3(V_1) = \mathsf{DB}_4(V_1)$*) to influence the final voting result.*

3.2 AN AXIOMATIC CHARACTERIZATION OF THE DELEGATIVE BANZHAF INDEX

There are of course many ways in which a power index could be defined on LDEs. To underpin Equation 3.5 we present a characterization of the delegative Banzhaf index. We want to axiomatically identify DB among all power index functions $p : \mathbb{V} \to (N \to \mathbb{R})$ for LDEs on N. To do so, we borrow ideas and techniques from existing axiomatizations of the Banzhaf index for weighted voting games [29, 53, 60].

The strategy we follow consists in generalizing a known characterization of the Banzhaf index for standard weighted voting due to Barua *et al.* [6]. We use the same axioms of that characterization (Axioms 2-5 below), with the addition of one axiom for so-called dummy agents (Axiom 1). Crucially, however, we show how to adapt the key definitions upon which the axioms are based from the standard weighted voting setting to LDEs. This concerns in particular the definitions of composition and bloc formation (Definitions 16, 17 and 18) which play an important role in the proof. As a result, one can retrieve the known characterization of the standard Banzhaf index from ours, by simply restricting to the class of LDEs where profiles are trivial, and therefore delegations do not matter.

3.2.1 Preliminary Definitions

We start by introducing standard definitions from the theory of simple games. Assume an LDE $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ be given.

Definition 11 (Dummy Agent)**.** *An agent $i \in N$ is dummy if for any $C \subseteq N$ ($i \in C$), $f'_V(C) = f'_V(C \setminus \{i\})$, where f' is the characteristic function of the delegative simple game of V (Definition 9). Let $dum(N)$ denote all dummy agents of the LDE.*

That is, an agent is dummy whenever she cannot influence $f'_V(C)$ by quitting or joining any coalition $C \subseteq N$. It is worth observing that in LDEs there are three ways in which an agent can be dummy: if the agent abstains (i.e., delegates to 0); if the agent is linked by a chain to a delegation cycle; if the agent—call it i—is such that $\sum_{j \in \delta_{\mathbf{d}}(i, d^*_i) \cup \{d^*_i\}} w(j) \geq \beta$, that is, the delegation distance between i and her guru in $\mathbf{d}$ is larger than or equal to β; we call such an agent *distant* (in $\mathbf{d}$). Intuitively, an distant agent i is dummy (cannot become a swing agent) since:

1. she can swing only for a coalition C, where all the intermediaries between she and her guru and her guru are contained in C, otherwise her weight cannot be count; but

2. $C \setminus \{i\}$ never weighs less than β.

Definition 12 (Dictator). *An agent $i \in N$ is a dictator if for any $C \subseteq N$, $f'_V(C) = 1$ if and only if $i \in C$.*

That is, an agent i is a dictator of V whenever it belongs to all and only the winning coalitions of the delegative simple game of V. In an LDE this occurs if the dictator i is a guru and $\beta \leq w(i)$, that is, i meets the quota on her own.

Definition 13 (Symmetric Agents). *Any two agents $i, j \in N$ are symmetric if for all $C \subseteq N \backslash \{i, j\}$, $f'_V(C \cup \{i\}) = f'_V(C \cup \{j\})$.*

Symmetric agents are swing for exactly the same coalitions in the delegative simple game of V. Note that a pair of symmetric agents do not necessarily have the same weight.

Example 14 (Example 13 continued). *Consider V_1 in Figure 3.2a. Since $\beta = 3$ and the delegation distance $\Delta_{\mathbf{d}}(1, 4) = 3$, agent 1 is a distant (and therefore dummy) agent. Next consider agents 1 and 2 (or any pair of $\{1, 2, 3\}$) in V_2, each of whom directly delegates to agent 4. For any coalition $C \subseteq N \setminus \{1, 2\}$, $f'_{V_2}(C \cup \{1\}) = f'_{V_2}(C \cup \{2\})$, thus 1 and 2 are symmetric. There is no dictator in Example 13.*

The following definitions generalize the standard theory of simple games to account for delegations.

Definition 14 (Minimally Winning Coalition). *A coalition $C \subseteq N$ is a minimally winning coalition if for any $i \in De^*(C, \mathbf{d})$, $f'_V(C) = 1$ and $f'_V(C \backslash \{i\}) = 0$.*

That is, a coalition C is minimally winning if it is winning (in the delegative simple game of V), but becomes losing if any agent who is linked to a guru in C via agents in C is removed. So a minimally winning coalition is a coalition that contains just enough gurus with just enough support through intermediaries in the same coalition to meet the quota. It follows that no distant agent may be included in a minimally winning coalition. Notice, however, that a minimally winning coalition may contain agents that are not linked to gurus in C by intermediaries in C (i.e., that do not belong to $De^*(C, \mathbf{d})$) and therefore it may not be minimal with respect to set inclusion.

Definition 15 (Unanimity Liquid Democracy Election (ULDE)). *$V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ is a unanimity LDE if the quota $\beta = \sum_{i \in Gu(\mathbf{d})} w(i, \mathbf{d})$. We call such a quota unanimity quota and denote it by β^U.*

That is, in a ULDE, the quota equals the sum of weights of all agents who directly or indirectly delegate to gurus.

The last two definitions concern operations on LDEs: how to combine two LDEs into a new one; and how to build an LDE from another one by merging two agents into a 'bloc'.

Definition 16 (Conjunction Composition)**.** *Let two LDEs* $V_1 = \langle N_1, \mathbf{w}_1, \mathbf{d}_1, \beta_1 \rangle$ *and* $V_2 = \langle N_2, \mathbf{w}_2, \mathbf{d}_2, \beta_2 \rangle$ *be given, such that for any* $i \in N_1 \cap N_2$*:*

1. *if* $\mathbf{d}_1(i) = j$ *and* $j \in N_2$, $\mathbf{d}_2(i) = j$, *otherwise* $\mathbf{d}_2(i) = 0$;
2. *if* $\mathbf{d}_2(i) = j$ *and* $j \in N_1$, $\mathbf{d}_1(i) = j$, *otherwise* $\mathbf{d}_1(i) = 0$; *and*
3. *and* $w_1(i) = w_2(i)$.

We define the conjunction composition ($\wedge$ *composition*) *of* V_1 *and* V_2 *as* $V_1 \wedge V_2 = \langle N_1 \cup N_2, \mathbf{w}_{1\wedge 2}, \mathbf{d}_{1\wedge 2}, \beta_1 \wedge \beta_2 \rangle$, *where:*

- *for any* $i \in N_1$, $w_{1\wedge 2}(i) = w_1(i)$, *and for any* $i \in N_2$, $w_{1\wedge 2}(i) = w_2(i)$;
- *for any* $i \in N_1 \setminus N_2$, $\mathbf{d}_{1\wedge 2}(i) = \mathbf{d}_1(i)$, *for any* $i \in N_2 \setminus N_1$, $\mathbf{d}_{1\wedge 2}(i) = \mathbf{d}_2(i)$, *and for any* $i \in N_1 \cap N_2$, *if* $\mathbf{d}_1(i) = \mathbf{d}_2(i)$ *then* $\mathbf{d}_{1\wedge 2}(i) = \mathbf{d}_1(i) = \mathbf{d}_2(i)$, *otherwise* $\mathbf{d}_{1\wedge 2}(i) = \mathbf{d}_k(i)$ *where* $k \in \{1, 2\}$ *and* $\mathbf{d}_k(i) \neq 0$;
- $\beta_1 \wedge \beta_2$ *is met by* $C \subseteq N_1 \cup N_2$ *if and only if* $\sum_{i \in De^*(C \cap N_1, \mathbf{d}_1)} w_1(i) \geq \beta_1$ and $\sum_{i \in De^*(C \cap N_2, \mathbf{d}_2)} w_2(i) \geq \beta_2$.

Definition 17 (Disjunction Composition)**.** *Let two LDEs* $V_1 = \langle N_1, \mathbf{w}_1, \mathbf{d}_1, \beta_1 \rangle$ *and* $V_2 = \langle N_2, \mathbf{w}_2, \mathbf{d}_2, \beta_2 \rangle$ *be given, such that for any* $i \in N_1 \cap N_2$*:*

1. *if* $\mathbf{d}_1(i) = j$ *and* $j \in N_2$, $\mathbf{d}_2(i) = j$, *otherwise* $\mathbf{d}_2(i) = 0$;
2. *if* $\mathbf{d}_2(i) = j$ *and* $j \in N_1$, $\mathbf{d}_1(i) = j$, *otherwise* $\mathbf{d}_1(i) = 0$; *and*
3. *and* $w_1(i) = w_2(i)$.

We define the disjunction composition ($\vee$ *composition*) *of* V_1 *and* V_2 *as* $V_1 \vee V_2 = \langle N_1 \cup N_2, \mathbf{w}_{1\vee 2}, \mathbf{d}_{1\vee 2}, \beta_1 \vee \beta_2 \rangle$, *where:*

- *for any* $i \in N_1$, $w_{1\vee 2}(i) = w_1(i)$, *and for any* $i \in N_2$, $w_{1\vee 2}(i) = w_2(i)$.
- *for any* $i \in N_1 \setminus N_2$, $\mathbf{d}_{1\vee 2}(i) = \mathbf{d}_1(i)$, *for any* $N_2 \setminus N_1$, $\mathbf{d}_{1\vee 2}(i) = \mathbf{d}_2(i)$, *and for any* $i \in N_1 \cap N_2$, *if* $\mathbf{d}_1(i) = \mathbf{d}_2(i)$ *then* $\mathbf{d}_{1\vee 2}(i) = \mathbf{d}_1(i) = \mathbf{d}_2(i)$, *otherwise* $\mathbf{d}_{1\vee 2}(i) = \mathbf{d}_k(i)$ *where* $k \in \{1, 2\}$ *and* $\mathbf{d}_k(i) \neq 0$;
- $\beta_1 \vee \beta_2$ *is met by* $C \subseteq N_1 \cup N_2$ *if and only if* $\sum_{i \in De^*(C \cap N_1, \mathbf{d}_1)} w_1(i) \geq \beta_1$ or $\sum_{i \in De^*(C \cap N_2, \mathbf{d}_2)} w_2(i) \geq \beta_2$.

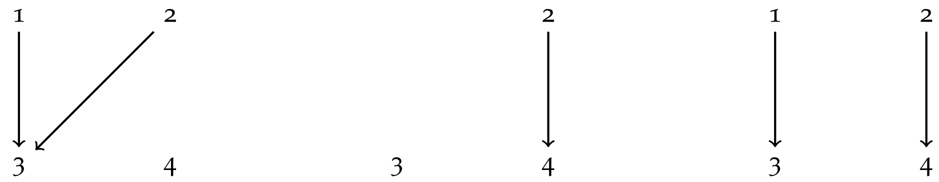


Figure 3.3: The delegation profiles in Example 15. Left: the original $\mathbf{d}_1$. Middle: $\mathbf{d}_2$. Right: the revised $\mathbf{d}_1$.

Two LDEs can be composed provided the delegation graphs at their intersection coincide or, if they do not, provided that this is because an agent in $N_1 \cap N_2$ delegates outside the intersection under one profile while she abstains (i.e., delegates to 0) under the other profile. The condition is required to guarantee the coherency of delegations in the composition. We use the following example to illustrate this conditions.

Example 15. *Consider two LDEs V_1 with $N_1 = \{1,2,3,4\}$ and $\mathbf{w}_1 = (1,1,2,1)$, and V_2 with $N_2 = \{2,3,4\}$ and $\mathbf{w}_2 = (1,1,1)$; the delegation graphs of $\mathbf{d}_1$ and $\mathbf{d}_2$ are depicted in Figure 3.3 left and middle. These two LDEs cannot be composed since:*

1. *The weights of agent 3 in the two LDEs are not consistent. Since agent 3 is in the intersection of N_1 and N_2, she should have identical weight in both LDEs to guarantee that we can clearly define her weight in the composition LDE.*

2. *The delegation strategy of agent 2 is not consistent in the two LDEs. All of the agents 2, 3 and 4 are in the intersection of N_1 and N_2, however, it is unclear what the delegation strategy of agent 2 is in either $V_1 \wedge V_2$ or $V_1 \vee V_2$, since $\mathbf{d}_1(2) = 3$ but $\mathbf{d}_2(2) = 4$.*

However, if we revise V_1 so that $\mathbf{w}_1 = (1,1,1,1)$ and $\mathbf{d}_1$ as shown in Figure 3.3 right, then V_1 and V_2 can be composed. Agent 1 is not relevant for whether the two LDEs can be composed or not, since she does not belong to $N_1 \cap N_2$.

Subsequently, the quotas in the composition are so defined as to guarantee the following:

1. Coalitions in the delegative simple game of the conjunction composition are winning if and only if they are winning in both of the delegative simple games of the LDEs; or

2. Coalitions in the delegative simple game of the disjunction composition are winning if and only if they are winning in at least one of the delegative simple games of the LDEs.

Note that, by abuse of notation, $\beta_1 \wedge \beta_2$ and $\beta_1 \vee \beta_2$ are not real numbers as defined in Section 3.1.1. This property is crucial in the proof of Lemma 1.

Definition 18 (Bloc formation). *Given $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ and for any $i, j \in N$ such that $d_i = j$ or $i, j \in Gu(\mathbf{d})$, $V' = \langle N', \mathbf{w}', \mathbf{d}', \beta \rangle$ is called the bloc LDE joining i and j into a bloc ij, where*

- $N' = N \backslash \{i, j\} \cup \{ij\}$;
- *For $\mathbf{d}'$, if $d_i = j$, $d'_{ij} = d_j$, and for any $a \in N$, such that $d_a = i$ or $d_a = j$, $d'_a = ij$, but if $i, j \in Gu(\mathbf{d})$, for any $a \in N$ such that $d_a = i$ or $d_a = j$, $d'_a = ij$;*
- $w'(ij) = w(i) + w(j)$.

A bloc LDE treats two agents i and j, who are either adjacent on a delegation chain or both gurus, as one new agent ij. By applying the operation in Definition 18 repeatedly, it is possible to coalesce all agents who share the same guru into one bloc. Furthermore, any pair of delegation chains can also be joined into one bloc by joining their gurus into one bloc. Such operations play an important role in the proof of our characterization result (in particular, in Lemma 2).

3.2.1.1 *Axioms*

We can now introduce the axioms on the power index function p of our characterization. Assume again that an LDE $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ is given.

The first three axioms consider individual agents. We assign minimum power (i.e., 0) to dummy agents, maximum (i.e., 1) to dictators, and identical power to symmetric agents:

Axiom 1 (No Power (**NP**)). *If $i \in dum(N)$, $p_i(V) = 0$.*

Axiom 2 (Maximum Power (**MP**)). *If i is a dictator, $p_i(V) = 1$.*

Axiom 3 (Equal Treatment (**ET**)). *For any pair of symmetric agents $i, j \in N$, $p_i(V) = p_j(V)$.*

The last two axioms concern how the power index should behave with respect to composition and bloc formation.

Axiom 4 (Bloc Principle (**BP**)). *For any two agents $i, j \in N$ such that $d_i = j$, or $i, j \in Gu(\mathbf{d})$, let V' be the bloc LDE by joining i and j into bloc ij. Then $p_{ij}(V') = p_i(V) + p_j(V)$.*

Axiom 5 (Sum Principle (**SP**)). *For any pair of LDEs $V_1 = \langle N_1, \mathbf{w}_1, \mathbf{d}_1, \beta_1 \rangle, V_2 = \langle N_2, \mathbf{w}_2, \mathbf{d}_2, \beta_2 \rangle \in \mathbb{V}$, such that any $i \in N_1 \cap N_2$ satisfies the conditions in Definition 16 and Definition 17:*

1. *if $\mathbf{d}_1(i) = j$ and $j \in N_2$, $\mathbf{d}_2(i) = j$, otherwise $\mathbf{d}_2(i) = 0$;*
2. *if $\mathbf{d}_2(i) = j$ and $j \in N_1$, $\mathbf{d}_1(i) = j$, otherwise $\mathbf{d}_1(i) = 0$; and*

3. *and* $w_1(i) = w_2(i)$,

we have $p_i(V_1 \wedge V_2) + p_i(V_1 \vee V_2) = p_i(V_1) + p_i(V_2)$ *for any* $i \in N_1 \cup N_2$.

Intuitively, Axiom 4 describes the situation where we can form a bloc from two agents, who satisfy the condition that the two agents are two gurus, or two adjacent agents on a delegation chain. This condition guarantees that in the resulting delegation profile (in the bloc LDE), no confusing delegation profile is produced, e.g., the formed bloc delegates to multiple neighbors, or new delegation cycles are formed. Then the power of the bloc is the sum of the voting powers of the individual agents. Axiom 5 requires that the sum of any agent's power in the two LDEs $V_1 \wedge V_2$ and $V_1 \vee V_2$ to be the sum of her power in V_1 and V_2.

3.2.1.2 *Characterization*

The characterization is based on two lemmas. We start by fixing some auxiliary notation. Let an LDE $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ be given. For any $i \in N$, let $(\mathbf{d}_{-i}, d'_i)$ be the profile in which any $j \in N \backslash \{i\}$ delegates as in $\mathbf{d}$, while agent i chooses delegation d'_i. Furthermore, $\mathbf{d}_C$ denotes the profile restricted to a coalition $C \subseteq N$. Such a restricted profile is a mapping $\mathbf{d}_C : C \to C \cup \{0\}$ defined as follows, for all $i \in C$:

$$\mathbf{d}_C(i) = \begin{cases} \mathbf{d}(i), & \text{if } \mathbf{d}(i) \in C \\ 0, & \text{otherwise} \end{cases}. \tag{3.6}$$

That is, in $\mathbf{d}_C$ all agents in C either delegate to agents in C or abstain. Recall the notation $De^*(C, \mathbf{d})$ (Equation 3.2), i.e., the set of all agents that delegate to some guru in $C \subseteq N$ via a delegation chain contained in C, in $\mathbf{d}$. The same set, for a different profile $\mathbf{d}'$, is denoted as $De^*(C, \mathbf{d}')$.

Lemma 1. DB *satisfies* ***MP***, ***SP***, ***BP***, ***ET***, *and* ***SP***.

Proof. We start with the **MP** axiom, assume that agent i is a dictator in LDE $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$. Then, for any coalition $C \subseteq N \backslash \{i\}$, $f'_V(C) = 0$ and $f_V(C \cup \{i\}) = 1$ by Definition 12. This implies that i is a swing agent for all coalitions in $\{C \cup \{i\} | C \subseteq N \setminus \{i\}\}$, in which the number of elements is 2^{n-1}. Thus we obtain $\mathsf{DB}_i(V) = 1$.

Now **NP** simply follows, because for any $i \in dum(N)$, $\mathsf{DB}_i(V) = \sum_{C \subseteq N \backslash \{i\}} (f'_V(C \cup \{i\}) - f'_V(C))/2^{n-1} = 0$ by Definition 11.

To show that DB satisfies the **SP** axiom, one first has to show that by the way in which weights $\beta_1 \wedge \beta_2$ and $\beta_1 \vee \beta_2$ are set in Definition 16 and Definition 17, we have that for any coalition $C \subseteq N_1 \cup N_2$, $f'_{V_1 \wedge V_2}(C) = 1$ iff $f'_{V_1}(C \cap N_1) = 1$ *and* $f'_{V_2}(C \cap N_2) = 1$, and $f'_{V_1 \vee V_2}(C) = 1$ iff $f'_{V_1}(C \cap N_1) = 1$ *or* $f'_{V_2}(C \cap N_2) = 1$.

The proof can then proceed with a standard argument, which considers agents in $N_1 \setminus N_2$, $N_1 \cap N_2$, and $N_2 \setminus N_1$, respectively.

We first consider any $i \in N_1 \setminus N_2$, i.e., agent i is contained in N_1 but not in N_2. Let m_i^V denote the number of times that agent i is swing in the delegative simple game for V, i.e., $m_i^V = |\{C \subseteq N \setminus \{i\} \mid f'_V(C) = 0, f'_V(C \cup \{i\}) = 1\}|$. Then, if i is swing in $C \subseteq N_1$ in LDE V_1, she is also swing in $(C \cup C') \cap N_1$, for any $C' \subseteq N_2 \setminus N_1$, since C' does not influence the value of f'_{V_1}. Therefore, in LDE $V_1 \vee V_2$, we have $m_i^{V_1 \vee V_2} = m_i^{V_1} 2^{|N_2 \setminus N_1|}$, where $m_i^{V_1 \vee V_2}$ is the number of times that i is swing in LDE $V_1 \vee V_2$. Additionally, since $i \in N_1 \setminus N_2$, $m_i^{V_2} = 0$, that is, i cannot be swing in LDE V_2, which implies $m_i^{V_1 \wedge V_2} = 0$. Hence we have for $i \in N_1 \setminus N_2$ that $m_i^{V_1 \vee V_2} = m_i^{V_1} 2^{|N_2 \setminus N_1|} + m_i^{V_2} 2^{|N_1 \setminus N_2|} - m_i^{V_1 \wedge V_2}$. Identical equations can be developed for agent $i \in N_2 \setminus N_1$ or $i \in N_1 \cap N_2$. We then divide each side of the equation by $2^{|N_1 \cup N_2|-1}$ and obtain that, for any $i \in N_1 \cup N_2$, $\frac{m_i^{V_1 \vee V_2}}{2^{|N_1 \cup N_2|-1}} = \frac{m_i^{V_1}}{2^{|N_1|-1}} + \frac{m_i^{V_2}}{2^{|N_2|-1}} - \frac{m_i^{V_1 \wedge V_2}}{2^{|N_1 \cup N_2|-1}}$, which implies that $\mathsf{DB}_i(V_1 \wedge V_2) + \mathsf{DB}_i(V_1 \vee V_2) = \mathsf{DB}_i(V_1) + \mathsf{DB}_i(V_2)$ as desired.

To prove the **BP** axiom, we rewrite $\mathsf{DB}_i(V) + \mathsf{DB}_j(V)$ as:

$$
\begin{aligned}
&2^{n-1}(\mathsf{DB}_i(V) + \mathsf{DB}_j(V)) \\
=&\sum_{C \subseteq N \setminus \{i\}} (f'_V(C \cup \{i\}) - f'_V(C)) + \sum_{C \subseteq N \setminus \{j\}} (f'_V(C \cup \{j\}) - f'_V(C)) \\
=&\sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{i\}) - f'_V(C) + f'_V(C \cup \{i,j\}) - f'_V(C \cup \{j\})) \\
&+ \sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{j\}) - f'_V(C) + f'_V(C \cup \{i,j\}) - f'_V(C \cup \{i\})) \\
=&2 \sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{i,j\}) - f'_V(C)).
\end{aligned}
$$

Since $\mathsf{DB}_{ij}(V') = 1/2^{n-2} \sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{i,j\}) - f'_V(C))$, we have that $\mathsf{DB}_i(V) + \mathsf{DB}_j(V) = \mathsf{DB}_{ij}(V')$, where V' is the bloc LDE by forming i and j into a bloc.

As for **ET**, assume that i and j are symmetric agents. We show that whenever i is a swing agent, so is j, and vice versa. Then i serves as a swing agent for any coalition which contains agent j, or not, as shown in the following two cases:

(1) For any $C \subseteq N \setminus \{i,j\}$, such that $f'_V(C \cup \{i\}) - f'_V(C) = 1$, by Definition 13, we obtain

$$f'_V(C \cup \{j\}) - f'_V(C) = f'_V(C \cup \{i\}) - f'_V(C) = 1.$$

(2) For any $C \subseteq N \backslash \{i,j\}$, such that $f'_V(C \cup \{i,j\}) - f'_V(C \cup \{j\}) = 1$, by Definition 13, we obtain that

$$f'_V(C \cup \{i,j\}) - f'_V(C \cup \{i\}) = f'_V(C \cup \{i,j\}) - f'_V(C \cup \{j\}) = 1.$$

That is, each time i serves as a swing agent, j also serves as a swing agent once. By a similar argument, it can be shown that each time j serves as a swing agent, i also serves as a swing agent once. Then $\mathsf{DB}_i(V) = \mathsf{DB}_j(V)$. □

Lemma 2. *If a power index p for LDEs satisfies **NP**, **MP**, **ET**, **BP**, and **SP**, then p is* DB.

Proof. We start by introducing the following claim.

Claim 1. *A power index p for unanimity LDEs satisfies **MP**, **NP**, **SP**, **ET**, and **BP**, only if it is* DB.

The proof is obtained by first showing that Lemma 2 holds if Claim 1 holds and showing that Claim 1 holds.

First, we show that if p is DB for all unanimity LDEs, then the power index p is DB for all LDEs. Assume that an arbitrary LDE V is given and let $\mathscr{C} = \{C_1, \ldots, C_m\}$ denote all minimally winning coalitions (recall Definition 14). Notice that any winning coalition can be represented as the union of a subset of $\mathscr{C}$. Hence V can be represented as the disjunction of m unanimity LDEs, i.e., $V = V_1 \vee V_2 \vee \cdots \vee V_m$ where $V_j = \langle C_j, \mathbf{w}, \mathbf{d}_{C_j}, \beta^U \rangle$ $(1 \leq j \leq m)$ is a unanimity LDE. Observe that any agent's delegation strategy is consistent in all unanimity games, that is, the condition of Definition 17 is satisfied: for any pair of LDEs $V_i, V_j \in \{V_1, \ldots, V_m\}$, for any agent $k \in C_i \cap C_j$, if $d_{C_i}(k) = h$ (resp. $d_{C_j}(k) = h$) and $h \in C_j$ (resp. $h \in C_i$), $d_{C_j}(k) = h$ (resp. $d_{C_i}(k) = h$), otherwise $d_{C_j}(k) = 0$ (resp. $d_{C_i}(k) = 0$).

We prove by induction on the size of the disjunction composition of unanimity LDEs. As the basis, $p_i(V_j)$ is DB by the assumption that p is DB for any unanimity LDE given in Claim 1, where $i \in N$ and $1 \leq j \leq m$.

Henceforth, we assume that for any LDE which is the disjunction of k $(k < m)$ unanimity games in $\{V_1, \ldots, V_m\}$ p is equivalent to DB, then prove that for any LDE which is the disjunction of $k+1$ unanimity games in $\{V_1, \ldots, V_m\}$ p is also equivalent to DB. Without loss of generality, assume that $p_i(V_1 \vee \cdots \vee V_k)$ is DB, and we prove that $p_i(V_1 \vee \cdots \vee V_k \vee V_{k+1})$ is also DB. By **SP**, we have $p_i(V_1 \vee \cdots \vee V_k \vee V_{k+1}) = p_i(V_1 \vee \cdots \vee V_k) + p_i(V_{k+1}) - p_i((V_1 \vee \cdots \vee V_k) \wedge V_{k+1})$.

Observe that if an agent belongs to $(V_1 \vee \cdots \vee V_k) \wedge V_{k+1}$, then she belongs to the intersection of V_{k+1} with at least one of $V_1, \ldots,$ and V_k. That is, the agent belongs to $(V_1 \wedge V_{k+1}) \vee \cdots \vee (V_k \wedge V_{k+1})$. Therefore, we have that $(V_1 \vee \cdots \vee V_k) \wedge V_{k+1} = (V_1 \wedge V_{k+1}) \vee \cdots \vee (V_k \wedge V_{k+1})$ (distributive law). Since V_j is a unanimity LDE, $V_j \wedge V_{k+1}$ is equivalent to the unanimity LDE $\langle C_j \cup C_{k+1}, \mathbf{w}, \mathbf{d}_{C_j \wedge C_{k+1}}, \beta^U \rangle$.

Therefore, by the assumption that p is DB for each disjunction of k unanimity LDEs, we have p is DB for $(V_1 \vee \cdots \vee V_k) \wedge V_{k+1}$. Hence it implies that p is DB for $V_1 \vee \cdots \vee V_k \vee V_{k+1}$. Intuitively, the number of times that any agent $i \in \bigcup_{1 \leq j \leq k+1} C_j$ serves as a swing agent in $V_1 \vee \cdots \vee V_k \vee V_{k+1}$ is the sum of her swing times in $V_1 \vee \cdots \vee V_k$ and V_{k+1}, subtracting her swing times in $(V_1 \vee \cdots \vee V_k) \wedge V_{k+1}$. Therefore, we proved that if Claim 1 holds, Lemma 2 holds automatically.

Next, we prove Claim 1 by induction on the size of the agent set. Consider an arbitrary unanimity LDE $\dot{V} = \langle N, \mathbf{w}, \mathbf{d}, \beta^U \rangle$. Let $\hat{N} = N \setminus dum(N)$ denote all non-dummy agents, and $\hat{n} = |\hat{N}|$. As the basis of the induction, consider the case in which there is only one agent, i.e., $N = \{i\}$. If i is a dummy agent, then by **NP**, $p_i(\dot{V}) = 0$. On the other hand, if $i \notin dum(N)$, then i is a dictator, which implies that $p_i(\dot{V}) = 1 = 1/2^{\hat{n}-1}$ due to **MP**.

Then for the induction step, we assume that p is DB if $|N| = k$ ($k \in \mathbb{N}_+$, i.e., positive integer), and prove that p is also DB if $|N| = k+1$. That is in $\dot{V}$ ($|N| = k+1$), we prove that for any $i \in \hat{N}$, $p_i(\dot{V}) = 1/2^{\hat{n}-1}$ and for any $i \in dum(N)$, $p_i(\dot{V}) = 0$, which is identical to DB. For any unanimity LDE, let's consider three exhaustive cases: (1) all agents are dummy agents, (2) only one non-dummy agent exists, and (3) more than one non-dummy agents exist in the unanimity LDE.

Case 1. *$N = dum(N)$. That is, all agents are dummy agents. Then by **NP**, for all $i \in N$, $p_i(\dot{V}) = 0$.*

Case 2. *$|dum(N)| = k$. In this case, there is only one non-dummy agent, denoted by i. Then i is a dictator, and $p_i(\dot{V}) = 1 = 1/2^{\hat{n}-1}$ by **MP**. On the other hand, for any $j \in dum(N)$, $p_i(\dot{V}) = 0$ by **NP**.*

Case 3. *$|N| - |dum(N)| > 1$. That is there are more than one non-dummy agents in the unanimity LDE. Among these non-dummy agents, there must exist at least one guru, otherwise all agents become dummy due to lack of representation. Therefore, we consider three exhaustive subcases according to the number of gurus in the set of non-dummy agents: (1) all non-dummy agents are gurus, (2) there is only one guru among non-dummy agents, and (3) there are more than one gurus among the non-dummy agents, but not all non-dummy agents are gurus, i.e., at least one non-dummy agent is a delegator.*

Case 3.1. *$\forall i \in N \setminus dum(N)$, $i \in N^{\mathbf{d}}$. That is, any non-dummy agent is a guru. Let $i, j \in N \setminus dum(N)$ such that $i \neq j$, then we form i and j into a bloc and obtain the bloc LDE V'. Observe that V' has k agents. Therefore, by assumption and **BP**, $p_i(\dot{V}) + p_j(\dot{V}) = p_{ij}(V') = 1/2^{\hat{n}-2}$. Since i and j are symmetric in $\dot{V}$, $p_i(V') = p_j(V') = 1/2^{\hat{n}-1}$ due to **ET**. Moreover, any agent $i' \in \hat{N} \setminus \{i, j\}$ is symmetric with i (or j), thus $p_{i'}(V') = p_i(V') = 1/2^{\hat{n}-1}$, and for any $a \in dum(N)$, $p_a(\dot{V}) = 0$ by **NP**.*

Case 3.2. *$Gu(\mathbf{d}) = \{i\}$. In this case, there is only one guru, which is i, and any other delegator has i as their guru. Therefore, for all $j \in N \setminus dum(N)$ such that $j \neq i$, $\mathbf{d}^*(j) =$*

i. Assume $j \in N$, such that $d_j = i$. Then we obtain a bloc game V' by forming i and j into a bloc ij. By ***BP*** *and the assumption, $p_i(\dot{V}) + p_j(\dot{V}) = p_{ij}(V') = 1/2^{\hat{n}-2}$. Then, since* ***ET***, *$p_i(\dot{V}) = p_j(\dot{V}) = 1/2^{\hat{n}-1}$. Additionally, for all $a \in dum(N)$, $p_a(\dot{V}) = 0$ due to* ***NP***, *and for all $a \in N \setminus dum(N)$, $p_a(\dot{V}) = 1/2^{\hat{n}-1}$ due to* ***ET***.

Case 3.3. *$|Gu(\mathbf{d})| > 1$ and $Gu(\mathbf{d}) \subset N \setminus dum(N)$. In this case there are more than one gurus and at least one delegator delegates to one of the gurus. We can then apply similar arguments to those provided for Case 3.1 or Case 3.2 to join agents into a bloc and thus prove that f is equivalent to* DB.

This completes the proof of Claim 1, and the whole proof. □

3.3 FURTHER PROPERTIES OF THE DELEGATIVE BANZHAF INDEX

The above axioms are somewhat technical. In this section we focus on properties of the delegative Banzhaf power index that have direct intuitive appeal. These are: *power loss by delegation*, *power monotonicity*, *direct delegations are better*, *short chains are better* and *equal power in unanimity LDE*. The property *power loss by delegation* (Fact 1) shows that to be a guru is always weakly better than to delegate in terms of voting power. Properties *power monotonicity* (Fact 2), *direct delegations are better* (Fact 3), and *short chains are better* (Fact 4) illustrates that more direct delegations, or short delegation chains, are preferred by both delegators and gurus. Finally for unanimity LDEs, *equal power in unanimity LDE* shows that each agent has equal voting power.

Fact 1 (Power Loss by Delegation)**.** *For any pair of LDEs $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ and $V' = \langle N, \mathbf{w}, \mathbf{d}', \beta \rangle$, such that $\mathbf{d}' = (\mathbf{d}_{-i}, d'_i)$, $\mathbf{d}(i) = i$ and $d'_i = j$ ($i \neq j$), we have that* $\mathsf{DB}_i(V) \geq \mathsf{DB}_i(V')$.

Proof. To prove the fact, it is sufficient to prove that, for any coalition $C \subseteq N$ with $i \in C$, if i is not a swing agent for C in V, neither is she a swing agent for C in V'. Towards a contradiction, we assume that i is a swing agent for C in V' but i is not a swing agent in V. Then we have that $\sum_{a \in De^*(C,\mathbf{d})} w(a) \geq \beta$, and $\sum_{a \in De^*(C \setminus \{i\},\mathbf{d})} w(a) = \sum_{a \in De^*(C,\mathbf{d})} w(a) - w(i) < \beta$. Since the only difference between $\mathbf{d}$ and $\mathbf{d}'$ is the strategy of i, we have $\sum_{a \in De^*(C \setminus \{i\},\mathbf{d})} w(a) = \sum_{a \in De^*(C \setminus \{i\},\mathbf{d})} w(a)$, i.e., the weight of $C \setminus \{i\}$ is identical in both LDEs V and V'. Moreover, since i is a guru in V and $i \in C$, it holds that $\sum_{a \in De^*(C),\mathbf{d}} w(a) = \sum_{a \in De^*(C \setminus \{i\},\mathbf{d})} w(a) + w(i) = \sum_{a \in De^*(C \setminus \{i\},\mathbf{d})} w(a) + w(i) \geq \beta$, which contradicts the assumption that i is not a swing agent in V. □

Fact 1 implies that delegations never lead to an increase in power for the delegator. In fact, one can show that the inequality $\mathsf{DB}_i(V) \geq \mathsf{DB}_i(V')$ can be strict.

The next two facts show that agents closer to the guru have more power than those farther away and that, power-wise, delegating directly to a guru is better than doing that indirectly.

Fact 2 (Power Monotonicity). *For any pair of agents $i, j \in N$, such that $d_i = j$, $\mathsf{DB}_i(V) \leq \mathsf{DB}_j(V)$.*

Proof. We prove the fact by showing that $\mathsf{DB}_i(V) - \mathsf{DB}_j(V) \leq 0$. By the definition of the delegative simple game of V, we substitute $\mathsf{DB}_i(V)$ and $\mathsf{DB}_j(V)$ in $\mathsf{DB}_i(V) - \mathsf{DB}_j(V) \leq 0$ as follows.

$$\begin{aligned}
&2^{n-1}(\mathsf{DB}_j(V) - \mathsf{DB}_i(V))\\
=&\sum_{C \subseteq N \setminus \{j\}} (f'_V(C \cup \{j\}) - f'_V(C)) - \sum_{C \subseteq N \setminus \{i\}} (f'_V(C \cup \{i\}) - f'_V(C))\\
=&\sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{j\}) - f'_V(C) + f'_V(C \cup \{i,j\}) - f'_V(C \cup \{i\}))\\
&- \sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{i\}) - f'_V(C) + f'_V(C \cup \{i,j\}) - f'_V(C \cup \{j\}))\\
=&2 \sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{j\}) - f'_V(C \cup \{i\})).
\end{aligned}$$

Concerning the above equation, for any $C \subseteq N \setminus \{i, j\}$, we consider two possible cases:

1. $f'_V(C \cup \{j\}) = 0$.
   This implies that $\sum_{a \in De^*(C \cup \{j\}, \mathbf{d})} w(a) < \beta$, and consequently, $\sum_{a \in De^*(C, \mathbf{d})} w(a) \leq \sum_{a \in De^*(C \cup \{j\}, \mathbf{d})} w(a) < \beta$. Since $d_i = j$ and $j \notin C$, we have that $i \notin De^*(C \cup \{i\}, \mathbf{d})$. Therefore, we have $\sum_{a \in De^*(C \cup \{i\}, \mathbf{d})} w(a) = \sum_{a \in De^*(C, \mathbf{d})} w(a) < \beta$, which implies that $f'_V(C \cup \{i\}) = 0$. Hence $f'_V(C \cup \{j\}) - f'_V(C \cup \{i\}) = 0$.

2. $f_V(C \cup \{j\}) = 1$.
   This implies that $\sum_{a \in De^*(C \cup \{j\}, \mathbf{d})} w(a) \geq \beta$. Now we consider two possible cases:

   a) $\sum_{a \in De^*(C, \mathbf{d})} w(a) < \beta$.
      Since $i \notin De^*(C \cup \{i\}, \mathbf{d})$, it can be inferred that $\sum_{a \in De^*(C \cup \{i\}, \mathbf{d})} w(a) = \sum_{a \in De^*(C, \mathbf{d})} w(a) < \beta$, which implies that $f'_V(C \cup \{i\}) = 0$. Therefore, $f'_V(C \cup \{j\}) - f'_V(C \cup \{i\}) > 0$.

   b) $\sum_{a \in De^*(C, \mathbf{d})} w(a) \geq \beta$.
      We can obtain that $\sum_{a \in De^*(C \cup \{i\}, \mathbf{d})} w(a) \geq \sum_{a \in De^*(C, \mathbf{d})} w(a) \geq \beta$, which

implies that $f'_V(C \cup \{i\}) = 1$. Therefore, $f'_V(C \cup \{j\}) - f'_V(C \cup \{i\}) = 0$.

Hence, to sum up, we have

$$\sum_{C \subseteq N \setminus \{i,j\}} (f'_V(C \cup \{j\}) - f'_V(C \cup \{i\})) \geq 0,$$

which implies that

$$\mathsf{DB}_j(V) - \mathsf{DB}_i(V) \geq 0.$$

This completes the proof. □

Fact 3 (Direct Delegations Are Better). *Let $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ be a LDE where N contains three agents i, j and k such that $\mathbf{d}(i) = j$ and $\mathbf{d}(k) = i$. Let $V' = \langle N, \mathbf{w}, \mathbf{d}', \beta \rangle$, such that $\mathbf{d}' = (\mathbf{d}_{-k}, d'_k)$ where $d'_k = j$. Then $\mathsf{DB}_k(V') \geq \mathsf{DB}_k(V)$.*

Proof. It is sufficient to show that if k is a swing agent for coalition C in LDE V, then she is also a swing agent for C in V'. We have $\sum_{a \in De^*(C,\mathbf{d})} w(a) \geq \beta$ and $\sum_{a \in De^*(C \setminus \{k\},\mathbf{d})} w(a) < \beta$. This implies that $k \in De^*(C, \mathbf{d})$, from which we can infer that $i, j \in De^*(C, \mathbf{d})$. Note that the only difference between $\mathbf{d}$ and $\mathbf{d}'$ is the strategy of k, and $\mathbf{d}(k) = i$ while $\mathbf{d}'(k) = j$. Therefore, we obtain $k \in De^*(C, \mathbf{d}')$, and consequently $De^*(C, \mathbf{d}) = De^*(C, \mathbf{d}')$. Hence $\sum_{a \in De^*(C,\mathbf{d}')} w(a) \geq \beta$ and $\sum_{a \in De^*(C \setminus \{k\},\mathbf{d}')} w(a) < \beta$, which implies that k is also a swing agent for C in V'. □

Note that the above inequality $\mathsf{DB}_k(V') \geq \mathsf{DB}_k(V)$ becomes strict if k is a swing agent for some coalition C not containing agent i in LDE V'.

It is not only delegators who can benefit from getting closer to gurus, but shorter delegation chains are also better for gurus.

Fact 4 (Short Chains Are Better). *Let $V = \langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$ and $V' = \langle N, \mathbf{w}, \mathbf{d}', \beta \rangle$, such that $d_i = j$ and $d_j = k$, and $\mathbf{d}' = (\mathbf{d}_{-i}, d'_i = k)$. Then we have $\mathsf{DB}_k(V') \geq \mathsf{DB}_k(V)$.*

Proof. We show in this proof that if agent k is swing for a coalition in LDE V, she is also swing for this coalition in LDE V', but not vice versa.

Assume that agent k is swing for coalition $C \subseteq N$ under profile $\mathbf{d}$, i.e., in LDE V. This implies that $k \in De^*(C, \mathbf{d})$ since k contributes voting weight to $De^*(C, \mathbf{d})$, i.e., $\sum_{a \in De^*(C,\mathbf{d})} w(a) \geq \beta$ and $\sum_{a \in De^*(C \setminus \{k\},\mathbf{d})} w(a) < \beta$. Since the only difference between $\mathbf{d}$ and $\mathbf{d}'$ is the delegation strategy of agent i, where $d'_i = k$, and $d_i = j$ and $d_j = k$, we obtain that if k is not contained in a coalition, then i's weight cannot be counted for the weight of the coalition. This immediately indicates that $\sum_{a \in De^*(C \setminus \{k\},\mathbf{d}')} w(a) = \sum_{a \in De^*(C \setminus \{k\},\mathbf{d})} w(a) < \beta$. Then, we consider the weight of coalition C. Observe that we have $\sum_{a \in De^*(C,\mathbf{d}')} w(a) = \sum_{a \in De^*(C,\mathbf{d})} w(a) \geq \beta$ for all cases except for the one where $j \notin C$ but $i \in C$. In this special case, we have that $\sum_{a \in De^*(C,\mathbf{d}')} w(a) = \sum_{a \in De^*(C,\mathbf{d})} w(a) + w(i) \geq \beta$, since the chain between i

and k is broken in C under profile $\mathbf{d}$. That is, the claim "if k is a swing agent for C in LDE V, then she is also a swing agent for C in V'" holds.

However, a case may exist where k swings for coalition $C \subseteq N$, which does not contain j, under $\mathbf{d}'$, and $0 \leq \sum_{a \in De^*(C,\mathbf{d}')} w(a) - \beta < w(i)$. In this case, $\sum_{a \in De^*(C,\mathbf{d})} w(a) = \sum_{De^*(C,\mathbf{d}')} w(a) - w(i) < \beta$. Then k is not a swing agent for C under $\mathbf{d}$. □

The last fact is for unanimous LDEs: all agents have equal power in a ULDE.

Fact 5 (Equal Power in Unanimity LDE). *In any ULDE V, for any pair of agents $i, j \in N$, $\mathsf{DB}_i(V) = \mathsf{DB}_j(V)$.*

Proof. Observe that in any Unanimity LDE V, each agent $i \in N$ can only swing for coalition N, i.e., $f'_V(N \setminus \{i\}) = 0$ and $f'_V(N) = 1$. Therefore $\mathsf{DB}_i(V) = \frac{1}{2^{n-1}}$, which completes the proof. □

CONCLUSION

This chapter developed a power index for voting in liquid democracy. We showed that the index generalizes the Banzhaf index for standard weighted voting and can be axiomatized in a similar fashion (Section 3.2). It was also shown that, as expected, the index is highly dependent to the delegation structure, which is reflected in the facts proved in Section 3.3.

The theory proposed in this chapter has stimulated further interest by researchers working on liquid democracy. In particular, D'Angelo *et al.* [25] study the computational complexity of DB and a similar generalization of the Shapley-Shubik power index (Equation 2.4). They further showed that it is computationally hard to find a solution to a defined voting power bribery problem, where they try to maximize/minimize a given agent's voting power by changing the delegation structure under a given budget constraint.

Different from our work, Colley *et al.* [22] define a Banzhaf index based power index in liquid democracy, called LD-Penrose-Banzhaf index, which does not depend on the prior information of the delegation graph. Instead, the LD-Penrose-Banzhaf index measures the probability that each agent can change the voting result under all possible delegation graphs. They show that it is #*P*-hard to compute the LD-Penrose-Banzhaf index for a group of heterogeneous weight agents. However, it costs pseudo-polynomial time to compute LD-Penrose-Banzhaf index in proxy voting and liquid democracy with a complete social network.

Several research directions remain open and we would like to mention one. First, it would be nice to obtain a characterization of DB based on more natural axioms, such as the properties studied in Section 3.3.

In the next chapter, we incorporate the proposed DB into the game-theoretical model of [11], and investigate the structure of equilibria and agents' behavior in simulations in specific social networks.

4 DELEGATION GAMES WITH POWER

In this chapter, we incorporate the voting power theory of Chapter 3 into the theory of delegation games developed by Bloembergen *et al.* [11] (recall Section 2.4). In the resulting games, agents aim at achieving a trade-off between: (1) delegating to high-accuracy agents, who are able to make correct decisions accurately; and (2) retaining voting power.

CHAPTER CONTRIBUTION We show that, in delegation games of this type, Nash equilibria (NE) cannot be guaranteed to exist in general but they exist under some specific conditions. Finally, we run computer simulations. The empirical results show that considering to retain voting power reduces agents' incentive to delegate, especially via long delegation chains. At the same time, this leads to more equal voting power distribution in the group.

4.1 POWER-SENSITIVE DELEGATION GAMES: DEFINITION

Note again that, like in [11], we will be working with the more general setting of underlying networks: those represented as a directed graph as introduced in Section 2.1.

We then begin by introducing the definition of delegation games.

In the delegation game by Bloembergen *et al.* [11], agents' payoffs in an LDE depend solely on the accuracy of their gurus and the effort agents incur should they vote directly. Here we abstract from the effort element of the model and focus instead on incorporating a power-seeking component in agents' utilities. The key intuition behind our extension is to model agents that are not only interested in voting accurately, but also in their own influence during the vote. So our agents choose their delegations by aiming at maximizing the trade-off between pursuing high accuracy and seeking more power.

Definition 19. *A* power-sensitive delegation game *is a tuple $\mathcal{D} = \langle N, R, \mathbf{q}, \Sigma, \beta, u \rangle$, where: N is a finite set of agents; $R = \langle N, E \rangle$ is a directed graph; $\mathbf{q}$ is the accuracy profile; $\Sigma_i \in N$ is the delegation strategy space of agent i (i.e., agent i's neighborhood*

$R'(i)$ in the underlying network), $\beta \in [0, n]$ is the quota; and u is the utility function, defined as follows:

$$u_i(\mathbf{d}) = \begin{cases} \mathsf{DB}_i(\mathbf{d})^\alpha \cdot q_{d_i^*}, & \textit{if } d_i^* \in N \\ 0, & \textit{otherwise} \end{cases}, \tag{4.1}$$

where $\alpha \in [0, 1]$. Note that, in the above utility function, if agent i is linked to a delegation cycle in $\mathbf{d}$, she has no guru, i.e., $d_i^* \notin N$, and hence her utility is 0 since $\mathsf{DB}_i(\mathbf{d}) = 0$ because she is a dummy agent (recall Axiom 1). Observe that we modify the definition of delegation games in [11] (Definition 3) in three ways:

1. $\mathbb{P}$ is degenerated to single point distribution: each agent benefits equally from choosing the true alternative.
2. Voting does not cost effort in our definition.
3. The power-sensitive component $\mathsf{DB}_i(\mathbf{d})^\alpha$ is added.

Intuitively, the power-sensitive component in Equation 4.1 denotes the extent that agents are motivated to retain power: parameter α is used to control how much agents are influenced by power in the range going from no influence to influence equal to that of accuracy. The structure of Equation 4.1 depicts that agents should keep a balance between pursuing high accuracy (i.e., the accuracy component q_{d*_i}) and retaining power (i.e., $\mathsf{DB}_i(\mathbf{d})^\alpha$). Agents would lose power when they delegate to inherit high accuracy (Fact 1), especially through long delegation chains (Fact 2 and Fact 3)

When no confusion arises, we simply call a power-sensitive delegation game a delegation game in the rest of this chapter. In the theoretical part of this chapter, we will be working with a restricted version of Equation 4.1 by assuming $\alpha = 1$, and in the empirical section (Section 4.3), we will be working with the general setting $\alpha \in [0, 1]$.

Observe that the strategy profiles of this game are delegation profiles. Each such profile $\mathbf{d}$ induces an LDE $\langle N, \mathbf{w}, \mathbf{d}, \beta \rangle$. Note also that we assume $\mathbf{w}$ to be the standard weight function assigning weight 1 to each agent.[1] The utility of profile $\mathbf{d}$ for i is the accuracy that i acquires in $\mathbf{d}$, multiplied by i's power in $\mathbf{d}$, measured by $\mathsf{DB}_i(\mathbf{d})$.[2] Notice that, therefore, the utility of a dummy agent is 0 and that the utility of a dictator equals her accuracy.

The following simple example illustrates a trade-off between pursuing high inherited individual accuracy and retaining voting power.

---

1 Our results can be extended to general weight profiles $\mathbf{w}$ by computing agents' DBs based on their initial voting weight.

2 Notice that we slightly abuse notation here by using $\mathbf{d}$ directly as input for the index, instead of the corresponding LDE.

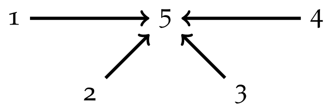
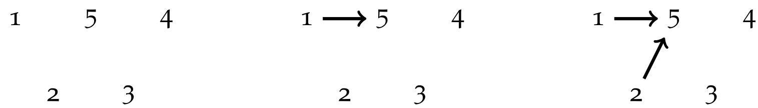


Figure 4.1: Underlying network of Example 16.



(a) $\mathbf{d}_0$ (b) $\mathbf{d}_1$ (c) $\mathbf{d}_2$

Figure 4.2: Delegation profiles of Example 16.

Example 16. *Consider a delegation game in which* $N = \{1, 2, 3, 4, 5\}$ *with accuracy profile* $\mathbf{q} = (0.6, 0.6, 0.8, 0.7, 0.9)$, *quota* $\beta = 3$, *and underlying network as represented in Figure 4.1. Then starting with the trivial delegation profile* $\mathbf{d}_0$ *(Figure 4.2a), we investigate the change of agents* 1 *and* 2*'s utility if they delegate to the more accurate neighbor* 5.

- *Agent* 1. *In profile* $\mathbf{d}_0$, *agent* 1*'s utility equals* $u_1(\mathbf{d}_0) = q_1 \mathsf{DB}_1(\mathbf{d}_0) = 0.6 * \frac{6}{2^4}$. *If agent* 1 *changes to delegate to agent* 5 *(*$\mathbf{d}_1$*, Figure 4.2b), her voting power becomes half of that in* $\mathbf{d}_0$*, i.e.,* $\mathsf{DB}_1(\mathbf{d}_1) = \frac{3}{2^4}$*, but she inherits a higher accuracy from agent* 5*, namely* 0.9*. However, agent* 1 *would not delegate, otherwise she loses too much voting power, which cannot be compensated by the higher accuracy inherited from agent* 5*, i.e.,* $u_1(\mathbf{d}_1) = q_5 \mathsf{DB}_1(\mathbf{d}_1) < u_1(\mathbf{d}_0) = q_1 \mathsf{DB}_1(\mathbf{d}_1)$.
- *Agent* 2. *We then consider the strategy of agent* 2 *based on profile* $\mathbf{d}_1$ *(Figure 4.2b). Since now agent* 1 *delegates to agent* 5*, agent* 2*'s voting power also changes (*$\mathsf{DB}_2(\mathbf{d}_1) = \frac{4}{2^4}$ *comparing to* $\mathsf{DB}_2(\mathbf{d}_0) = \frac{6}{2^4}$*) even though she is still a guru as in* $\mathbf{d}_0$*. Therefore agent* 2 *would choose to delegate to agent* 5 *(to form profile* $\mathbf{d}_2$*, Figure 4.2c), since her voting power will change to* $\mathsf{DB}_2(\mathbf{d}_2) = \frac{3}{2^4}$ *while she also inherits an accuracy of* 0.9 *from agent* 5*. This leads to* $u_2(\mathbf{d}_2) = q_5 \mathsf{DB}_2(\mathbf{d}_2) > u_2(\mathbf{d}_1) = q_2 \mathsf{DB}_2(\mathbf{d}_1)$.

Note that for our experiments later in this chapter, we will be using the more general form of (4.1) given by $\mathsf{DB}_i(\mathbf{d})^\alpha \cdot q_{d_i^*}$, with $\alpha \in [0, 1]$.

4.2 EQUILIBRIUM ANALYSIS

In this section, we investigate the existence of Nash equilibria in delegation games. We first show that Nash equilibria are not guaranteed to exist in general, but they always exist in several subclasses of delegation games.

4.2.1 Equilibria Do Not Exist in General

In this section, we ask the natural question whether the games of Definition 19 have a Nash equilibrium (NE) in pure strategies. In general, the answer to this question is negative:

Theorem 10. *There are power-sensitive delegation games that have no (pure strategy) NE.*

Proof. Consider the delegation game defined as follows. $N = \{1, 2, 3, 4, 5, 6\}$, $q_1 = 0.51, q_2 = 0.7, q_3 = 0.9, q_4 = 0.6, q_5 = 0.7, q_6 = 0.9$, $\beta = 4$, and for the underlying graph $R = \langle N, E \rangle$, $E = \{(1, 3), (2, 3), (4, 6), (5, 6)\}$, which can be represented as in Figure 4.3. Notice that this is a directed graph.
Below, we list all possible delegation profiles based on the underlying network. Then in each possible delegation profile, we show that at least one agent has an incentive to deviate from the delegation profile.

- The trivial profile $\mathbf{d}_0$, in which each agent is a guru, as shown in Figure 4.4.
(1) $\mathbf{d}_0$. Agent 1 deviates from $\mathbf{d}_0$ to $\mathbf{d}_1$, thus from $\mathsf{DB}_1(\mathbf{d}_0) = 0.1875$ to $\mathsf{DB}_1(\mathbf{d}_1) = 0.3125$, and from $u_1(\mathbf{d}_0) = 0.1594$ to $u_1(\mathbf{d}_1) = 0.1688$.

- Profiles with only one delegating agent, as shown in Figure 4.5.
(2) $\mathbf{d}_1$. Agent 4 deviates from $\mathbf{d}_1$ to $\mathbf{d}_7$, thus from $\mathsf{DB}_1(\mathbf{d}_1) = 0.25$ to $\mathsf{DB}_1(\mathbf{d}_1) = 0.1875$, and from $u_1(\mathbf{d}_0) = 0.15$ to $u_1(\mathbf{d}_1) = 0.1688$.
(3) $\mathbf{d}_2$. Agent 1 deviates from $\mathbf{d}_2$ to $\mathbf{d}_5$, thus from $\mathsf{DB}_1(\mathbf{d}_2) = 0.25$ to $\mathsf{DB}_1(\mathbf{d}_5) = 0.1875$, and from $u_1(\mathbf{d}_0) = 0.1275$ to $u_1(\mathbf{d}_1) = 0.1688$.
(4) $\mathbf{d}_3$. Agent 1 deviates from $\mathbf{d}_3$ to $\mathbf{d}_7$, thus from $\mathsf{DB}_1(\mathbf{d}_3) = 0.25$ to $\mathsf{DB}_1(\mathbf{d}_7) = 0.15625$, and from $u_1(\mathbf{d}_3) = 0.1275$ to $u_1(\mathbf{d}_7) = 0.1406$.
(5) $\mathbf{d}_4$. Agent 1 deviates from $\mathbf{d}_4$ to $\mathbf{d}_8$, thus from $\mathsf{DB}_1(\mathbf{d}_4) = 0.25$ to $\mathsf{DB}_1(\mathbf{d}_8) = 0.15625$, and from $u_1(\mathbf{d}_4) = 0.1275$ to $u_1(\mathbf{d}_8) = 0.1406$.

- Profiles with two delegating agents, as shown in Fig 4.6.
(6) $\mathbf{d}_5$. Agent 2 deviates from $\mathbf{d}_5$ to $\mathbf{d}_1$, thus from $\mathsf{DB}_2(\mathbf{d}_5) = 0.1875$ to $\mathsf{DB}_2(\mathbf{d}_1) = 0.25$, and from $u_2(\mathbf{d}_5) = 0.1688$ to $u_2(\mathbf{d}_1) = 0.175$.
(7) $\mathbf{d}_6$. Agent 5 deviates from $\mathbf{d}_6$ to $\mathbf{d}_3$, thus from $\mathsf{DB}_5(\mathbf{d}_6) = 0.1875$ to $\mathsf{DB}_5(\mathbf{d}_3) = 0.25$, and from $u_2(\mathbf{d}_5) = 0.1688$ to $u_2(\mathbf{d}_1) = 0.175$.
(8) $\mathbf{d}_7$. Agent 4 deviates from $\mathbf{d}_7$ to $\mathbf{d}_1$, thus from $\mathsf{DB}_4(\mathbf{d}_7) = 0.15625$ to $\mathsf{DB}_4(\mathbf{d}_1) = 0.25$, and from $u_4(\mathbf{d}_7) = 0.1406$ to $u_4(\mathbf{d}_1) = 0.15$.

Figure 4.3: The underlying Network.

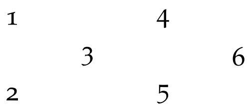
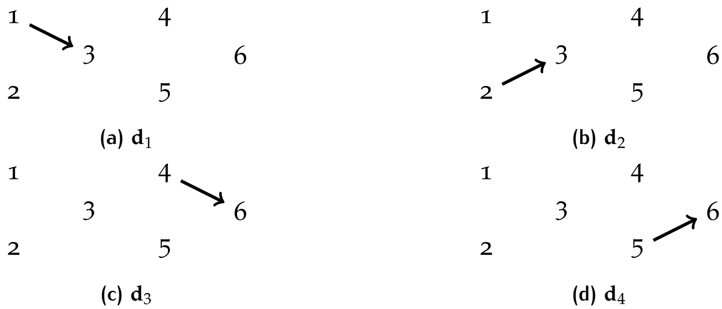


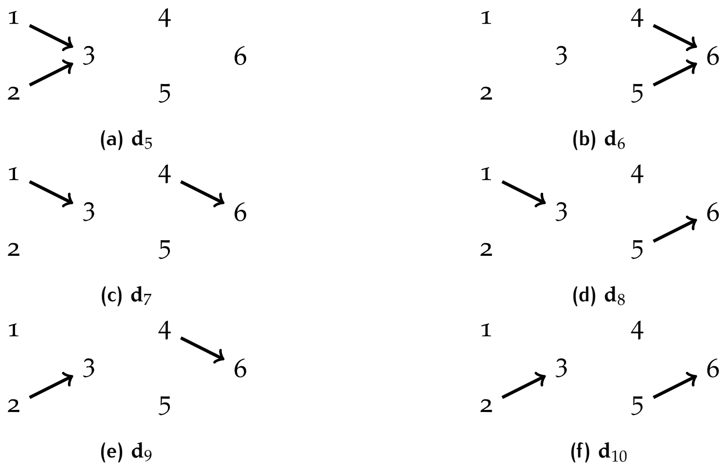
Figure 4.4: The trivial profile $\mathbf{d}_1$.



(a) $\mathbf{d}_1$ (b) $\mathbf{d}_2$ (c) $\mathbf{d}_3$ (d) $\mathbf{d}_4$

Figure 4.5: Profiles with only one delegating agent.

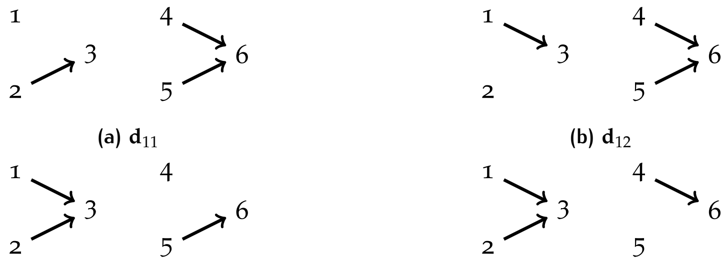
(9) $\mathbf{d}_8$. Agent 5 deviates from $\mathbf{d}_8$ to $\mathbf{d}_1$, thus from $\mathsf{DB}_5(\mathbf{d}_8) = 0.15625$ to $\mathsf{DB}_5(\mathbf{d}_1) = 0.25$, and from $u_5(\mathbf{d}_8) = 0.1406$ to $u_5(\mathbf{d}_1) = 0.175$.
(10) $\mathbf{d}_9$. Agent 4 deviates from $\mathbf{d}_9$ to $\mathbf{d}_2$, thus from $\mathsf{DB}_4(\mathbf{d}_9) = 0.15625$ to $\mathsf{DB}_4(\mathbf{d}_2) = 0.25$, and from $u_4(\mathbf{d}_9) = 0.1406$ to $u_4(\mathbf{d}_2) = 0.15$.
(11) $\mathbf{d}_{10}$. Agent 5 deviates from $\mathbf{d}_{10}$ to $\mathbf{d}_2$, thus from $\mathsf{DB}_5(\mathbf{d}_{10}) = 0.15625$ to $\mathsf{DB}_5(\mathbf{d}_2) = 0.25$, and from $u_5(\mathbf{d}_{10}) = 0.1406$ to $u_5(\mathbf{d}_2) = 0.175$.



(a) $\mathbf{d}_5$ (b) $\mathbf{d}_6$ (c) $\mathbf{d}_7$ (d) $\mathbf{d}_8$ (e) $\mathbf{d}_9$ (f) $\mathbf{d}_{10}$

Figure 4.6: Profiles with two delegating agents.

- Profiles with three delegating agents, as shown in Fig 4.7.
  (12) $\mathbf{d}_{11}$. Agent 2 deviates from $\mathbf{d}_{11}$ to $\mathbf{d}_6$, thus from $\mathsf{DB}_2(\mathbf{d}_{11}) = 0.09375$ to $\mathsf{DB}_2(\mathbf{d}_6) = 0.1875$, and from $u_2(\mathbf{d}_{11}) = 0.0844$ to $u_2(\mathbf{d}_6) = 0.1313$.
  (13) $\mathbf{d}_{12}$. Agent 1 deviates from $\mathbf{d}_{12}$ to $\mathbf{d}_6$, thus from $\mathsf{DB}_1(\mathbf{d}_{12}) = 0.09375$ to $\mathsf{DB}_1(\mathbf{d}_6) = 0.1875$, and from $u_1(\mathbf{d}_{12}) = 0.0844$ to $u_1(\mathbf{d}_6) = 0.0956$.
  (14) $\mathbf{d}_{13}$. Agent 5 deviates from $\mathbf{d}_{13}$ to $\mathbf{d}_5$, thus from $\mathsf{DB}_5(\mathbf{d}_{13}) = 0.09375$ to $\mathsf{DB}_5(\mathbf{d}_5) = 0.1875$, and from $u_5(\mathbf{d}_{13}) = 0.0844$ to $u_5(\mathbf{d}_5) = 0.1313$.
  (15) $\mathbf{d}_{14}$. Agent 4 deviates from $\mathbf{d}_{14}$ to $\mathbf{d}_5$, thus from $\mathsf{DB}_4(\mathbf{d}_{14}) = 0.09375$ to $\mathsf{DB}_4(\mathbf{d}_5) = 0.1875$, and from $u_4(\mathbf{d}_{14}) = 0.0844$ to $u_4(\mathbf{d}_5) = 0.1125$.



(a) $\mathbf{d}_{11}$ (b) $\mathbf{d}_{12}$ (c) $\mathbf{d}_{13}$ (d) $\mathbf{d}_{14}$

Figure 4.7: Profiles with three delegating agents.

- Due to the restriction by the underlying graph, agent 3 and agent 6 can only be gurus. Therefore in the last possible profile $\mathbf{d}_{15}$, agents 1, 2, 4 and 5 are all delegating agents (Fig 4.8).
  (16) $\mathbf{d}_{15}$. Agent 5 deviates from $\mathbf{d}_{15}$ to $\mathbf{d}_{14}$, thus from $\mathsf{DB}_5(\mathbf{d}_{15}) = 0.09375$ to $\mathsf{DB}_5(\mathbf{d}_{14}) = 0.15625$, and from $u_5(\mathbf{d}_{15}) = 0.0844$ to $u_5(\mathbf{d}_{14}) = 0.1094$.

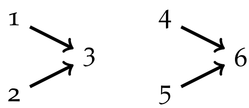


Figure 4.8: Profile $\mathbf{d}_{15}$.

Therefore, there is no NE in this delegation game as all profiles admit a profitable unilateral deviation. □

Although a pure strategy NE do not exist in general, we can still prove its existence in some restricted classes of delegation games. In particular, we study games with minority quota, unanimous quota, and complete networks.

4.2.2 Existence of Equilibria: Quota Conditions

Since we assumed that a quota rule is used in delegation games (and LDEs), it should be observed that the specific quota value β can also influence agents' behaviors in the game. Note that in the following theorem, we consider delegation games with a small quota, even though we specify that $\beta > |N|/2$ in Section 3.1.1. When β is small, specifically, smaller than $|N|/2$, pure strategy NE can be guaranteed.

Theorem 11. *In any power-sensitive delegation game $\mathcal{D} = \langle N, R, \mathbf{q}, \Sigma, \beta, u \rangle$, where $\beta \leq \lceil \frac{n}{2} \rceil$ ($n = |N|$), there always exists at least one (pure strategy) NE.*

Proof. We show that if the above condition is satisfied, the trivial profile $\mathbf{d}$, i.e., $\forall j \in N, \mathbf{d}_j = j$, is a NE. Towards a contradiction, assume that an agent $i \in N$ exists, such that i has an incentive to deviate from $\mathbf{d}$. That is, another agent $i' \in N$ exists, such that i obtains higher utility if she delegates to i', formally, $u_i(\mathbf{d}') > u_i(\mathbf{d})$, where $\mathbf{d}' = (\mathbf{d}_{-i}, \mathbf{d}'(i) = i')$. In profile $\mathbf{d}$, we obtain that $\mathsf{DB}_i(\mathbf{d}) = \binom{n-1}{\beta-1}/2^{n-1}$ since all agents are gurus and i is a swing agent for any subset of size $\beta - 1$ in $N \setminus \{i\}$. Therefore $u_i(\mathbf{d}) = q_i \mathsf{DB}_i(\mathbf{d}) = q_i \binom{n-1}{\beta-1}/2^{n-1}$. On the other hand, in profile $\mathbf{d}'$, since $\mathbf{d}'(i) = i'$, i can be a swing agent in a coalition only if i' is also contained in the coalition. Thus $\mathsf{DB}_i(\mathbf{d}') = \binom{n-2}{\beta-2}/2^{n-1}$, by which $u_i(\mathbf{d}') = q_{i'} * \mathsf{DB}_i(\mathbf{d}') = q_{i'} \binom{n-2}{\beta-2}/2^{n-1}$ follows. By the assumption,

$$q_{i'} \binom{n-2}{\beta-2} / 2^{n-1} > q_i \binom{n-1}{\beta-1} / 2^{n-1}. \tag{4.2}$$

Rewrite Equation 4.2 and obtain

$$\frac{\beta - 1}{n - 1} > \frac{q_i}{q_{i'}}.$$

Since for any $j \in N$, $q_j \in (0.5, 1]$, we have

$$\frac{\beta - 1}{n - 1} > \frac{1}{2},$$

which contradicts $\beta \leq \lceil \frac{n}{2} \rceil$. □

Intuitively, when the quota β is less than half of the entire weight, based on the trivial profile, an agent will lose more than half of her voting power if she delegates. This loss of voting power cannot be compensated by the improvement of her individual accuracy, since her inherited individual accuracy is always less than two times her original accuracy, as each agent's accuracy is in the range $(0.5, 1]$.

Under the unanimous quota, i.e., $\beta > n-1$ in Equation 3.1, by Fact 5, the delegative Banzhaf index of each agent is identical in any profile. Therefore agents' utility depends only on their gurus' accuracy. This guarantees the existence of a pure strategy NE.

Theorem 12. *In any delegation game $\mathcal{D} = \langle N, R, \mathbf{q}, \Sigma, \beta, u\rangle$, where $n-1 < \beta \leq n$, there always exists at least one pure strategy NE.*

Proof. We prove the claim by construction. Given an arbitrary delegation game $\mathcal{D}$, such that $\beta \in (n-1, n]$, we construct a delegation profile $\mathbf{d}$, and prove that this profile is a pure strategy NE for $\mathcal{D}$.

We construct profile $\mathbf{d}$ by Algorithm 2. In the algorithm, we use a sequence σ which is a permutation of N, such that for the i-th and j-th elements, namely $\sigma(i)$ and $\sigma(j)$, $q_{\sigma(i)} \geq q_{\sigma(j)}$ if $i < j$. For any $C \subseteq N$, σ_C is the sequence only containing agents in C, with the consistent pairwise order according to σ. In a social network $R = \langle N, E\rangle$, for any agent $i \in N$, let $R^-(i)$ denote all agents who are able to delegate to i, i.e., $R^-(i) = \{j \in N \mid (j, i) \in E\}$.

Algorithm 2 Building a Nash Equilibrium Under Unanimity Quota

INITIALIZATION: $\mathbf{d} \leftarrow \hat{\mathbf{d}}$, $\theta \leftarrow N$, σ, $Ne \leftarrow []$.

ITERATION:

1: **while** $\theta \neq \emptyset$ **do**
2: $\quad Ne.append(\sigma_\theta(1))$.
3: $\quad$ **while** $Ne \neq []$ **do**
4: $\quad\quad$ **for** all $i \in R^-(Ne_1)$ **do**
5: $\quad\quad\quad$ **if** $d_i = i$ **then**
6: $\quad\quad\quad\quad d_i = Ne_1$.
7: $\quad\quad\quad\quad Ne.append(i)$.
8: $\quad\quad\quad\quad \theta \leftarrow \theta \setminus \{i\}$.
9: $\quad\quad\quad$ **end if**
10: $\quad\quad$ **end for**
11: $\quad\quad Ne.pop(1)$.
12: $\quad$ **end while**
13: **end while**

RETURN: $\mathbf{d}$

Now we show that the output profile $\mathbf{d}$ by Algorithm 2 is a Nash Equilibrium. By the algorithm, initially, for the highest-accuracy agent (i.e., $\sigma_\theta(1)$ and $\theta = N$), all of her neighbors are designated to delegate to her. Then, all agents having access to $\sigma_\theta(1)$ are added to Ne and are linked with delegation chains pointing to $\sigma_\theta(1)$, by lines 3-12 in the iteration component of Algorithm 2. Such a process is applied to all connected components of the underlying network R, so that each agent in N delegates to a guru who has the highest accuracy among those to which the agent has access in the social network. Then by Fact 5, in every delegation profile, each agent's DB is identical when $\beta > n-1$. Therefore in $\mathbf{d}$, each agent obtains the highest possible utility, from which it follows that $\mathbf{d}$ is a NE. □

Intuitively, when β is a unanimous quota, each agent can only be a swing agent for the entire set N, that is, each agent has identical DB. Therefore, the agents' utility solely depends on their gurus' accuracy. By [11], such a game always admits a NE, in which each agent delegates to a guru with the highest individual accuracy to which she has access within the underlying network.

4.2.3 Existence of Equilibria: Complete Networks

The following result shows that a pure strategy NE can also be guaranteed to exist when the underlying network is complete.

Theorem 13. *In any power-sensitive delegation game $\mathcal{D} = \langle N, R, \mathbf{q}, \Sigma, \beta, u \rangle$ where R is a complete network, there exists at least one (pure strategy) NE.*

Proof. Note that in the complete network R, any agent can observe and interact with any other agent. Hence by Fact 3, no delegation chain is longer than 1. We prove the theorem by construction, that is, we use Algorithm 3 to output a profile and verify that the profile is a NE.

To introduce Algorithm 3, we first introduce a sequence σ over $N \setminus \{i^*\}$, where i^* is the agent with the highest accuracy (ties are broken lexicographically). The sequence is a bijection $\sigma : N \setminus \{i^*\} \longleftrightarrow [n-1]$ ($[k] = \{1, \ldots, k\}$ for any $k \in \mathbb{N}$), and for any $i \in N \setminus \{i^*\}$, $\sigma(i)$ denotes the ranking of i in σ, and σ_k ($k \in [n-1]$) denotes the k-th agent in σ. For example, if agent i is the k-th agent in sequence σ, $\sigma_k = i$ and $\sigma(i) = k$. Furthermore, for any coalition $C \subseteq N$, let σ^C denote the sequence that is consistent with σ but restricted to agents in C.

In other words, in Algorithm 3, in turns according to σ, each agent in $N \setminus \{i^*\}$ chooses between being a guru or delegating to i^*. If an agent changes from being a guru to delegating to i^* (to obtain higher utility), she cannot change her strategy anymore. When no agent wants to change, the algorithm terminates and returns the profile $\mathbf{d}$.

Next, we verify that $\mathbf{d}$ is a NE. We show that in $\mathbf{d}$, (*Claim* 1) i^* will not deviate, (*Claim* 2) any delegator will not deviate, and (*Claim* 3) any guru, other than i^*, will not deviate.

First, *Claim* 1 obviously holds since (i) i^* will not change to delegate to any delegator to form a delegating cycle otherwise she would get utility of 0 by Equation 4.1; (ii) by Fact 1, i^* will not delegate to any other guru, otherwise she would obtain a lower DB and inherits a lower accuracy.

Algorithm 3 Constructing an NE in Complete Networks

INITIALIZATION: i^*, $C^0 = \emptyset$, $C^1 = N\backslash\{i^*\}$, σ, $j = 1$, $\mathbf{d}$: for any $i \in N$, $\mathbf{d}(i) = i$.

BEST RESPONSE SEQUENCE:

while $C^j \neq C^{j-1}$ **do**
    $j \leftarrow j + 1$
    $C^j \leftarrow C^{j-1}$
    for $k = 1$ to $|C^{j-1}|$ **do**
        $\mathbf{d}(i) \leftarrow arg\max_{a \in \{i^*, \sigma_k^{C^{j-1}}\}} u_i((\mathbf{d}_{-\sigma_k^{C^{j-1}}}, d'_{\sigma_k^{C^{j-1}}} = a))$
        if $\mathbf{d}(\sigma_k^{C^{j-1}}) = i^*$ **then**
            $C^j \leftarrow C^j \backslash \{\sigma_k^{C^{j-1}}\}$
        end if
    end for
end while

RETURN: $\mathbf{d}$

Next we show *Claim* 2. It is clear that a delegator would not change to delegate to another delegator by Fact 3. Then we show that any delegator will not deviate to be a guru. We use Lemma 3, which illustrates that if more agents delegate to i^*, then all current delegators' DB would not change, and Lemma 4, which illustrates that if more agents delegate to i^*, then all remaining gurus' (except for i^*) DB will be weakly worse off.

Lemma 3. *Given a power-sensitive delegation game* $\mathcal{D} = \langle N, R, \mathbf{q}, \Sigma, \beta, u\rangle$ *and a profile* $\mathbf{d}$*, where for any* $j \in N$*, if* $\mathbf{d}(j) \neq j$*,* $\mathbf{d}(j) = i^*$*, we construct another profile* $\mathbf{d}' = (\mathbf{d}_{-i}, \mathbf{d}'(i) = i^*)$*, where* $\mathbf{d}(i) = i$ *and* $i \neq i^*$*. Then we have for all* $j \in N$ *such that* $\mathbf{d}(j) = i^*$ *that* $\mathsf{DB}_j(\mathbf{d}) = \mathsf{DB}_j(\mathbf{d}')$*.*

Proof. We show that for any delegator j under $\mathbf{d}$, she is a swing agent for a coalition $C \subseteq N$ under $\mathbf{d}$ if and only if she is a swing agent for C under $\mathbf{d}'$. Since the only difference between $\mathbf{d}$ and $\mathbf{d}'$ is the strategy of i, and i is a guru under $\mathbf{d}$ and $\mathbf{d}'(i) = i^*$, we have for any coalition $C \subseteq N$ such that $i^* \in C$ that the sum of initial endowed weight of all agents in $De^*(C, \mathbf{d})$ and $De^*(C, \mathbf{d}')$ is identical, i.e., $\sum_{a \in De^*(C,\mathbf{d})} w(a) = \sum_{a \in De^*(C,\mathbf{d}')} w(a)$. Note that j can be a swing agent only if she is contained in $De^*(C, \mathbf{d})$. Since $\mathbf{d}(j) = \mathbf{d}'(j) = i^*$, we have that if j is a swing agent for C, then $i^* \in C$ under both $\mathbf{d}$ and $\mathbf{d}'$. Therefore, we have that under $\mathbf{d}$, $\sum_{a \in De^*(C,\mathbf{d})} w(a) \geq \beta$ and $\sum_{a \in De^*(C\backslash\{j\},\mathbf{d})} w(a) < \beta$ if and only if $\sum_{a \in De^*(C,\mathbf{d}')} w(a) \geq \beta$ and $\sum_{a \in De^*(C\backslash\{j\},\mathbf{d}')} w(a) < \beta$. Thus, $\mathsf{DB}_j(\mathbf{d}) = \mathsf{DB}_j(\mathbf{d}')$. □

Lemma 4. *Given a delegation game $\mathcal{D} = \langle N, R, \mathbf{q}, \Sigma, \beta, u \rangle$ and a profile $\mathbf{d}$, where for any $j \in N \setminus Gu(\mathbf{d})$ with $\mathbf{d}(j) = i^*$, we construct another profile $\mathbf{d}' = (\mathbf{d}_{-i}, \mathbf{d}'(i) = i^*)$, where $i \in Gu(\mathbf{d}) \setminus \{i^*\}$. Then we have for all $j \in N \setminus Gu(\mathbf{d}')$ that $\mathsf{DB}_j(\mathbf{d}) \geq \mathsf{DB}_j(\mathbf{d}')$.*

Proof. We compare the times of any $j \in N \setminus Gu(\mathbf{d}')$ serving as a swing agent under $\mathbf{d}$ and $\mathbf{d}'$. Since the only difference between $\mathbf{d}$ and $\mathbf{d}'$ is the strategy of agent i, it is sufficient to consider coalitions containing i. Then, under $\mathbf{d}$ and $\mathbf{d}'$ respectively, we count the number of coalitions for which j is a swing agent. Consider the two possible cases: (1) $i^* \in C$ and (2) $i^* \notin C$.

1. $i^* \in C$. Since i is a guru under $\mathbf{d}$ and $\mathbf{d}'(i) = i^*$, we have $|De^*(C, \mathbf{d})| = |De^*(C, \mathbf{d}')|$. Therefore, $|De^*(C, \mathbf{d})| \geq \beta$ and $|De^*(C \setminus \{j\}, \mathbf{d})| < \beta$ if and only if $|De^*(C, \mathbf{d}')| \geq \beta$ and $|De^*(C \setminus \{j\}, \mathbf{d}')| < \beta$. That is, j serves as a swing agent for C under $\mathbf{d}$ if and only if j is a swing agent for C under $\mathbf{d}'$.
2. $i^* \notin C$. Since $i^* \notin C$ and $\mathbf{d}'(i) = i^*$, we have $|De^*(C, \mathbf{d})| = |De^*(C, \mathbf{d}')| + 1$, and j cannot be a swing agent for C under both profiles. Then we consider two possible (exhaustive) sub-cases:
   a) j is a swing agent for C under $\mathbf{d}$, but becomes a non-swing agent for C under $\mathbf{d}'$. That is, $|De^*(C, \mathbf{d})| = \lceil \beta \rceil$ and $|De^*(C \setminus \{j\}, \mathbf{d})| = |De^*(C, \mathbf{d}')| = \lceil \beta \rceil - 1$. Since $i^* \notin C$, none of delegators is contained in $De^*(C, \mathbf{d})$ or $De^*(C, \mathbf{d}')$. Then let C^* denote the set of gurus, except for i^*, under $\mathbf{d}$, i.e., $C^* = Gu(\mathbf{d}) \setminus \{i^*\}$, and $n^* = |C^*|$. Therefore, $|De^*(C, \mathbf{d})| = |De^*(C \cap C^*, \mathbf{d})|$. Thus in this sub-case, the number of coalitions C for which j is a swing agent is $\binom{n^*-2}{\lceil \beta \rceil - 2}$, i.e., C contains $\lceil \beta \rceil - 2$ agents in $Gu(\mathbf{d}) \setminus \{i, j, i^*\}$, and C also contains i, j.
   b) j is a swing agent for C under $\mathbf{d}'$, but is a non-swing agent for C under $\mathbf{d}$. That is $|De^*(C, \mathbf{d})| = \lceil \beta \rceil + 1$ and $|De^*(C, \mathbf{d}')| = |De^*(C, \mathbf{d})| - 1 = \lceil \beta \rceil$. Then, the number of C, for which j is a swing agent under $\mathbf{d}'$, is $\binom{n^*-2}{\lceil \beta \rceil - 1}$, i.e., C contains $\lceil \beta \rceil - 1$ agents in $Gu(\mathbf{d}) \setminus \{i, j, i^*\}$, and C also contains i, j.

   By Theorem 11, if $\beta < \lceil \frac{n}{2} \rceil$, Algorithm 3 returns the trivial profile since remaining a guru maximizes each agent's utility, and the trivial profile is a NE. Then $\beta \geq \frac{n+1}{2}$, and we have $\binom{n^*-2}{\lceil \beta \rceil - 2} \geq \binom{n^*-2}{\lceil \beta \rceil - 1}$ since $n^* \leq n - 1$. Therefore, the number of times that j serves as a swing agent under $\mathbf{d}$ is weakly more than that under $\mathbf{d}'$.

□

Therefore, in Algorithm 3, if an agent i chooses to delegate to i^*, she has no incentive to change back to be a guru under $\mathbf{d}$ since: (1) by Lemma 3, as more agents delegate to i^*, i's utility does not change since DB and q_{i^*} do not change;

(2) by Lemma 4, her utility becomes even lower than that before she chooses to delegate to i^*.

Next, we show that any delegator will not change to delegate to another guru, by using the following lemma.

Lemma 5. *Given a power-sensitive delegation game $\mathcal{D}$ and a profile* $\mathbf{d}$, *where for any* $j \in N \backslash Gu(\mathbf{d})$, $\mathbf{d}(j) = i^*$, *we construct another profile* $\mathbf{d}' = (\mathbf{d}_{-i}, \mathbf{d}'(i) = i')$, *where* $\mathbf{d}(i) = i^*$ *and* $i' \in Gu(\mathbf{d}) \backslash \{i^*\}$. *Then* $\mathsf{DB}_i(\mathbf{d}) \geq \mathsf{DB}_i(\mathbf{d}')$.

Proof. Analogously to Lemma 4, we also prove the lemma by comparing the number of times that i serves as a swing agent under $\mathbf{d}$ and $\mathbf{d}'$, respectively. First notice that for any coalition C such that $i^*, i' \in C$, $|De^*(C, \mathbf{d})| = |De^*(C, \mathbf{d}')|$. That is, i is a swing agent for C under $\mathbf{d}$ if and only if she is also a swing agent for C under $\mathbf{d}'$.
Then we consider any coalition C which contains only one of i^* and i'. Let $C' = N \backslash (De^*(\{i^*\}, \mathbf{d}) \cup \{i'\})$, namely, all agents except for delegators under $\mathbf{d}$, i^* and i', and let $n' = |C'|$ and $n^* = |De^*(\{i^*\}, \mathbf{d}) \backslash \{i^*\}|$. Then we consider two cases:
(1) i is swing for C under $\mathbf{d}$, but not swing under $\mathbf{d}'$. We can infer that $i^* \in C$ but $i' \notin C$, since $\mathbf{d}(i) = i^*$ and $i \in De^*(C, \mathbf{d})$. Therefore, $|De^*(C, \mathbf{d})| = \lceil \beta \rceil$ and the number of such coalitions (or times i is swing in this case) is $\binom{n'+n^*}{\lceil \beta \rceil - 2}$, i.e., C contains $\lceil \beta \rceil - 2$ agents in $C' \cup De^*(\{i^*\}, \mathbf{d}) - \{i^*\}$, and it also contains i, i^*. (2) i is not swing for C under $\mathbf{d}$, but is swing under $\mathbf{d}'$. Then we have that $i' \in C$, but $i^* \notin C$. Therefore, $|De^*(C, \mathbf{d}')| = \lceil \beta \rceil$, and the number of such coalitions (or the times of i being a swing agent in this case) is $\binom{n'}{\lceil \beta \rceil - 2} * 2^{n^*}$. That is, C contains $\lceil \beta \rceil - 2$ agents in C' as well as i, i', and since i^* is not in C, any agent delegating to i^* is dummy and does not influence the value of $|De^*(C, \mathbf{d})|$, thus it leads to 2^{n^*} times of $\binom{n'}{\lceil \beta \rceil - 2}$.
If $\beta < \lceil \frac{n}{2} \rceil$, the trivial profile is a NE by Theorem 11, and thus we assume $\beta \geq \frac{n+1}{2}$. Under this condition, we have $\binom{n'+n^*}{\lceil \beta \rceil - 2} \geq \binom{n'}{\lceil \beta \rceil - 2} * 2^{n^*}$ by Lemma 6, thus $\mathsf{DB}_i(\mathbf{d}) \geq \mathsf{DB}_i(\mathbf{d}')$. □

Lemma 6. *Given* $n, n', n^* \in \mathbb{Z}_+$ *and* $\beta \in (n/2, n]$ *such that* $n = n^* + n' + 2$, *we have* $\binom{n'+n^*}{\lceil \beta \rceil - 2} \geq \binom{n'}{\lceil \beta \rceil - 2} \cdot 2^{n^*}$.

Proof. Let $\alpha = \frac{(\lceil\beta\rceil-2)!(n'-\lceil\beta\rceil+2)!(n^*+n'-\lceil\beta\rceil+2)!}{n'!}$. We assume that $n' - \lceil\beta\rceil + 2 \geq 0$, otherwise $\binom{n'}{\lceil\beta\rceil-2} = 0$ and $\binom{n'+n^*}{\lceil\beta\rceil-2} \geq \binom{n'}{\lceil\beta\rceil-2} \cdot 2^{n^*}$ obviously holds. Then

$$\begin{aligned}
&\binom{n'}{\lceil\beta\rceil-2} \cdot 2^{n^*} \cdot \alpha \\
=&\frac{n'!}{(\lceil\beta\rceil-2)!(n'-\lceil\beta\rceil+2)!} \cdot 2^{n^*} \cdot \alpha \\
=&2^{n^*} \cdot \underbrace{(n'-\lceil\beta\rceil+3)(n'-\lceil\beta\rceil+4)\ldots(n'-\lceil\beta\rceil+2+n^*)}_{n^*} \\
=&\underbrace{2\cdot(n'-\lceil\beta\rceil+3)2\cdot(n'-\lceil\beta\rceil+4)\ldots 2\cdot(n'-\lceil\beta\rceil+2+n^*)}_{n^*}.
\end{aligned} \tag{4.3}$$

We also have that

$$\begin{aligned}
\binom{n'+n^*}{\lceil\beta\rceil-2} \cdot \alpha &= \frac{(n'+n^*)!}{(\lceil\beta\rceil-2)!(n'+n^*-\lceil\beta\rceil+2)!} \cdot \alpha \\
&= \underbrace{(n'+1)(n'+2)\ldots(n'+n^*)}_{n^*}.
\end{aligned} \tag{4.4}$$

We first compare $2\cdot(n'-\lceil\beta\rceil+2+n^*)$ and $n'+n^*$. Since $\lceil\beta\rceil \geq \frac{n+1}{2}$, $2\lceil\beta\rceil \geq n'+n^*+4$, we have

$$2n' + 2n^* - 2\lceil\beta\rceil + 4 \leq n' + n^*.$$

Therefore, $\binom{n'}{\lceil\beta\rceil-2} \cdot 2^{n^*} \cdot \alpha \leq \binom{n'+n^*}{\lceil\beta\rceil-2} \cdot \alpha.$ □

Finally, we show *Claim* 3 that any guru i ($i \neq i^*$) under **d** will not deviate. It is obvious that i has no incentive to change to delegate to i^* by the operation of Algorithm 3. That is, in line 5 of Algorithm 3, agent i's optimal delegation strategy is being a single guru. Then by Lemma 5, i can obtain even lower utility if she changes to delegate to another guru rather than i^*. Hence i will not deviate from **d**. □

It remains an open question whether the completeness of the underlying network could be replaced by weaker properties, e.g., symmetry.

4.3 EMPIRICAL STUDY

In this section, we empirically study agents' delegation behavior in power-sensitive delegation games. We evaluate the performance of two types of iteration by computer simulation. Recall that we theoretically showed that a pure strategy

NE cannot always be guaranteed to exist in delegation games. Therefore, as an extension of the theoretical analysis, we use an algorithm implementing a process of iterated better response dynamics as a process of equilibrium computation, for power-sensitive delegation games (Algorithm 4). Notice that in Algorithm 4, a sequence σ, i.e., a permutation of N, is also used, where $\sigma(i)$ denotes the i-th agent. Algorithm 4 is also compared with another algorithm, modeling *one shot iteration* (OSI). In OSI, starting from the trivial profile, each agent chooses once her best response strategy (see Definition 5) simultaneously, which maximizes her utility. Formally, let $\mathbf{d}^0$ be the trivial profile. In OSI, i chooses $d_i = argmax_{a \in R(i)} u_i((\mathbf{d}^0_{-i}, d_i = a))$.

Algorithm 4 Iterated Better Response Dynamincs (IBRD)

INITIALIZATION: $\mathbf{d}^0 : \forall i \in N, \mathbf{d}^0(i) = i, \sigma, j = 1, k = 0.$

ROUND j:

1. Let $i = 1$.
2. step 1: Randomly choose ℓ from $R(\sigma(i))$, and let $\tilde{\mathbf{d}}^j = (\mathbf{d}^j_{-\sigma(i)}, \tilde{\mathbf{d}}^j(\sigma(i)) = \ell)$.
3. step 2: If $u_{\sigma(i)}(\tilde{\mathbf{d}}^j) > u_{\sigma(i)}(\mathbf{d}^j)$, $\mathbf{d}^j \leftarrow \tilde{\mathbf{d}}^j$ and go to step 3,
   else if $u_{\sigma(i)}(\tilde{\mathbf{d}}^j) \leq u_{\sigma(i)}(\mathbf{d}^j)$ and $k > 2(|E(\sigma(i))| + 1)$, go to step 3,
   otherwise go to step 1 and $k = k + 1$.
4. step 3: If $i < n$, $i = i + 1$ and go to step 1;
   else if $i = 30$ and $\mathbf{d}^j = \mathbf{d}^{j-1}$, return $\mathbf{d}^j$;
   otherwise go to Round $j + 1$.

To evaluate the performance of the two algorithms, we measure several criteria of the resulting profiles. First, to measure structural features of a given profile, we compute the ratio of delegators, and the maximum and average length of all delegation chains. Since the DB is supposed to be related to the structure of delegation profiles, we also compute the maximum/minimum/average DB as well as the Gini coefficient [36], which reflects the balance of the distribution of DB: higher Gini coefficient implies more inequality. Moreover, the weighted average accuracy of all gurus in each profile is also computed as another measurement of performance of the algorithms. Finally, we investigate the ratio of converged instances of the IBRD for some parameters: when the converged instances take a low ratio, agents tend to change their delegations easily.

These criteria, namely, the ratio of delegators, the Gini coefficient of DB's and the average accuracy, reflect the information with which we are mostly concerned, i.e., the structure of the delegation graphs, the distribution of the DB's, and the individual decision quality. Therefore, we provide the statistical tests (ANOVA tests) to determine whether the qualifications of the described trends with respect to the variable on the x-axis (e.g., "increasing" and "decreasing") as well as the described difference between the algorithms are appropriate or not. As shown in Appendix A.1, the details of the ANOVA tests for the above three criteria (i.e., corresponding to Figures 4.9a, 4.9g, 4.9h, 4.10a, 4.10g, 4.10h, 4.11a, 4.10g and 4.11h) are provided.

4.3.1 Experimental Setup

We compare the two algorithms in three simulations, all of which use random networks (recall Section 2.1) as underlying networks.

As one might expect, the bottleneck in our experiments consists in the computation of DB in order to establish agents' utilities by Equation 4.1. It is well-known that computing the Banzhaf index in weighted voting games is intractable [55] in general. Matsui & Matsui [54] provided a pseudo-polynomial time algorithm to compute the Banzhaf index in weighted voting when agents' weights are small. The authors use an inductive method to compute the Banhzaf index fast. In each step of the induction, the algorithm requires that after removing a given agent from a coalition, the weight of the residual coalition should be computable based only on the individual weights of the agents in the residual coalition. Unfortunately, this method cannot be adapted to DB, since by removing a given agent from a coalition, the residual weight also depends on the structure of the delegation graph and the identity of the removed agent. We therefore implement the approximation method described by Bachrach *et al.* [3]. Each time we need to establish the DB of an agent, 15000 coalitions are randomly sampled (by uniform distribution), and the ratio of the coalitions for which the agent is swing is used as the estimator of the DB. By the analytical bounds proven by Bachrach *et al.* [3] with the above method, we know that the correct DB lies in the confidence interval $[\widehat{\mathsf{DB}} - 0.011, \widehat{\mathsf{DB}} + 0.011]$ with probability of 0.95, where $\widehat{\mathsf{DB}}$ is the estimator. So it should be clear that the statistics presented in this section report on values that depend on the estimator $\widehat{\mathsf{DB}}$, and that with high probability are close to the exact intended values.

The simulation is programmed in Python 3.7, on the Peregrine HPC cluster of the University of Groningen.[3] The source code can be found on `https://github.com/IamYoezy/Power-in-Liquid-Democracy`.

4.3.2 Parameter Setting

We set $|N| = 30$, and for each parameter setting, we use an accuracy vector $Q \in \mathbb{R}^{30}$, where each element in Q is drawn from a Gaussian distribution $\mathcal{N}(0.75, 0.125)$ and forced to be in range $[0.5, 1]$. All statistics are the mean value

3 `https://wiki.hpc.rug.nl/peregrine/start`

over 50 instances for each parameter setting, but note that for IBRD (Algorithm 4), we only take the instances converged in 100 iterations among all 50 instances.

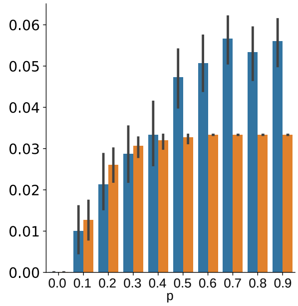
We will be working with three parameters. To test the effect of connectivity on power we assume that interaction happens on a random network and vary the probability p (see range in e.g., Figure. 4.9b) of any two agents being linked, with assumption of $\alpha = 1$ and $\beta = 16$ (i.e., the weighted majority voting rule), i.e., experiment A. To test the effect of different attitudes towards the importance of power for agents we work in experiment B with the general utility function defined by Equation 4.1 with $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ and assuming an underlying random network with $p = 0.75$ and $\beta = 16$. Finally to test the effect of the quota β we vary the value of β in range $\{16, 18, 21, 24, 27\}$ in experiment C, i.e., from weighted majority to almost unanimous voting, while keeping $\alpha = 1$ and connectivity parameter $p = 0.75$.

4.3.3 Connectivity: Experiment A

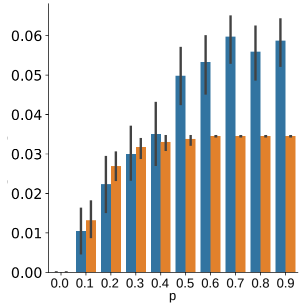
Figure 4.9a shows that the higher the connectivity (the larger the p), the weakly more agents tend to delegate both in equilibrium (IBRD) and in one-shot interaction (OSI). This is in line with expectations because agents have more chance to interact with high-accuracy agents. It is worth observing, however, that the ratio of delegators is very low (less than 0.06 on average). That is, very few agents delegate on average. This is in contrast with the behavior of the delegation game where utility is solely based on accuracy (cf. Bloembergen *et al.* [11]). We will see in experiment B that the influence of power on agents' utility seems to be an important factor in limiting vs. facilitating delegations.

The delegation structure indicators, i.e., the average length of delegation chains (Figure 4.9b) and the longest delegation chain (Figure 4.9c) further show more details of agents' behavior. By Figure 4.9c, all delegations are one-hop in all instances. This indicates that delegation through long chains is significantly prevented in order to avoid loss of voting power, since agents' utility strongly depends on DB (e.g., $\alpha = 1$ in the setting of Experiment A). Hence the higher ratio of delegators is the only reason leading to the increase of the average length of delegation chains (Figure 4.9b). When $p \geq 0.4$, the delegator ratio in OSI remains stable with value of approximately 0.33, but the ratio in IRBD is increasingly higher. By the fact that no agent gets access to high-accuracy agents by longer delegation chains in IRBD (all delegations are one-hop), we conclude that agents' delegations can motivate other agents to delegate by changing their voting power.

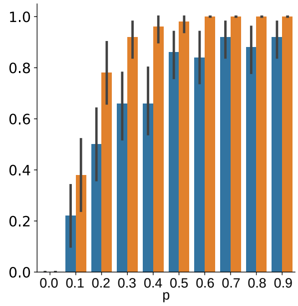
Despite the small number of delegations, we can still observe that increasing p lowers the mean value of DB (Figure. 4.9f) and weakly increases inequality in the distribution of DB, measured by the Gini coefficient (Figure 4.9g). It should be stressed, however, that the Gini coefficient remains very low due to the small fraction of delegators. Intuitively, an increase in delegations enhances some



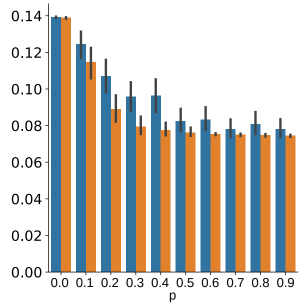
(a) A: ratio of delegators



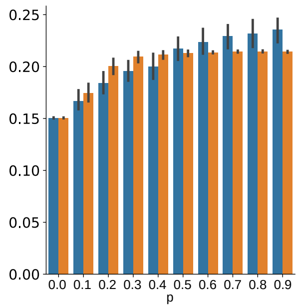
(b) A: Average delegation chain length



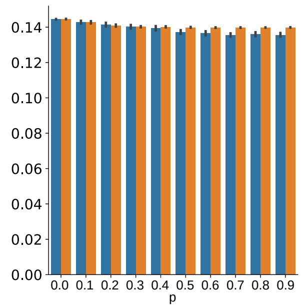
(c) A: The longest delegation chain



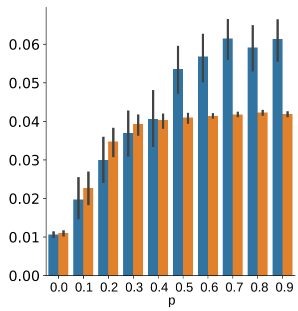
(d) A: Minimum DB



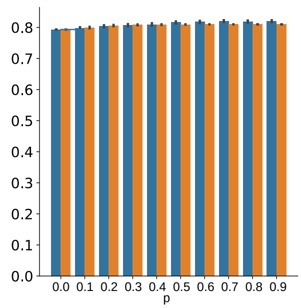
(e) A: Maximum DB



(f) A: Average DB



(g) A: Gini coefficient



(h) A: Average accuracy

Figure 4.9: Experiment A: (orange) for OSI and (blue) for IBRD.

agents' power (Figure 4.9e), but reduces the power of other agents (Figure 4.9d), be they gurus or delegators.

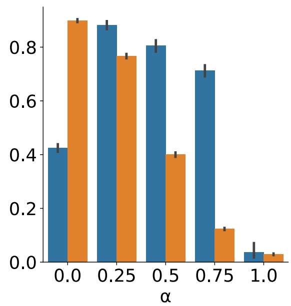
Additionally, by Figure 4.9h, as more delegations happen in the network, the average accuracy becomes weakly higher, since delegators inherit high accuracy from their gurus. However, the difference of the average accuracies is not obvious because of the low delegator ratio, and the average accuracy of IBRD is slightly higher than that of OSI, since more agents delegate in IBRD. This trend coincides with the one observed in [11], where the average accuracy becomes higher as more agents delegate to agents with higher accuracies.

4.3.4 Power: Experiment B

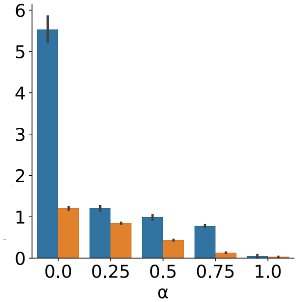
Figure 4.10a shows that larger values of α correspond to significantly fewer delegators for OSI. As agents put more weight on power, they are more reluctant to delegate in the initial profile (recall Fact 1). For IRBD, this effect is observable only for $\alpha \geq 0.25$. We argue that this may depend on the fact that IRBD, at the initial profile, allows for delegations to take place that only suboptimally improve utility, triggering further delegations at later iterations.

Observe that, in Figure 4.10b and 4.10c, both of the average lengths of delegation chains and the lengths of the longest delegation chains significantly reduce from $\alpha = 0$ to $\alpha = 0.25$, and keep (mildly) lowering until $\alpha = 1$. As agents start to consider to retain voting power (from $\alpha = 0$ to $\alpha = 0.25$), long delegation chains are prevented to maintain voting power, reflecting Fact 2, 3 and 4. Observe that when $\alpha > 0$, all delegations in OSI are direct, i.e., one-hop delegations, which indicates that the difference of the accuracies between the low-accuracy agent and the high-accuracy agent should be large enough, so that the inherited accuracy through delegation can cover the loss of voting power. Furthermore, when $\alpha > 0$, a few agents still delegate through delegation chains longer than 1 (with the longest delegation chain of 2 but the average length around 1), hence we infer that existing delegation may facilitate delegation (through long delegation chains).

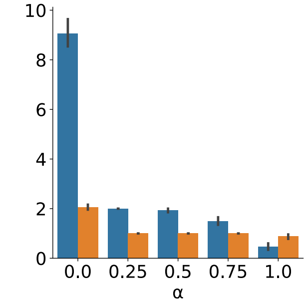
It is worth observing that when $\alpha = 0$, by Figure 4.11d, 4.11e and 4.11f, the minimum/maximum/average DB's are all low (close to 0), due to the long delegation chains. By Fact 3 and 4, both delegators and gurus suffer from losing voting power through long delegation chains. This problem becomes severe in Experiment C: agents lose almost all power due to the long delegation chains (average 5.5 and longest 9). As α grows, the average power increases (Figure 4.10f) and inequality in the distribution of power decreases (Figure 4.10g). Though almost all other trends are monotonic, the maximum DB (Figure 4.10e) of OSI (resp. IBRD) peaks at $\alpha = 0.5$ (resp. $\alpha = 0.75$), from which we infer that the reduction of delegation within some range may promote super-voters to retain voting power.



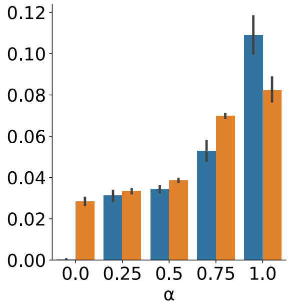
(a) B: ratio of delegators



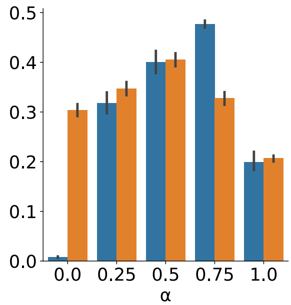
(b) B: Average delegation chain length



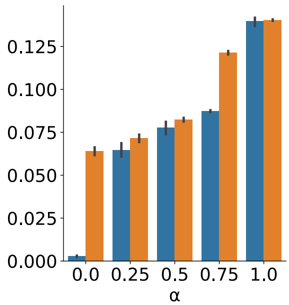
(c) B: The longest delegation chain



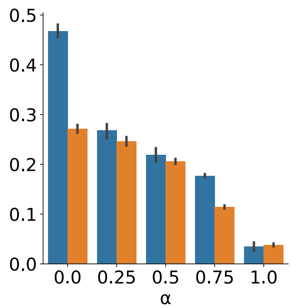
(d) B: Minimum DB



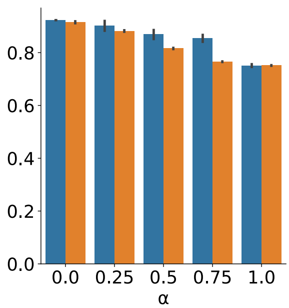
(e) B: Maximum DB



(f) B: Average DB



(g) B: Gini coefficient



(h) B: Average accuracy

Figure 4.10: Experiment B: ■ for OSI and ■ for IBRD.

Consequently, as delegation becomes prevented due to higher α, the average individual accuracy (Figure 4.10h) of both OSI and IBRD decreases.

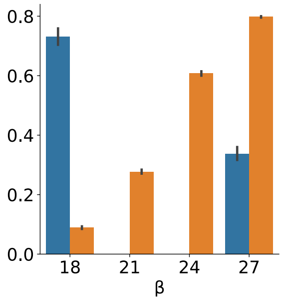
4.3.5 Quota: Experiment C

First observe in Figure 4.11i that much fewer instances of IBRD converge in Experiment C than in Experiment A and Experiment B: e.g., approximately 20 among 50 when $\beta = 27$ and none when $\beta = 21$ and $\beta = 24$. This indicates less robust results than those reported in Experiment A and Experiment B. We conjecture that this is due to the fact that, on average, DB in instances of $\beta = 21$ and 24 is low and hence agent's utility is sensitive to DB's approximation error. This, in turn, should make convergence harder because agents keep changing delegation strategies due to the change of agents' utilities caused by DB's approximation error. However, when $\beta = 27$, the quota is approximately unanimous and agents' DB's become almost identical (see Fact 5). Hence agents are primarily motivated to pursue to delegate to high-accuracy agents instead of to retain voting power. This results in the trend that some instances converge again comparing to those when $\alpha = 21$ and 24, and that the indices when $\beta = 27$ are similar to those when $\alpha = 0$ in Experiment B.

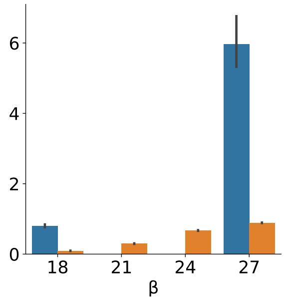
Though the results of the IBRD are less robust, the trends of OSI can still reflect the influence imposed by β. As β increases, delegation is significantly facilitated: by Figure 4.11a, the set of delegators grows from the minority (less than %10 when $\beta = 18$) to the main part of the whole set (approximately %80 when $\alpha = 27$). Note that all delegations are direct (Figure 4.11c). Consequently, we obtain lower average DB (Figure 4.11f) and higher inequality of DB's distribution (Figure 4.11g), as well as increasing average individual accuracy (Figure 4.11h). Notice that when $\beta = 18$, the delegator ratio of IBRD is much higher than that of OSI (Figure 4.11a), which indicates that agents' delegations enormously trigger other agents' incentive to delegate.

CONCLUSION

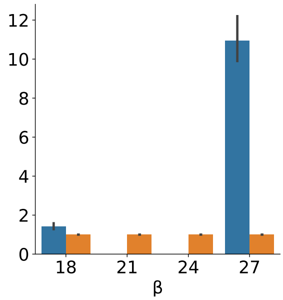
This chapter used the power index developed in Chapter 3 to model a variant of delegation games for liquid democracy where agents seek to find a tradeoff between increasing their accuracy and acquiring power in the system. We showed that for this sort of interaction, pure strategy Nash equilibria are not guaranteed to exist in general. However, the existence of NE was proved in several subclasses of the delegation games, i.e., delegation games with minority quota, with unanimous quota, and under complete underlying networks. Finally, we investigated by computer simulation that delegation is restrained by three parameter settings: less connectivity of the underlying networks, higher extent to which agents are



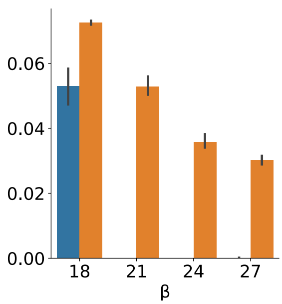
(a) C: ratio of delegators



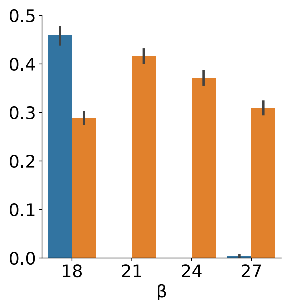
(b) C: Average delegation chain length



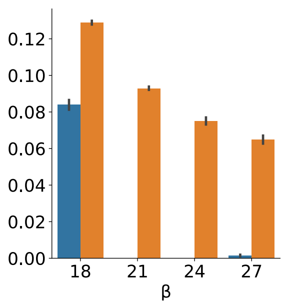
(c) C: The longest delegation chain



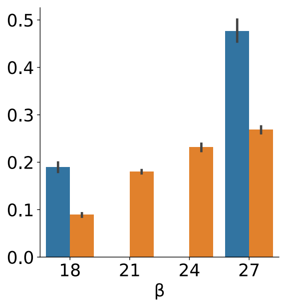
(d) C: Minimum DB



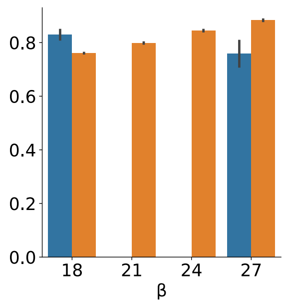
(e) C: Maximum DB



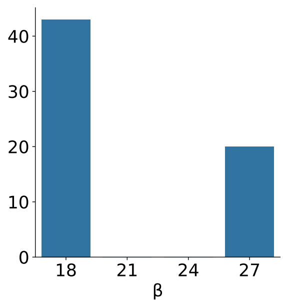
(f) C: Average DB



(g) C: Gini coefficient



(h) C: Average accuracy



(i) C: Converged number of instances

Figure 4.11: Experiment C: ■ for OSI, ■ for IBRD.

motivated by the acquisition of power, and less quota value. These trends are consistent with the theoretical results in Section 3.3. The restrained delegation further leads to lower average accuracy and more equality on voting power distribution among agents.

As to future work, we would like to mention three. First, it would be interesting to understand how much agents' attitude towards power could help in readdressing the deterioration of decision-making quality highlighted by [16, 46], through its equalizing effect on power distribution. (See Preliminaries for this research.) Second, it is worth studying the theoretical guarantee of the existence of pure strategy NE in more general subclasses of the delegation games than those studied in Section 4.2, e.g., delegation games under connected underlying networks. The last is to analytically study the beneficial effect of voting power in limiting accrual of voting weight in liquid democracy observed in the experiments.

5 TRACKING TRUTH BY WEIGHTING PROXIES

This chapter studies the truth-tracking behavior of liquid democracy when agents are allowed to express delegations consisting of the apportionment of shares of a unit weight (i.e., the agent's voting weight) to their proxies. This functionality is available in some implementations of liquid democracy (e.g., on the platform Congressus of the French Pirate Party[1]). We show that in this setting—unlike in the standard one where voting weight is delegated in full to only one proxy—it becomes possible to construct delegation structures that optimize the truth-tracking ability of the group. Then, in the next chapter, focusing on group accuracy, we contrast this centralized solution with the setting in which agents are free to choose their weighted delegations by greedily trying to maximize their own individual accuracy.

CHAPTER CONTRIBUTIONS We first interpret these apportioned weights probabilistically, that is, as mixing of pure delegations. The issue we are after is to understand the extent to which weighted delegations could help the truth-tracking behavior of liquid democracy. We make two contributions. *First*, we show that in this more general setting it is always possible for the agents to achieve maximal group accuracy by centrally coordinating their delegations (Theorem 15). *Second*, we provide an interpretation of weighted delegations alternative to mixing, in which weights are modeled as shares of voting power that agents apportion to their proxies. We show that these two interpretations of weighted delegation coincide under specific conditions, and the maximal group accuracy is also achievable by centralized coordinated delegations in this second interpretation. This part presents and extends material from [69].

5.1 WEIGHTING PROXIES

In this and the following chapter (together constituting Part III), we will be working with the class of symmetric directed graphs, called undirected graphs (recall Section 2.1), which are a subclass of the graphs used in Part II. In such a graph, if a pair of nodes are linked, two edges with different directions must exist between them. We make this assumption for mathematical tractability. We also assume that all underlying networks are connected. That is, in the undirected

1 https://partipirate.org/

graph $R = \langle N, E \rangle$, for any pair of agents $i, j \in N$ with $i \neq j$, we can find at least one path connecting i and j. Additionally, each agent is initially assigned a voting weight 1 (one person one vote). This also restricts us to the more special case $w(i) = 1$ for all $i \in N$.

5.1.1 Weighted Delegations

The model in Part III is also based on the one introduced in Section 2.1, where a set of agents make a decision on a binary issue through a liquid democracy system. We generalize this setting (under the aforementioned assumptions) by allowing agents to apportion parts of their voting power to different proxies: i's delegation now amounts to a stochastic vector $\mathbf{D}_i = (D_{i1}, \ldots, D_{in}) \in \mathbb{R}^n_{\geq 0}$ with $\sum_{j \in N} D_{ij} = 1$. We call such delegations *weighted delegations*. A profile of weighted delegations (*weighted profile*) is an $n \times n$-dimensional stochastic matrix $\mathbf{D} = (\mathbf{D}_1, \ldots, \mathbf{D}_n)$, and $\mathbb{D}$ is the collection of all such profiles. A standard delegation profile $\mathbf{d}$ then corresponds to a degenerate stochastic matrix where each row contains only one entry. We will be referring to standard delegations also as *pure delegations*, and standard delegation profiles as pure delegation profiles (or simply pure profiles).

A weighted profile $\mathbf{D}$ defines a weighted directed graph $G(\mathbf{D}) = \langle N, \xrightarrow{x} \rangle$ where for any pair of $i, j \in N$, a directed edge $i \xrightarrow{D_{ij}} j$ from i to j with label of weight D_{ij} exists whenever $D_{ij} > 0$. Weighted delegation chains, cycles and loops can then be defined on these graphs as we did for graphs of pure delegations in Section 2.1.

Example 17. *Consider a set of agents $N = \{1, 2, 3, 4, 5\}$, with weighted profile $\mathbf{D}$, such that*

- $\mathbf{D}_1$*: $D_{11} = 1$, i.e., maintaining her full voting weight,*
- $\mathbf{D}_2$*: $D_{22} = D_{25} = 0.5$, i.e., agent 2 delegates half to agent 5 and maintains the other half,*
- $\mathbf{D}_3$*: $D_{31} = D_{33} = 0.5$, i.e., agent 3 delegates half to agent 1 and maintains the other half,*
- $\mathbf{D}_4$*: $D_{43} = 1$, i.e., agent 4 fully delegates to agent 3, and*
- $\mathbf{D}_5$*: $D_{52} = 1$, i.e., agent 5 fully delegates to agent 2.*

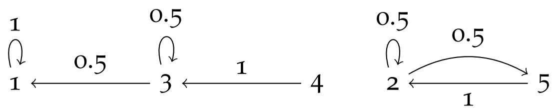


Figure 5.1: Delegation graph of Example 17

We can then denote the weighted delegation profile $\mathbf{D}$ *as the following matrix:*

$$\mathbf{D} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 & 0.5 \\ 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

The corresponding delegation graph is depicted in Figure 5.1.

Several interpretations of a voter's weight become possible under weighted delegations. We deal with two such interpretations. The first one is based on a probabilistic interpretation of the weights, and it will be the one we use to develop our framework. Later, in Section 5.3.2, we are going to interpret weights also as direct transfers of shares of voting weight, and compare the two approaches.

5.1.2 The first Weighting Approach: Expected Weight

Each weighted profile $\mathbf{D}$ can be thought of as describing a probability distribution over pure profiles where the probability of a pure profile $\mathbf{d}$ is $\Pr(\mathbf{d}) = \Pi_{i \in N} D_{i\mathbf{d}(i)}$. We say that pure profile $\mathbf{d}$ is *supported* by weighted profile $\mathbf{D}$, if it has a positive probability, i.e., $\Pr(\mathbf{d}) > 0$. The *weight transfer* of agent i in $\mathbf{D}$, is the vector $\boldsymbol{t}_{\mathbf{D}}(i) = (t_{\mathbf{D}}(i,1), \ldots, t_{\mathbf{D}}(i,n))$ describing how i's weight is distributed in expectation among all guru agents, where:

$$\boldsymbol{t}_{\mathbf{D}}(i,j) = \sum_{\mathbf{d} \in s(\mathbf{D})} \mathbb{1}_{\mathbf{d}^*(i)=j} \Pr(\mathbf{d}) \tag{5.1}$$

where $j \in N$, $s(\mathbf{D}) = \{\mathbf{d} \mid \Pr(\mathbf{d}) > 0\}$ denotes the support of (the probability distribution over pure profiles induced by) $\mathbf{D}$, and $\mathbb{1}_{\mathbf{d}^*(i)=j}$ is the indicator that j is the guru of i in $\mathbf{d}$, i.e., $\mathbb{1}_{\mathbf{d}^*(i)=j} = 1$ if $\mathbf{d}^*(i) = j$, otherwise 0. We will refer to this interpretation of weighted delegations as the *expected weight approach*.

Remark 1. *Interpreting agents' weighted delegations as stochastic strategies is also used in, e.g., the studies of Kahng* et al. *[46] and Halpern* et al. *[41]. In the notion of local delegation mechanism in [46], agents delegate to neighbors with higher accuracy* *with higher probability. In [41], a function $\phi(q_i, q_j)$ is used to decide agent i's stochastic delegation strategy, by taking q_i and q_j, i.e., the accuracy of each neighbor j of i's.*

Using Equation 5.1, we generalize the voting weight accrued by agent i ($i \in N$) in Equation 2.1 to the one consisting of the sum of the weights she receives from all agents:

$$w(i, \mathbf{D}) = \sum_{j \in N} t_{\mathbf{D}}(j, i). \tag{5.2}$$

Equation 5.2 defines, for each $\mathbf{D}$, a vector $\mathbf{w}(\mathbf{D}) = (w(1, \mathbf{D}), \ldots, w(n, \mathbf{D}))$ assigning a weight to each agent, which we call the *weight distribution* of $\mathbf{D}$. Then, as a generalization of the notion of gurus introduced in Section 2.1, we denote by $Gu(\mathbf{D}) = \{i \in N \mid w(i, \mathbf{D}) > 0\}$ the set of gurus in $\mathbf{D}$, that is, the set of agents with positive weight in the weight distribution of $\mathbf{D}$. It is worth observing that $\sum_{i \in N} w(i, \mathbf{D})$ can be less than n (recall that $n = |N|$), because agents may end up having no guru in the induced pure profiles where they are caught into delegation cycles. In such cases, the agent loses the weight corresponding to the probability attached to such pure profiles. We also observe that for any guru $i \in Gu(\mathbf{D})$ in weighted profile $\mathbf{D}$, $D_{ii} > 0$. That is, agent i is a guru in at least one pure profile in $s(\mathbf{D})$.

Example 18 (Example 17, continued)**.** *Consider the weighted delegation profile for the set of 5 agents in Example 17. By the expected weight approach, we consider the weight transfer of each agent as follows:*

- $\boldsymbol{t}_{\mathbf{D}}(1)$. *Agent* 1 *always fully delegates to herself, therefore* $\boldsymbol{t}_{\mathbf{D}}(1) = (1, 0, 0, 0, 0)$.
- $\boldsymbol{t}_{\mathbf{D}}(2)$. *With probability of* 0.5*, agent* 2 *and* 5 *form a delegation cycle, where the weight of the two agents is lost. With probability of* 0.5*, agent* 2 *acts as a guru and retain her weight of* 1*. Therefore* $\boldsymbol{t}_{\mathbf{D}}(2) = (0, 0.5, 0, 0, 0)$.
- $\boldsymbol{t}_{\mathbf{D}}(3)$. *Agent* 3 *delegates to agent* 1 *with probability of* 0.5*, and is a guru with probability of* 0.5*. Hence* $\boldsymbol{t}_{\mathbf{D}}(3) = (0.5, 0, 0.5, 0, 0)$.
- $\boldsymbol{t}_{\mathbf{D}}(4)$. *Agent* 4 *always fully delegates to agent* 3*. However, due to agent* 3*'s delegation strategy, this amount is retained by agent* 3 *with probability of* 0.5*, and by agent* 1 *with probability of* 0.5*. Hence* $\boldsymbol{t}_{\mathbf{D}}(4) = (0.5, 0, 0.5, 0, 0)$.
- $\boldsymbol{t}_{\mathbf{D}}(5)$. *Among all induced pure profiles, in a half, agents* 2 *and* 5 *form a cycle and the weight of these two agents is lost, and in the other half, agent* 5 *delegates to agent* 2*. Therefore* $\boldsymbol{t}_{\mathbf{D}}(5) = (0, 0.5, 0, 0, 0)$.

Then by Equation 5.2, the weight distribution of $\mathbf{D}$ *is* $\mathbf{w}(\mathbf{D}) = (2, 1, 1, 0, 0)$*. Notice that* $\sum_{i \in N} w(i, \mathbf{D}) = 4 < 5$*, since weight amount of* 1 *is lost the delegation cycle formed by agents* 2 *and* 5*.*

5.1.3 Group Accuracy with Weighted Delegations

Each agent i with positive weight in the weight distribution $\mathbf{w}(\mathbf{D})$ votes with accuracy q_i and weight $w(i, \mathbf{D})$. We are interested in the group accuracy under delegation (see Definition 2 in Section 2.2) when gurus vote with accrued weights determined by the expected weight approach described above. We generalize Definition 2 as follows:

$$q_{N,\mathbf{w}(\mathbf{D})} = \sum_{C \in \mathcal{W}(\mathbf{D})} \prod_{i \in C} q_i \prod_{i \in Gu(\mathbf{D}) \setminus C} (1 - q_i), \tag{5.3}$$

where $\mathcal{W}(\mathbf{D})$ is the set of *winning coalitions*, i.e.,

$$\mathcal{W}(\mathbf{D}) = \{C \subseteq Gu(\mathbf{D}) \mid \sum_{i \in C} w(i, \mathbf{D}) > \sum_{i \in Gu(\mathbf{D}) \setminus C} w(i, \mathbf{D})\}. \tag{5.4}$$

We abuse notation and extend $\mathcal{W}(\mathbf{D})$ to include one of the two equinumerous coalitions in case of ties uniformly at random if $\sum_{i \in C} w(i, \mathbf{D}) = \sum_{i \in Gu(\mathbf{D}) \setminus C} w(i, \mathbf{D})$. Note that this is different from the quota rule used in Chapter 3 where a deterministic tie breaker is used. It is worth noting that weight may be lost due to cycles (i.e., $\sum_{i \in Gu(\mathbf{d})} < n$). That is why a winning coalition needs to be defined with weight more than $\sum_{i \in Gu(\mathbf{D})} w(i, \mathbf{D})/2$ instead of $n/2$.

Observe that the group accuracy under weighted delegations depends on the weight distribution $\mathbf{w}(\mathbf{D})$, or, in other words, depends on the interpretation of weighted delegations. As mentioned above, we also introduce in Section 5.3.2 another method to interpret weighted delegations, which usually produces different weight distributions than the expected weight approach, and consequently results in different group accuracies.

Example 19 (Example 18, continued)**.** *We use Example 18 with accuracy profile* $\mathbf{q} = (0.9, 0.9, 0.6, 0.6, 0.6)$. *Consider the weight distribution* $\mathbf{w}(\mathbf{D}) = (2, 1, 1, 0, 0)$. *We have that* $Gu(\mathbf{D}) = \{1, 2, 3\}$. *Notice that there is a tie between coalitions* $\{2, 3\}$ *and* $\{1\}$. *Since the tie is broken uniformly at random, we have two possible cases:*

1. *Coalition* $\{2, 3\}$ *is chosen as the winning one. Then all winning coalitions are* $\mathcal{W}(\mathbf{D}) = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}\}$. *Then take the voting instance of the winning coalition* $\{1, 2\}$, *where agents* 1 *and* 2 *vote correctly but* 3 *incorrectly. This instance has probability* $q_1 q_2 (1 - q_3) = 0.324$. *Then the group accuracy is the sum of all accuracies of these winning coalitions, i.e.,* $q_{N,\mathbf{w}(\mathbf{D})} = 0.324 + 0.054 + 0.054 + 0.486 = 0.918$.

2. *Coalition* $\{1\}$ *is the winning one. By a similar computation, we have that* $q_{N,\mathbf{w}(\mathbf{D})} = 0.324 + 0.054 + 0.036 + 0.486 = 0.9$.

5.2 CENTRALIZED WEIGHTED DELEGATIONS

Our motivation to study weighted delegations comes from the known generalization of the Condorcet jury theorem (Theorem 4) showing that the chance that the voting outcome of the group is correct is maximized if a weighted majority rule is used with a specific choice of weights. That is, if for all agents $i \in N$, the voting weight is proportional to $\log(\frac{q_i}{1-q_i})$[2], then the group accuracy q_N achieves maximal.

We can leverage Theorem 4 to solve the optimal delegation problem: given a set of agents with different accuracies, what is the weighted delegation graph that maximizes group accuracy? We develop an answer to this question in two steps. First, to fix intuitions, we provide a solution for complete networks (Algorithm 5), and then move to connected networks (Algorithm 6).

5.2.1 Centralized Delegations in Complete Nets

In complete networks, all agents can delegate to all the other agents. We propose an algorithm that uses one-hop weighted delegations to reallocate weight from the less accurate to the more accurate agents in the group. We define the optimal weight of each agent $i \in N$ by:

$$w_i^\star = n \cdot \frac{\log \frac{q_i}{1-q_i}}{\sum_{j \in N} \log \frac{q_j}{1-q_j}}, \tag{5.5}$$

where n is the number of agents (and thus it is also the entire amount of voting weight due to our one-person-one-vote setting), $\lim_{q_i \to 1} \log \frac{q_i}{1-q_i} = \infty$, and $\lim_{q_i \to 0.5} \log \frac{q_i}{1-q_i} = 0$ (detailed value trend is shown in Figure 2.2). Notice that this weight is larger than 1 for the more accurate agents whereas it is smaller for the less accurate ones and, as desired, it is proportional to $\log \frac{q_i}{1-q_i}$. The idea behind the algorithm (Algorithm 5) is to have the agents i with $w_i^\star > 1$ apportion their full weight to themselves, and have each agent j with $w_j^\star < 1$ apportion shares $w_i^\star - 1$, normalized by the total excess weight of the i agents, of the excess weight $1 - w_j^\star$ to each agent i. Notice that if all agents are equally accurate ($N = N_3$), Algorithm 5 returns the trivial profile. We use Example 19 to illustrate Algorithm 5.

Example 20 (Example 19, continued). *We first compute* $\log \frac{q_i}{1-q_i}$ *for all* $i \in N$, *and obtain vector* $(0.9542, 0.9542, 0.1761, 0.1761, 0.1761)$. *Then the optimal weight vector* $\mathbf{w}_i^\star$ *is computed by normalizing the above vector by the entire weight, in this case,*

2 Note that all the following results do not change by different choice of the logarithm's base, and we use natural base of e.

Algorithm 5 Optimal delegations in complete networks

INPUT: N, $\mathbf{w}^\star$

INITIALIZE: $N_1 = \{i \in N \mid w_i^\star < 1\}$, $N_2 = \{i \in N \mid w_i^\star > 1\}$, $N_3 = \{i \in N \mid w_i^\star = 1\}$, $w = \sum_{i \in N_2}(w_i^\star - 1)$.

DELEGATE:

1: For $i \in N_3$: $D_{ii} = 1$.
2: For $i \in N_2$: $D_{ii} = 1$.
3: For $i \in N_1$: for all $j \in N_2$, $D_{ii} = w_i^\star$, $D_{ij} = (1 - w_i^\star)\frac{w_j^\star - 1}{w}$.

RETURN: $\mathbf{D}$

---

obtaining vector $\mathbf{w}^\star = (1.958, 1.958, 0.3613, 0.3613, 0.3613)$. *We can observe that* $w_1^\star$ *and* $w_2^\star$ *are larger than the initial weight* 1 *and therefore according to the algorithm, they will not delegate. For any other agents* $i \in \{3, 4, 5\}$*, they delegate the excess weight above* $w_i^\star = 0.3613$ *(i.e.,* $1 - w_i^\star$*) to agents* 1 *and* 2 *by share* $\frac{w_1^\star}{w_1^\star + w_2^\star}$ *and* $\frac{w_2^\star}{w_1^\star + w_2^\star}$ *respectively. That is, in this case, each agent* i *delegates to agents* 1 *and* 2 *equally since* $w_1^\star = w_2^\star$*. Therefore the returned profile is* $\mathbf{D}$*, in which:*

- $D_{11} = D_{22} = 1$,
- $\mathbf{D}_3 = (0.31935, 0.31935, 0.3613, 0, 0)$,
- $\mathbf{D}_4 = (0.31935, 0.31935, 0, 0.3613, 0)$, *and*
- $\mathbf{D}_5 = (0.31935, 0.31935, 0, 0, 0.3613)$.

Then for all $i \in N$, $w(i, \mathbf{D}) = w_i^\star$.

Theorem 14. *Given a set of agents* N*, with accuracy profile* $\mathbf{q}$*, and an underlying network* R*, if* R *is complete, then Algorithm 5 outputs an element of* $\arg\max_{\mathbf{D} \in \mathbb{D}} q_{N, \mathbf{w}(\mathbf{D})}$.

Proof. By Theorem 4, $\mathbf{D} \in arg\max_{\mathbf{D} \in \mathbb{D}} q_{N, \mathbf{w}(\mathbf{D})}$ if $w(i, \mathbf{D}) \propto \log(\frac{q_i}{1 - q_i})$ for all $i \in N$. Observe that (N_1, N_2, N_3) is a tri-partition of N. Agents in N_1 have optimal weight below 1, and this requires them to delegate part of their weight to agents in N_2 whose optimal weight is above 1. We discuss each element of the tri-partition in turn. For all agents in N_3, their optimal weight is exactly 1. Therefore they just need to be gurus without incoming delegations to reach their optimal weight, i.e., for all $i \in N_3$, by line 1 in Algorithm 5, $D_{ii} = 1$ and $w(i, \mathbf{D}) = 1 = w_i^\star$.

We then consider N_1 and N_2 in turn. Note that no weight is lost in the returned weighted profile, since no cycle can be formed by Algorithm 5.

First consider all agents in N_1. By line 3 of Algorithm 5, for all $i \in N_1$, $D_{ii} = w_i^\star$, and for all $j \in N \setminus \{i\}$, $D_{ji} = 0$, thus $w(i, \mathbf{D}) = D_{ii} = w_i^\star$. For the excess weight of i, i.e., $1 - w_i^\star$, she delegates a proportion of it to each agent in N_2. The proportion is decided by $\frac{w_j^\star - 1}{w}$ for all $j \in N_2$ (note that $w = \sum_{j \in N_2}(w_j^\star - 1)$), that is, agent j is expected to receive weight amount of $(1 - w_i^\star)\frac{w_j^\star - 1}{w}$ from all $i \in N_1$.

Then any agent $j \in N_2$, in total, is expected to receive $\sum_{i \in N_1}(1 - w_i^\star)\frac{w_j^\star - 1}{w}$. Moreover, $D_{jj} = 1$ (line 2, Algorithm 5), which indicates j is expected to delegate all her weight to herself, i.e., amount of 1. Notice that for all agents in N_1 and N_2, $\sum_{k \in N_1 \cup N_2} w_k^\star = |N_1| + |N_2|$, and hence $\sum_{k \in N_1}(1 - w_k^\star) = |N_1| - \sum_{k \in N_1} w_k^\star = \sum_{k \in N_1 \cup N_2} w_k^\star - |N_2| - \sum_{k \in N_1} w_k^\star = \sum_{k \in N_2} w_k^\star - |N_2| = \sum_{k \in N_2}(w_k^\star - 1)$. Therefore j collects $\sum_{k \in N_2}(w_k^\star - 1)\frac{w_j^\star - 1}{w} + 1 = w_j^\star$, as desired. □

The next example shows how the optimal accuracy via weighted delegations may be higher than that achievable via pure delegations.

Example 21 (Example 20, continued)**.** *Let us continue with Example 20. Following the intuition of achieving a trade-off between enhancing the individual accuracy level by coordinating low-accuracy agents to delegate to high-accuracy agents, and maintaining a large enough number of gurus such that more agents make decisions independently, we find that the optimal pure delegation profile is the one in which only one delegation happens: an agent with accuracy of* 0.6 *delegates to an agent with accuracy of* 0.9*. Then the optimal (pure profile) majority accuracy is* 0.918*, which is lower than the optimal accuracy,* 0.92664*, of the weighted profile* $\mathbf{D}$*.*

Intuitively, pure delegations allow for only discrete weights and can therefore only approximate the optimal weight distribution among gurus in which each winning coalition $C \subseteq Gu(\mathbf{d})$ of gurus is more accurate than the corresponding losing coalition $Gu(\mathbf{d}) \setminus C$.

Remark 2. *It is worth discussing Algorithm 5 also in the context of the GreedyCap algorithm of Kahng* et al. *[46]. GreedyCap is a semi-local probabilistic delegation algorithm, with a centralized element: a cap on the maximal number of delegations, which avoids the creation of too much dependence among voters and thus preserving wisdom-of-the-crowd effects. In GreedyCap, each agent may delegate to the highest-accuracy agent in her neighborhood as long as the number of received delegations of the highest-accuracy agent does not exceed the cap. Algorithm 5 implements a fully centralized approach to group accuracy by assuming delegations to be centrally determined.*

5.2.2 Centralized Delegations in Connected Nets

We now extend Algorithm 5 to the case of connected networks. We first fix some notation before introducing the algorithm. Given a non-negative matrix

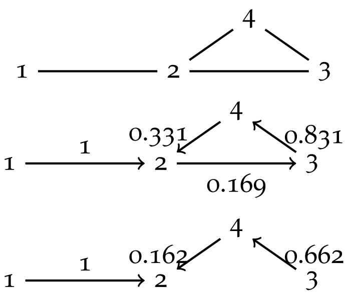


Figure 5.2: Network underlying Example 22 (top) and depiction of two intermediate steps of Algorithm 22 (middle and bottom)

$\mathbf{A} \in \mathbb{R}^{n \times n}_{\geq 0}$, $G(\mathbf{A}) = \langle N, E(\mathbf{A}) \rangle$ denotes the directed graph[3] induced by $\mathbf{A}$, i.e., for any pair of $i, j \in N$, $i \xrightarrow{A_{ij}} j$ if and only if $\mathbf{A}_{ij} > 0$, and the label $A_{ij} = \mathbf{A}_{ij}$. Let furthermore $\mathcal{C}(\mathbf{A})$ denote all cycles in $G(\mathbf{A})$.

Algorithm 6 generalizes the idea of Algorithm 5 to connected networks as follows. Similar to Algorithm 5, each agent expected to delegate (i.e., in N_1) transfers part of her excess weight to each of the agents who should receive delegations (i.e., in N_2). Note that $(N_1 \cup N_2) \subseteq N$. Then the **DETERMINE PATHS** component in Algorithm 6 first decides an acyclic path between each such pair of agents, and the **LABEL REQUIRED TRANSFER OF WEIGHT** component labels the expected transfer weight amount on each edge on the path. For example, between a pair of agents $i \in N_1$ and $j \in N_2$, a certain amount of weight is expected to be transferred from i to j through a path L_{ij}. On the edge between each pair of adjacent agents on L_{ij}, say k and k', the above amount is labeled as $e^{i,j}_{k,k'}$. Hence for each node, all incoming and outgoing edges are labeled with weight (one edge might have several labels). Observe that between any pair of nodes, two edges with different directions may exist. We then aggregate the net expected transfer amount between the pair of nodes. Note that cycles may exist. We break every cycle by subtracting from each edge in the cycle the amount equal to the minimum amount among all edge labels in the cycle, by the component **REMOVE CYCLES**. Finally by the **DECIDE WEIGHTS** component, each agent decides her delegation strategy by computing the proportion of each expected outgoing weight in total incoming weight (including her initial weight). An illustration of the algorithm follows.

Example 22. *Consider a network with 4 agents* $N = \{1, 2, 3, 4\}$, *and accuracy profile* $\mathbf{q} = (0.5, 0.9, 0.6, 0.9)$. *Let R be as in Figure 5.2 (top). By Equation* (5.5) *the optimal*

---

3 Note that we use this notation for delegation graphs, which are directed. The underlying networks used are still undirected as noticed in the beginning of this chapter.

Algorithm 6 Optimal delegations in connected networks

INPUT: $\mathbf{w}^*$, $R = \langle N, E \rangle$

INITIALIZE:

$A = 0^{n \times n}$, $\forall i \in N_1, j \in N$, $D_{i,j} = 0$, $N_1 = \{i \in N \mid w_i^\star < 1\}$, $N_2 = \{i \in N \mid w_i^\star > 1\}$.

DETERMINE PATHS:

For all $i, j \in N$ ($i \neq j$), select an arbitrary acyclic path L_{ij} in R.

LABEL REQUIRED TRANSFER OF WEIGHT:

For all $(k, k') \in L_{ij}$:

1: For $i \in N_1$, $j \in N_2$: $e^{i,j}_{k,k'} = (1 - w_i^\star) \frac{w_j^\star - 1}{\sum_{\ell \in N_2}(w_\ell^\star - 1)}$.

2: For $i \in N$ and $j \in R'(i)$: $\mathbf{A}_{i,j} = \sum_{k \in N_1, k' \in N_2} e^{k,k'}_{i,j} - \sum_{k \in N_2, k' \in N_1} e^{k,k'}_{j,i}$, if $\sum_{k \in N_1, k' \in N_2} e^{k,k'}_{i,j} - \sum_{k \in N_2, k' \in N_1} e^{k,k'}_{j,i} > 0$.

REMOVE CYCLES:

For $c \in \mathcal{C}(\mathbf{A})$, and $(k, k') \in c$:

1: $\mathbf{A}_{k,k'} = \mathbf{A}_{k,k'} - \min_{(\ell,\ell') \in c}(\mathbf{A}_{\ell,\ell'})$.

DECIDE WEIGHTS:

For $i \in N$ and $j \in R'(i)$:

1: If $\mathbf{A}_{i,j} > 0$: $D_{i,j} = \mathbf{A}_{i,j} / (\sum_{\ell \in R'(i), \mathbf{A}_{\ell,i} > 0} \mathbf{A}_{\ell,i} + 1)$.

2: $D_{i,i} = (\sum_{\ell \in R'(i), \mathbf{A}_{\ell,i} > 0} \mathbf{A}_{\ell,i} + 1 - \sum_{\ell \in R'(i), \mathbf{A}_{i,\ell} > 0} \mathbf{A}_{i,\ell}) / (\sum_{\ell \in R'(i), \mathbf{A}_{\ell,i} > 0} \mathbf{A}_{\ell,i} + 1)$.

RETURN: $\mathbf{D}$

---

weight distribution is $\mathbf{w}^\star = (0, 1.831, 0.338, 1.831)$, *and therefore* $N_1 = \{1, 3\}$ *and* $N_2 = \{2, 4\}$.

We first determine the weight transfer paths, and assume that $L_{1,2} = \{(1,2)\}$, $L_{1,4} = \{(1,2),(2,3),(3,4)\}$, $L_{3,4} = \{(3,4)\}$, *and* $L_{3,2} = \{(3,4),(4,2)\}$. *Then we compute the expected weight transfers. For instance, agent* 1 *should transfer amount* $(1 - w_1^\star \frac{w_2^\star - 1}{\sum_{i \in N_2}(w_i^\star - 1)}$ $= 0.5$ *to agent* 2. *Therefore, for each path in* $L_{1,2}$, *i.e.,* $(1,2)$, $e^{1,2}_{1,2} = 0.5$. *Similarly, we label all required transfer weights, aggregate the net amounts between each pair of agents, and obtain the labeled graph in Figure 5.2 (middle). Observe that there is a cycle formed by agents* 2, 3 *and* 4. *This cycle is broken by subtracting amount* 0.169, *which is the minimum amount among those of edges* $(2,3)$, $(3,4)$ *and* $(4,2)$, *from each edge in the cycle and obtain the acyclic graph in Figure 5.2 (bottom).*

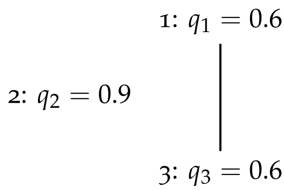


Figure 5.3: The underlying network and accuracy profile in Example 23.

Finally, we compute the weighted profile. Let us take agent 4 *as an instance. Observe that agent* 4 *is expected to transfer* 0.162 *to agent* 2*, while she simultaneously receives* 0.662 *from agent* 3. 0.162 *takes* 9.75% *among agent* 4*'s received amount* 0.662 *plus her initial weight amount of* 1. *We thus obtain* $D_{42} = 0.0975$ *and* $D_{44} = 1 - D_{42} = 0.9025$. *Similarly we have* $D_{12} = 1$, $D_{22} = 1$, $D_{34} = 0.662$, *and* $D_{33} = 0.338$.

We can prove that Algorithm 6 achieves optimal accuracy:

Theorem 15. *Given a set of agents* N*, with accuracy profile* $\mathbf{q}$*, and the underlying network* R*, if* R *is connected, Algorithm 6 outputs an element of* $\arg\max_{\mathbf{D}\in\mathbb{D}} q_{N,\mathbf{w}(\mathbf{D})}$.

Proof. First recall that $\mathbf{w}^\star$ is an optimal weight distribution given by Theorem 4. Then notice that in the algorithm, we transfer the amount of $(1 - w_i^\star)\frac{w_j^\star - 1}{\sum_{\ell\in N_2}(w_j^\star - 1)}$ from each $i \in N_1$ to each $j \in N_2$, by line 1 in the **Label Required Transfer of Weight** component, since we can find a path from any agent in N_1 to any agent in N_2 in the connected network R. This amount is labeled on each edge of each above path, and the net weight transfer is aggregated on each edge by line 2 in the Label Required Transfer of Weight component. These transfers are not cyclical because of Remove Cycles which takes care of removing cycles.

So at this point, the algorithm has constructed an acyclic graph encoding the required transfer $\mathbf{A}_{ij}$ of expected weight between any pair of agents i and j. This amount needs to be normalized for each agent by the Decide Weights routine. For any agent $i \in N$, if she is required to transfer positive weight $A_{i,i'}$ to a neighbor $i' \in R(i)$, the weighted strategy $\mathbf{D}_{ii'}$ should be the proportion of $\mathbf{A}_{ii'}$ in her total required incoming weight, plus her original endowed weight, i.e., $\sum_{\ell\in R'(i),\mathbf{A}_{\ell,i}>0} \mathbf{A}_{\ell,i} + 1$ (line 1 in the Decide Weights component). The obtained weighted profile $\mathbf{D}$ thus ensures that for any pair of agents $i \in N_1$ and $j \in N_2$, $t_{\mathbf{D}}(i,j) = (1 - w_i^\star)\frac{w_j^\star - 1}{\sum_{\ell\in N_2}(w_\ell^\star - 1)}$. Therefore by Equation (5.2), $\mathbf{w}(\mathbf{D}) = \mathbf{w}^\star$ as desired. □

We observe that the connectedness of R, which we assume throughout this chapter, is a necessary condition for Theorem 15 to go through.

Example 23 (Connectedness is necessary for Theorem 15)**.** *Let* $N = \{1, 2, 3\}$*, and let the set of edges in* R *be* $E : \{(1, 3)\}$ *(that is,* R *is disconnected), as shown in Figure 5.3.*

Observe that in this network, only one undirected edge exists from agent 1 *to* 3*, that is, only agents* 1 *and* 3 *can delegate to each other, and agent* 2 *can only be a guru. Now let the accuracy profile be:* $\mathbf{q} = (0.6, 0.9, 0.6)$.

Then, applying Algorithm 6 to the component consisting of agents 1 *and* 3*, and to the single component* 2*, respectively, would output a weighted profile* $\mathbf{D}$ *with the trivial weight distribution of* $\mathbf{w}(\mathbf{D}) = (1, 1, 1)$*, which is different from the optimal weight distribution* $\mathbf{w}^\star = (0.053, 2.894, 0.053)$*, since the weight in connected component* $\{(1, 3)\}$ *cannot be transferred to agent* 2 *because of the disconnected underlying network.*

Observe also that if R is complete, the path L_{ij} between any pair of agents $i \in N_1$ and $j \in N_2$ can be selected to be the one-hop edge (i, j). Algorithm 6 then reduces to Algorithm 5.

5.3 WEIGHTS AS SHARES OF VOTES

We have so far developed our theory based on the expected weight approach of Equations (5.1) and (5.2). This is not the only way in which agents' weights can be interpreted under weighted delegations. In this section we briefly highlight another interpretation, which we call the *limit weight approach*, and relate it to the *expected weight approach* we developed in the previous section.

By the limit weight approach, we investigate agents' weight transfer in the limit, studying weighted delegation profiles as Markov chains. In the next section, we introduce the basic concepts of Markov chain theory, which we will use to develop the limit weight approach.

5.3.1 Rudiments of Markov Chains

We will be concerned specifically with discrete-time Markov chains [28] with finite state space, to which we will refer simply as Markov chains.

Let $S = \{s_1, s_2, \ldots, s_m\}$ be a finite set of *states*. A *Markov chain* is a sequence of variables $X_1, X_2, X_3, \ldots$, where each variable $X_t \in S$ describes the state at time step $t \geq 1$, and the sequence satisfies the Markov property, i.e.,

$$\Pr(X_{n+1} \mid X_1, X_2, \ldots, X_n) = \Pr(X_{n+1} \mid X_n).$$

Intuitively, in a Markov chain, the probability of moving from the current state (i.e., X_n) to the next state (i.e., X_{n+1}) only depends on the current state X_n, but not on any previous states.

Given that there are m states, we can then use a *transition matrix* $\mathbf{A} \in \mathbb{R}^{m \times m}_{\geq 0}$ to denote the transition probability between each pair of states. Any element in $\mathbf{A}$, $A_{i,j}$, denotes the probability of moving from state s_i to state s_j, i.e., $\Pr(X_{n+1} = s_j \mid X_n = s_i) = A_{i,j}$. Therefore, $\mathbf{A}$ must be a *stochastic matrix*, where $\sum_{j=1}^{m} A_{i,j} = 1$

for all $i \in \{1, 2, \ldots, m\}$[4]. Then, the power of k to the transition matrix $\mathbf{A}^k$ denotes the probability of moving from one state to another state in k steps. For instance, for any pair of states $s_i, s_j \in S$, $\mathbf{A}^2_{i,j} = \sum_{1 \leq \ell \leq m} A_{i,\ell} A_{\ell,j}$ denotes the probability that from the current state s_i, the Markov chain moves to state s_j in two steps.

As we already observed earlier, a transition matrix $\mathbf{A}$ can be denoted as a directed graph $\langle S, E \rangle$, where the set of nodes is S and between each pair of nodes (or states) $s_i, s_j \in S$, a directed edge $(s_i, s_j) \in E$ exists if and only if $A_{i,j} > 0$, which is labeled with $A_{i,j}$. That is, each edge in the graph denotes the probability of transition between the pair of states.

CONVERGENCE We can now use a stochastic m-dimensional vector $\pi \in \mathbb{R}^m_{\geq 0}$, called a *distribution vector*, to denote the probability of each state to which the chain moves. For example, a distribution vector with the i-th ($1 \leq i \leq m$) element being 1 denotes that the Markov chain moves to state s_i with probability of 1. From a distribution vector π^t at time step t, we can induce the distribution vector of the next time step (i.e., $t + 1$) by the production of π^t and the transition matrix $\mathbf{A}$, i.e., $\pi^{t+1} = \pi^t \mathbf{A}$. Given a transition matrix $\mathbf{A}$, a distribution vector π *converges* if a time step t^* exists, such that for all $t \geq t^*$, $\pi^t \mathbf{A} = \pi^{t+1} = \pi^*$, and we call π^* a *stationary* distribution vector. Intuitively, a distribution vector converges if it stabilizes after a finite number of time steps through the transition matrix.

We call a state s_i an *absorbing* state if, starting from s_i, the Markov chain stays in s_i for all following time steps, i.e., $A_{i,i} = 1$. In other words, if at some time step, the Markov chain moves to s_i, it is then stuck in s_i forever.

However, a distribution vector cannot always converge. If that is the case, some state in the Markov chain is *periodic*. Let $\Pr^k(s_i, s_j)$ denote the probability that a Markov chain moves from state $s_i \in S$ to state $s_j \in S$ in k steps.

RECURRENCY State s_i is *recurrent* if $\sum_{k \geq 1} \Pr^k(s_i, s_i) = 1$, that is, the Markov chain moves from s_i back to itself with probability 1 at some future time step.

PERIODICITY In a given Markov chain, for a state $s_i \in S$, we have a set of positive integers $K = \{k \in \mathbb{Z}_{\geq 0} \mid \Pr^k(s_i, s_i)\}$. Then state s_i is periodic if $\gcd\{K\} > 1$, where gcd is the greatest common divisor. Periodicity is an important condition in verifying whether a distribution vector converges or not.

Lemma 7 ([28]). *If a Markov chain can move from any recurrent state to any other recurrent state with positive probability and each state is aperiodic, any distribution vector converges.*

Observe also that in the directed graph of transition matrix $\mathbf{A}$, if a state in a strongly connected component[5] is periodic, all the other states in the strongly

4 Such a stochastic matrix is also called a *right stochastic matrix*.
5 Recall that, in a directed graph, a strongly connected component is a subgraph, in which there is at least a path from each node to each of the other nodes.

connected component are also periodic [28]. Therefore, we say that a strongly connected component is periodic if all states in the component are periodic, and we further say that a directed graph is periodic if all states (or nodes) in the graph are periodic.

We illustrate some of the above concepts in the following example.

Example 24. *Consider a transition matrix* $\mathbf{A}$ *with* 8 *states.*

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.5 & 0.5 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.5 & 0.5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

This transition matrix is described as a graph in Figure 5.4. Below we illustrate the evolution of the chain from three different starting states.

1. *The initial state is* s_1 *with probability 1. The distribution vector is* $\pi^1 = (1, 0, 0, 0, 0, 0, 0, 0)$. *At time step 2,* $\pi^2 = \pi^1\mathbf{A} = (0, 1, 0, 0, 0, 0, 0, 0)$, *which implies that the Markov chain must move from* s_1 *to* s_2*. At time step 3, we have that* $\pi^3 = \pi^1\mathbf{A}^2 = (\frac{1}{2}, \frac{1}{2}, 0, 0, 0, 0, 0, 0)$, *which means that in two steps, the Markov chain may move to* s_1 *or* s_2 *with probabilities* $\frac{1}{2}$ *and* $\frac{1}{2}$*, respectively. Then, in the limit, the distribution vector converges, and* $\lim_{k\to\infty} \pi^1\mathbf{A}^k = (\frac{1}{3}, \frac{2}{3}, 0, 0, 0, 0, 0, 0)$, *which is stationary. This means that after a large enough number of time steps, the Markov chain may move to state* s_1 *(respectively,* s_2*) with probability* $\frac{1}{3}$ *(respectively,* $\frac{2}{3}$*).*

2. *The initial state is* s_3*. The distribution vector is* $\pi^1 = (0, 0, 1, 0, 0, 0, 0, 0)$*. Then we have from time step 1 to 4:*
   - $\pi^2 = \pi^1\mathbf{A} = (0, 0, 0, \frac{1}{2}, \frac{1}{2}, 0, 0, 0)$;
   - $\pi^3 = \pi^1\mathbf{A}^2 = (0, 0, 0, 0, 0, 1, 0, 0)$;
   - $\pi^4 = \pi^1\mathbf{A}^3 = (0, 0, 1, 0, 0, 0, 0, 0)$.

   Notice that the distribution vector loops back in 3 steps, i.e., $\pi^4 = \pi^1$*. We actually have that in this case, for any* $k \geq 1$, $\pi^k = \pi^{k+3}$*. Then* s_3 *is periodic.*

3. *The initial state is* s_7*. The starting distribution vector is* $\pi^1 = (0, 0, 0, 0, 0, 0, 1, 0)$, *and then it becomes* $\pi^2 = \pi^1\mathbf{A} = (0, 0, 0, 0, 0, 0, 0, 1)$, *which is stationary. Observe that no matter whether starting from states* s_7 *or* s_8*, the stationary distribution*

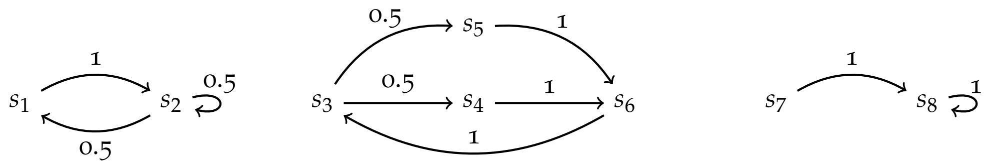


Figure 5.4: The figure denoting the transition matrix in Example 24.

vector can only be the one in which state s_8 has probability of 1. Therefore, s_8 is an absorbing state.

With the above concepts, we are ready to introduce the limit weight approach to weighted delegations.

5.3.2 Limit Weight

Each weighted profile $\mathbf{D}$ describes the direct transfer of voting weight between any two agents: D_{ij} is the share of i's power transferred to j. The indirect transfer of weight, via transitive delegations, is therefore described by the powers of $\mathbf{D}$. For example, $\mathbf{D}^2_{ij} = \sum_{k \in N} D_{ik} D_{kj}$ is the share of power transferred in two steps from i to j. In this view, the weight transfer of agent i consists of the transfer of i's weight in the limit, described by the limit vector

$$\mathring{t}_{\mathbf{D}}(i) = \lim_{k \to \infty} \mathbf{1}_i \mathbf{D}^k \tag{5.6}$$

when such limit exists,[6] and where $\mathbf{1}_i$ is the n-dimensional vector where all elements are 0's except for the i-th one which is 1. This approach treats each agent as a state and $\mathbf{D}$ as a stochastic transition matrix describing to whom i's original weight of 1 'flows' in $\mathbf{D}$. So we are treating the distribution of votes as a Markov chain. The j-th element of $\mathring{t}_{\mathbf{D}}(i)$, denoted as $\mathring{t}_{\mathbf{D}}(i, j)$, is the weight transfer from agent i to agent j under weighted profile $\mathbf{D}$, and an agent's accrued weight is then:

$$\mathring{w}(i, \mathbf{D}) = \sum_{j \in N} \mathring{t}_{\mathbf{D}}(j, i), \tag{5.7}$$

defining the weight distribution $\mathring{\mathbf{w}}(\mathbf{D}) = (\mathring{w}(1, \mathbf{D}), \ldots, \mathring{w}(n, \mathbf{D}))$, to which we refer as the stationary weight distribution. We call an agent an *absorbing agent* if she has a loop weighted 1, and a strongly connected component of the delegation

6 It is worth remarking that the inexistence of $\lim_{k \to \infty} \mathbf{D}^k$ does not necessarily imply the inexistence of $\lim_{k \to \infty} \mathbf{1}_i \mathbf{D}^k$.

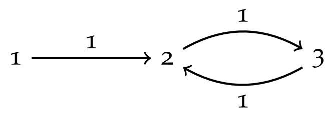


Figure 5.5: Delegation graph of Example 25

graph of $\mathbf{D}$ an *absorbing component* if it has no out-degree[7]. In other words, if weight "flows" to an absorbing agent or component, then it will be caught into the agent or component forever. Note that for the limit weight approach, we may also have $\sum_{i \in N} \mathring{w}(i, \mathbf{D}) < n$, since each agent caught in a non-convergent strongly connected component of the delegation graph loses her weight.

Remark 3. *We remark that the definition of convergence in Equation 5.6 is slightly different from the standard convergence definition in Markov chain theory which requires that the distribution vector becomes stationary in finite steps, and is therefore stronger.*

Example 25. *In this example, we show a case in which weight does not converge in the limit. Consider $N = \{1, 2, 3\}$, with weighted profile as in Figure 5.5. In the first step, agents 2 and 3 exchange their voting weights (amount of 1), and meanwhile, agent 1 delegates her full weight to agent 2. Therefore after step 1, agent 2 accrues weight of 2 and agent 3 accrues 1. Then, in time step 2, agents 2 and 3 keep exchanging their weights, and as a result, agent 2 now has weight 1 while agent 3 has 2, and their weights keep exchanging thereafter. Hence the transfer of weight does not converge in this case.*

Observe that the strongly connected component consisting of agents 2 and 3 is periodic *and each agent in the component is recurrent. That is, the state of each agent of 2 and 3 repeatedly returns to the same state in every two steps with probability 1.*

Remark 4. *The limit weight approach is related to the so-called influence matrices studied in the literature on the DeGroot model [26, 44] and on power in organizations [35, 42, 43]. Both these strands of literature define limit influence notions as done in Equation (5.6).*

Remark 5. *Note that the Markov chain of the weight distribution in a delegation graph consisting of only a delegation cycle where each agent fully delegates to the next agent in the cycle (as in Figure 5.6), does not converge. Even though it might seem that we could have a stabilized weight distribution of $(1, 1, \ldots, 1)$ in the one-person-one-vote setting in such a delegation graph, the weight transfer of each agent does not converge. Starting with the initial weight distribution of an agent in the cycle $(0, \ldots, 1, \ldots, 0)$, her weight transfer changes back after each n steps, so each agent is a periodic state in the chain.*

7 We say a strongly connected component has no out-degree if no agent in the component has any out-degree pointing to an agent who does not belong to the component.

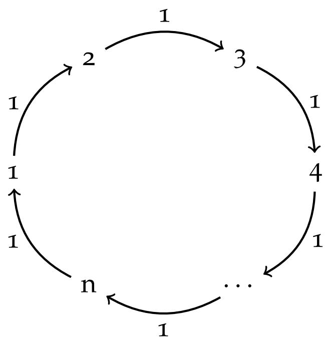


Figure 5.6: A full delegation cycle.

5.3.3 Expected Weight vs. Limit Weight

In general, the two weight approaches introduced above may result in different weight distributions. We first provide a comparison of the two approaches in the following two examples.

Example 26. *As in Example 18, consider* $N = \{1, 2, 3, 4, 5\}$*, with weighted profile* $\mathbf{D}$*, such that* $D_{11} = 1$ *(i.e., maintaining her full voting weight),* $\mathbf{D}_2 = (\dots, D_{22} = 0.5, \dots, D_{25} = 0.5)$*,* $\mathbf{D}_3 = (D_{31} = 0.5, \dots, D_{33} = 0.5, \dots)$*,* $D_{43} = 1$*, and* $D_{52} = 1$ *(recall Figure 5.1).*

By the limit weight approach, for the component consisting of agents $1, 3$ *and* 4*, since agent* 1 *is the only absorbing agent in the component (of the delegation graph), all weight in that component 'flows' to her in the limit, even though agent* 3 *has a loop with label of* 0.5*. Hence* $\mathring{w}(1, \mathbf{D}) = 3$*, while* $\mathring{w}(3, \mathbf{D}) = \mathring{w}(4, \mathbf{D}) = 0$*.*

For the component consisting of agents 2 *and* 5*, we observe that it is strongly connected and aperiodic because of the loop of agent* 2*. Then, in the limit, the weight transfers of agents* 2 *and* 5 *converge, and agent* 2 *always keeps half of the stabilized weight and delegates half to agent* 5*, while agent* 5 *accrues the weight delegated from agent* 2 *to agent* 5 *since this amount steadily "flows" through agent* 5*. That is, in the stationary weight distribution, agent* 2 *retains a weight amount twice of that retained by agent* 5*. Therefore,* $\mathring{w}(2, \mathbf{D}) = 2 \times 2/3 = 4/3$ *and* $\mathring{w}(5, \mathbf{D}) = 2/3$*. So* $\mathring{\mathbf{w}}(\mathbf{D}) = (3, 4/3, 0, 0, 2/3)$*. Unlike in the expected weight case (Example 18) there is therefore no weight loss in* $\mathring{\mathbf{w}}(\mathbf{D})$*.*

Due to the different weight distributions resulted from the above two weight approaches, we usually have different group accuracies. We can then modify Equation 5.3 by simply replacing the weight distribution of the expected weight approach by that of the limit weight approach, and obtain:

$$q_{N, \mathring{\mathbf{w}}(\mathbf{D})} = \sum_{C \in \mathcal{W}(\mathbf{D})} \prod_{i \in C} q_i \prod_{i \in Gu(\mathbf{D}) \setminus C} (1 - q_i), \tag{5.8}$$

where $\mathcal{W}(\mathbf{D})$ is the set of *winning coalitions*, such that

$$\mathcal{W}(\mathbf{D}) = \{C \subseteq Gu(\mathbf{D}) \mid \sum_{i \in C} \mathring{w}(i, \mathbf{D}) > \sum_{i \in Gu(\mathbf{D}) \setminus C} \mathring{w}(i, \mathbf{D})\}, \tag{5.9}$$

where ties are resolved uniformly at random.

We illustrate the difference between the two group accuracies by the following example.

Example 27 (Example 18, continued). *We continue to use Example 18 with accuracy profile* $\mathbf{q} = (0.9, 0.9, 0.6, 0.6, 0.6)$. *Unlike in Example 19, where we applied the expected weight approach, we apply the limit weight approach here. The stationary weight distribution is then* $\mathring{\mathbf{w}}(\mathbf{D}) = (3, 4/3, 0, 0, 2/3)$ *and the group accuracy is then* $q_{N, \mathring{\mathbf{w}}(\mathbf{D})} = 0.9$, *since agent* 1 *is the dictator (agent* 1 *alone is a winning coalition), and is therefore lower than that in the expected weight case.*

Intuitively, the difference in Example 26 is caused by two features:

1. In the component consisting of agents 2 and 5, by the expected weight approach, a part of the total weight is lost due to cycles. However, the component converges by the limit weight approach, and hence a positive amount of weight is accrued by each agent in the limit.

2. In the other component containing agents 1, 3 and 4, by the expected weight approach, agent 3 accrues part of the weight since she becomes a guru in some pure profiles supported by the weighted profile. But by the limit approach, all weight flows to agent 1, since she is the only absorbing node in that component.

It appears that cycles and loops are the main reasons for different weight distributions in the two approaches. In the following theorem, we specify sufficient conditions under which the two weight approaches coincide, leveraging the above observation.

Before showing the following theorem, we first introduce the concept of connected component into delegation graphs. In an undirected graph, a connected component is a connected subgraph which is not a part of any larger connected subgraph. Then, given a delegation graph (which is directed), we call a subgraph a *connected component* if in the undirected graph which eliminates all edges' directions in the delegation graph, the corresponding nodes and undirected edges form a connected component.

Theorem 16. *Let* $\mathbf{D}$ *be an arbitrary weighted profile. For any agent* $i \in N$, $\boldsymbol{t}_{\mathbf{D}}(i) = \mathring{\boldsymbol{t}}_{\mathbf{D}}(i)$ *if all connected components of the delegation graph satisfy the following two properties:*

1. *every cycle is only contained in a periodic strongly connected component, in which each agent is recurrent, and all agents linked to this strongly connected component are not linked to any other agent who has a loop; and*

2. *every loop has a label of* 1.

Proof. Without loss of generality, we assume that the delegation graph consists of only one connected component, otherwise we can apply the argument below to each connected component respectively.

We consider two exhaustive cases:

Case 1. the delegation graph does not contain any cycle; and

Case 2. the delegation graph contains cycles and satisfies the condition that any cycle is only contained in a periodic strongly connected component, in which each agent is recurrent, and all agents linked to this strongly connected component are not linked to any other agent who has a loop.

CASE 1. In this case, an agent is a guru if and only if she has a loop labeled 1, which means that the agent fully delegates to herself in both weight approaches. Then we only need to verify that for any agent $i_1 \in N$ and any delegation chain from this agent to any guru, the (expected) amount of weight transferred through this delegation chain is the same by both weight approaches, which immediately implies $\boldsymbol{t}_{\mathbf{D}}(i_1) = \mathring{\boldsymbol{t}}_{\mathbf{D}}(i_1)$.

Let $(i_1, \ldots, i_\ell)$ be a path from agent i_1 to guru i_ℓ. Then, by the expected weight approach, the expected amount of weight transferred from i_1 to i_ℓ is $\prod_{1 \leq j \leq \ell-1} D_{i_j i_{j+1}}$ through this path. This amount exactly equals the amount transferred from i_1 to i_ℓ in ℓ steps along the chain by the limit weight approach, since no agent in $\{i_1, \ldots, i_\ell\}$ has a loop. The weight transferred from i_1 to i_ℓ equals the entire weight transferred through all such paths, through each of which the amounts by both weight approaches coincide. Therefore, both weight approaches output the same weight transfer for all agents.

CASE 2. In this case, we first prove that the weight of all agents contained in cycles is lost. Since all cycles are contained in strongly connected components which are periodic and all agents in those strongly commented components are recurrent, we have that the weight delegated from any agent in any of the periodic strongly connected components is delegated back to the agent after a finite number (larger than one) of steps. Therefore, we can infer the following statements:

(a) No loop exists in the strongly connected component since the component is periodic, by the definition of periodicity.

(b) The strongly connected component has no out-degree, otherwise it violates that all agents in the component are recurrent: a certain amount of weight cannot be delegated back to agents due to the out-degree.

(c) The weight of all agents in the strongly connected component is lost in both weight approaches since:

- the weight transfer of any agent in the component is non-convergent by the limit weight approach;
- by the expected weight approach, all supported pure profiles contain only cycles, since no loop exists in the component.

Now, we show that any agent linked to such strongly connected component also loses her full weight. By the second condition stated in the theorem, any agent linked to such a strongly connected component is not linked to any agent with a loop. By (a) above, we have that

(d) by the limit weight approach, all the weight of this agent flows to strongly connected components which are periodic and all agents in those components are recurrent;

(e) by the expected weight approach, she is caught in a cycle in any induced pure profile.

Therefore, her full weight is lost.

It follows that in **Case 2**, we also obtain that the weight transfer of any agent is identical in both weight approaches, which completes the proof. □

With the above theorem, we immediately infer that Algorithm 6 and Algorithm 5 usually do not output weighted profiles with identical weight distribution by the two weight approaches, since all agents in N_1 have a loop with a label less than 1.

Corollary 1. *There exists* $\mathbf{D}$ *output by Algorithm 5 (respectively, Algorithm 6), such that* $\mathbf{w}(\mathbf{D}) \neq \mathring{\mathbf{w}}(\mathbf{D})$.

Hence these two algorithms fail to output the optimal weighted profiles for the limit weight approach.

5.3.4 Centralized Delegations in Connected Nets for The Limit Weight

We now propose another algorithm (Algorithm 7), which is able to construct delegation profiles, so as to achieve the optimal weight distribution $\mathbf{w}^\star$ as per Equation 5.5, for a subclass of connected underlying networks. The idea is to elaborately construct a delegation cycle, which converges by the limit weight approach and each agent maintains a stabilized amount of weight equal to the optimal amount in the limit. Recall that the results so far in the chapter are based on connected underlying networks, which are represented as undirected graphs. In this section, we will be working with the subclass of Hamiltonian graphs, which ensures the existence of a delegation cycle visiting each agent exactly once.

Definition 20 (Hamiltonian graphs [4])**.** *In an undirected graph, a cycle is called a Hamiltonian cycle if it visits each node exactly once. An undirected graph containing a Hamiltonian cycle is called a Hamiltonian graph.*

However, a Hamiltonian cycle is not guaranteed to exist in a connected undirected graph, and determining the existence of Hamiltonian cycles is NP-complete [4]. The following lemma provides a sufficient condition which guarantees that a Hamiltonian cycle always exists.

Lemma 8 (Balakrishnan & Ranganathan [4])**.** *If a connected undirected graph is the line graph of a Eulerian graph, a Hamiltonian cycle always exists, where a Eulerian graph is a connected undirected graph in which each node has even degree, and we construct the line graph $L(R)$ from an undirected graph R by*

1. *for each edge in R, making a node in $L(R)$; and*
2. *for each pair of edges in R that have a common node, making an edge between the corresponding nodes in $L(R)$.*

To introduce the algorithm, we further recall that, in a directed graph, a path L can be denoted as a sequence of nodes. Let L_i denote the i-th node on the path, and $|L|$ denote the number of nodes on L, where $1 \leq i \leq |L|$. Now, we are ready to introduce the algorithm outputting a weight profile with the optimal weight distribution in the limit weight approach.

In Algorithm 7, all agents are contained in one cycle. Notice that, by Line 2 of the **DELEGATE** component, unless all agents have identical optimal weight, at least one agent exists such that she has a loop. Therefore agents' weight transfers converge in the limit by the limit weight approach. Then in the stationary weight distribution, an amount of weight keeps "flowing" in the cycle (determined by the weighted delegation specified in Line 6 of the **DELEGATE** component), and for each agent, the stabilized weight equals this amount plus the amount she retains by her loop labeled in Line 7 of the **DELEGATE** component. We use the following example to show how Algorithm 7 works.

Example 28. *Consider three agents $N = \{1, 2, 3\}$, with accuracy profile $\mathbf{q} = (0.6, 0.6, 0.9)$, under the restriction of network R shown in Figure 5.7 left. By the* **DETERMINE THE CYCLE** *component, the cycle $L = (1, 2, 3, 1)$ is selected as shown by the arrows in Figure 5.7 right. Then, we compute the optimal weight distribution $\mathbf{w}^\star = (0.4044, 0.4044, 2.1912)$ (rounded), by Equation 5.5. Therefore, $w^\star_{min} = 0.4044$. We will construct a stationary weight distribution such that the amount of $w^\star_{min}$ keeps "flowing" in the delegation cycle stabilizedly. By Line 1 in the* **DELEGATE** *component, we have that $D_{1,2} = \frac{w^\star_{min}}{w^\star_1} = 1$, and similarly, $D_{2,3} = 1$ and $D_{3,1} = 0.1846$. By Line 7, we have that $D_{3,3} = 0.8154$, and therefore the algorithm outputs the following weighted profile:*

$$\mathbf{D} = \begin{pmatrix} 0, 1, 0 \\ 0, 0, 1 \\ 0.1846, 0, 0.8154 \end{pmatrix},$$

which induces $\mathbf{w}^\star$ in the limit weight approach.

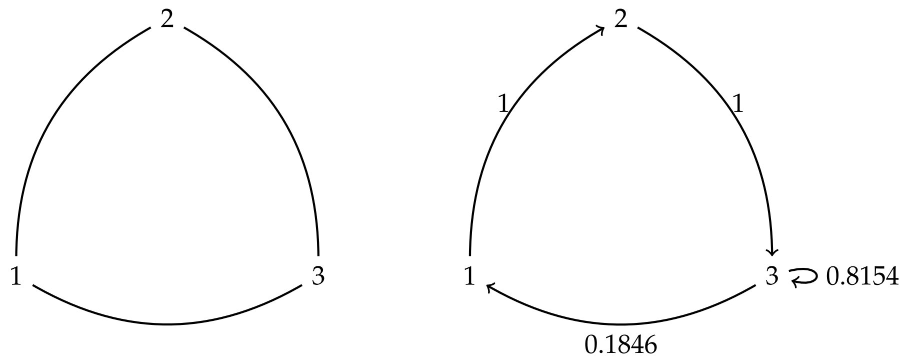


Figure 5.7: Figures in Example 28: the underlying network (left) and the weighted delegation graph output by Algorithm 7 (right).

We can then prove that by the limit weight approach, Algorithm 7 outputs a weighted delegation profile inducing the optimal weight distribution $\mathbf{w}^\star$.

Theorem 17. *Given a set of agents N, with accuracy profile $\mathbf{q}$, and a network R which satisfies the condition specified in Lemma 8, Algorithm 7 outputs a weighted delegation profile belonging to arg* $\max_{\mathbf{D}\in\mathbb{D}} q_{N,\mathring{\mathbf{w}}(\mathbf{D})}$.

Proof. First note that by Lemma 8, the **DETERMINE THE CYCLE** component can always select a delegation cycle L that visits each agent exactly once.

We then prove this theorem by showing first that the delegation graph of the output weighted profile converges, and then the stabilized weight distribution is identical to $\mathbf{w}^\star$.

We first consider the special case where each agent has the same accuracy. Then by Line 2 of the **DELEGATE** component $\mathring{w}(i) = 1 = w_i^\star$ for all $i \in N$.

We then consider the other general cases. By the **DETERMINE THE CYCLE** component in Algorithm 7, the delegation graph of the output weighted profile **D** consists of one cycle, which contains all agents in N. Moreover, by Line 7 of the **DELEGATE** component, there must exist at least one agent who has a loop. Therefore, by Lemma 7, all agents' weight transfer vectors converge since all agents are recurrent and the delegation graph is aperiodic due to the loops.

Therefore, in the stabilized status, an amount of weight keeps "flowing" through the delegation cycle. That is, at each time step in the limit, each agent receives an amount of delegation weight and delegates the same amount to her successor. Thus the stabilized weight of each agent is this "flowing" amount plus her retaining amount, i.e., the amount retained by her loop.

Assume the "flowing" amount is w. For agent $L_i \in N$,

$$\mathring{w}(i, \mathbf{D})\frac{w^\star_{min}}{w^\star_{L_i}} = w, \tag{5.10}$$

Algorithm 7 Optimal delegations in connected networks by limit weight approach

INPUT: $\mathbf{w}^\star$, $R = \langle N, E \rangle$ satisfying the condition in Lemma 8.

DETERMINE THE CYCLE:
Select an arbitrary path L which visits each agent in N exactly once and the first and the last agents in L coincide.

DELEGATE:

1: If $w_i^\star = w_j^\star$ for all $i, j \in N$:
2: $D_{L_i,L_i} = 1$ for all $i \in N$.
3: Else:
4: $w^\star_{min} = \min_{i \in N}\{w_1^\star, \ldots, w_n^\star\}$.
5: For $i \in [1, n]$:
6: $D_{L_i,L_{i+1}} = \frac{w^\star_{min}}{w^\star_{L_i}}$, where $L_{n+1} = L_1$.
7: $D_{L_i,L_i} = \frac{w^\star_{L_i} - w^\star_{min}}{w^\star_{L_i}}$.

RETURN: $\mathbf{D}$

i.e., the amount agent i delegates to her successor. We then have that

$$\mathring{w}(L_i, \mathbf{D}) = w \frac{w^\star_{L_i}}{w^\star_{min}}. \tag{5.11}$$

Moreover, the sum of the weights of all agents equals

$$\sum_{i=1}^{n} \mathring{w}(L_i, \mathbf{D}) = w \frac{\sum_{j=1}^{n} w^\star_{L_i}}{w^\star_{min}} = n, \tag{5.12}$$

where $\sum_{j=1}^{n} w^\star_{L_i} = n$. Therefore $w = w^\star_{min}$. Then by Equation 5.11, $\mathring{w}(L_i, \mathbf{D}) = w \frac{w^\star_{L_i}}{w^\star_{min}} = w^\star_{L_i}$ for agent $L_i \in N$ as desired. □

CONCLUSION

We studied a variant of liquid democracy with weighted proxies in this chapter. The theory of two different interpretations of weighted delegations has been developed. We showed that centralized delegations enable optimal group accuracy for both interpretations of weights refer to the theorems. We also identify one natural sufficient condition, under which the two interpretations coincide.

The work presented in this chapter relies on a connectedness assumption on the underlying network, which we aim at lifting in future work. Although we provided some results about the limit weight approach to weighted delegations, much more has to be understood about that setting.

In the next chapter, we adapt the theory of weighted delegations to the delegation games introduced in Section 2.4 and we study the existence and quality of equilibria. We will also provide further insights on these equilibria by way of computer simulations.

6 DELEGATION GAMES WITH WEIGHTED PROXIES

In this chapter, focusing on group accuracy, we contrast the centralized solution to optimize group accuracy that we described in the previous chapter with a decentralized approach to proxy weighting. To do so, like we did in Chapter 4, we extend the strategic model of liquid democracy developed in [11] to the setting involving weighted delegations, where agents greedily try to maximize their individual accuracies.

CHAPTER CONTRIBUTION We show that weighted delegations enable equilibria that are better in terms of group accuracy, with respect to equilibria with pure delegations (Theorem 18). This, however, comes at the cost of a higher price of anarchy with respect to games with pure delegations. We also show that the two interpretations of weighted delegations developed in Chapter 5 lead to different notions of utility in delegation games, and therefore to different equilibria. We prove the resulting notion of equilibrium of the limit weight approach (Section 5.3.2) to be weaker than that of the expected weight approach (Section 5.1.2) in Theorem 20.

Finally, to gain further insights into this game-theoretic model, we provide experimental evidence, via simulations, of the high truth-tracking performance of weighted delegations even in decentralized settings, if agents are boundedly rational according to a special model of bounded rationality known as quantal response equilibrium [56]. To the best of our knowledge, ours is the first application of this notion of equilibrium to liquid democracy. We observe that under specific conditions, weighted delegations can lead to higher group accuracy than simple majority voting.

6.1 DECENTRALIZED DELEGATIONS

In this section, we define the delegation games for the expected weight approach (Section 5.1.2) and we analyze the existence and quality of Nash equilibria. Then, in the next section, we alter the definition of delegation games for the limit weight approach (Section 5.3.2) and investigate the connection between the two definitions.

Algorithms 5 and 6 in Section 5.2 provide us with tractable (with time complexity polynomial in N) centralized mechanisms to achieve weighted delegation profiles that are optimal with respect to truth-tracking. Now we move to define a setting in which agents decide their delegations autonomously, assuming that they greedily aim at maximizing their own individual accuracy. We are interested in determining—analytically in this section, and empirically in Section 6.2—the effects of decentralized delegations on the truth-tracking performance of the group.

6.1.1 Weighted Delegation Games

Recall that, as described in Section 2.4, a delegation game for pure delegations has the property that any agent i may fully delegate to another agent or be a guru of herself, and her utility $u_i(\mathbf{d})$ is the accuracy she inherits from her guru, i.e., $u_i(\mathbf{d}) = q_{d^*(i)}$ whenever $d^*(i)$ exists. In weighted profiles, each agent may transfer weight to several gurus so the above setting can be extended by assigning to i a utility equal to the weighted average of the accuracies of i's gurus, weighted by the weights that i transfers to those gurus.

Formally, given a weighted profile $\mathbf{D}$ and its associated weight transfer profile $\boldsymbol{t}_{\mathbf{D}}(i)$ for agent i (recall Equation (5.1)), i's utility is given by:

$$U_i(\mathbf{D}) = \sum_{j \in N} q_j t_{\mathbf{D}}(i, j). \tag{6.1}$$

Observe that vector $\boldsymbol{t}_{\mathbf{D}}(i)$ can be interpreted as a probability distribution over i's gurus when none of i's weight is lost due to cycles. Equation (6.1) then gives us the expected individual accuracy of i in $\mathbf{D}$ or, in other words, the expectation $\mathbb{E}(u_i)$ over u_i given $\mathbf{D}$. Note also that in this section, we focus on the expected weight approach and in the next section, we replace the weight transfer $\boldsymbol{t}_{\mathbf{D}}(i)$ by that of the limit weight (Section 5.3.2) and study the delegation games with the other weight approach.

Equipped with the notion of utility, we move to define delegation games with weighted delegations.

Definition 21 (Delegation games with weighted delegations). *A delegation game with weighted delegations is a tuple $G = \langle N, R, S, U \rangle$, where $N = \{1, 2, \ldots, n\}$ is the set of agents, R is an undirected connected graph, $S = \{s_i\}_{i \in N}$ is the strategy space of each agent $i \in N$ where each element in each s_i is a weighted delegation as defined in Section 5.1.1, and U is the function defined by Equation* (6.1).

In the remainder of this chapter, we refer to delegation games with weighted delegations simply as delegation games.

Observe that, since U_i equals the expectation over u_i given the distribution over pure profiles induced by a weighted profile $\mathbf{D}$, the corresponding delegation game can be viewed as the mixed-strategy version of the delegation game with pure delegations. By Nash's theorem [61], we therefore know that such games always have Nash equilibria (NE), and we call such equilibria U-NE. We will also write $\mathcal{E}(G)$ to denote the set of all U-NE of G. It has already been shown that the pure delegation variant of these games (where each $\mathbf{D}_i$ is a degenerate probability vector) also always admits an NE [11], and we call such a pure strategy NE a u-NE.

An important feature of U-NE is that they do not contain any weighted delegation cycle[1]:

Lemma 9. *No U-NE contains a weighted delegation cycle.*

Proof. We reason towards a contradiction. Assume that $\mathbf{D}$ is a U-NE and it contains a weighted delegation cycle $(i_1, \dots, i_j, \dots, i_\ell, i_1)$. We show that there exists an agent in the cycle who has a better response, i.e., she obtains a better utility by deviating from the current weighted profile.

In the following reasoning, if an integer j is in the range $[\ell+1, 2\ell]$, let $i_j = i_{j-\ell}$, and if it is in the range $[-\ell+1, 0]$, let $i_j = i_{j+\ell}$. For any agent i_j in the delegation cycle ($1 \leq j \leq \ell$), we call $U_{i_j}(\mathbf{D}^e)$ the *external utility* of i_j, which is obtained by redistributing the weighted strategy $D_{i_j i_{j+1}}$ by the same proportion as her strategy $\mathbf{D}_{i_j}$ to *the other* neighbors $R(i_j) \setminus \{i_{j+1}\}$ out of the delegation cycle. Formally, for all $k \in R(i_j) \setminus \{i_{j+1}\}$,

$$\mathbf{D}^e_{i_j k} = \left(1 + \frac{D_{i_j i_{j+1}}}{1 - D_{i_j i_{j+1}}}\right) D_{i_j k}.$$

Assume that agent i_k ($1 \leq k \leq \ell$) is the agent in the delegation cycle who has the maximal external utility, i.e., for all i_j where $1 \leq j \leq \ell$, it holds that $U_{i_k}(\mathbf{D}^e) \geq U_{i_j}(\mathbf{D}^e)$. Then, if i_k deviates from the weighted profile $\mathbf{D}$ by taking the weighted delegation strategy $\mathbf{D}^e$, she obtains $U_{i_k}(\mathbf{D}^e)$. However, in $\mathbf{D}$, agent i_k obtains the following:

$$\begin{aligned}
U_{i_k}(\mathbf{D}) =& (1 - D_{i_k i_{k+1}}) U_{i_k}(\mathbf{D}^e) + D_{i_k i_{k+1}} (1 - D_{i_{k+1} i_{k+2}}) U_{i_{k+1}}(\mathbf{D}^e) + \\
& \cdots + D_{i_k i_{k+1}} \dots D_{i_{k-2} i_{k-1}} (1 - D_{i_{k-1} i_k}) U_{i_{k-1}}(\mathbf{D}(-i_{k-1} i_k)) \\
\leq& (1 - D_{i_k i_{k+1}}) U_{i_k}(\mathbf{D}^e) + D_{i_k i_{k+1}} (1 - D_{i_{k+1} i_{k+2}}) U_{i_k}(\mathbf{D}^e) + \\
& \cdots + D_{i_k i_{k+1}} \dots D_{i_{k-2} i_{k-1}} (1 - D_{i_{k-1} i_k}) U_{i_k}(\mathbf{D}^e) \\
=& (1 - D_{i_k i_{k+1}}) U_{i_k}(\mathbf{D}^e) + (D_{i_k i_{k+1}} (1 - D_{i_{k+1} i_{k+2}}) + D_{i_k i_{k+1}} D_{i_{k+1} i_{k+2}} \\
& (1 - D_{i_{k+2} i_{k+3}}) + \cdots + D_{i_k i_{k+1}} \dots D_{i_{k-2} i_{k-1}} (1 - D_{i_{k-1} i_k})) U_{i_k}(\mathbf{D}^e) \\
=& (1 - D_{i_k i_{k+1}}) U_{i_k}(\mathbf{D}^e) + (D_{i_k i_{k+1}} - D_{i_k i_{k+1}} \dots D_{i_{k-2} i_{k-1}} D_{i_{k-1} i_k}) U_{i_k}(\mathbf{D}^e) \\
=& (1 - D_{i_k i_{k+1}} \dots D_{i_{k-2} i_{k-1}} D_{i_{k-1} i_k}) U_{i_k}(\mathbf{D}^e).
\end{aligned}$$

1 Recall that a loop, i.e., a cycle with length 1, is not a delegation cycle (Section 2.1)

Since the delegation cycle $(i_1, i_2, \ldots, i_\ell, i_1)$ exists, we have that $D_{i_k i_{k+1}} \ldots D_{i_{k-2} i_{k-1}} D_{i_{k-1} i_k} > 0$, which leads to $U_{i_k}(\mathbf{D}) < U_{i_k}(\mathbf{D}^e)$.

Therefore, a contradiction is obtained. □

The above lemma indicates that there is no weight loss in U-NE. Therefore, we further obtain that all agents obtain the same utility in a U-NE with weighted delegation and pure delegation, since each agent delegates to the highest-accuracy agents accessible in the underlying network. This generalizes an observation made by [11] in the pure delegations case.

Corollary 2. *Given a delegation game G on N, let $\mathbf{D}$ be a U-NE and let $\mathbf{d}$ be a u-NE. Then, for all $i \in N$, it holds that $U_i(\mathbf{D}) = u_i(\mathbf{d})$.*

We illustrate the intuition by the following example.

Example 29. *Consider a delegation game G with four agents $N = \{1, 2, 3, 4\}$, accuracy profile $\mathbf{q} = (0.6, 0.8, 0.9, 0.9)$, and underlying network as shown in Figure 6.1 left.*

Figure 6.1 middle shows a pure-strategy Nash equilibrium

$$\mathbf{d}^* = (2, 3, 3, 4),$$

where agent 1 delegates to agent 2, agent 2 delegates to agent 3, and agents 3 and 4 are gurus. Therefore, agents 1 and 2 obtain the same utility, since they inherit the accuracy from the same guru, agent 3. In the other pure-strategy Nash equilibrium, the only difference is that agent 2 delegates to agent 4. In both equilibria, each agent obtains the same utility: $u_i(\mathbf{d}^*) = 0.9$ for all $i \in N$.

Figure 6.1 right shows a weighted Nash equilibrium

$$\mathbf{D}^* = \begin{pmatrix} 0,1,0,0 \\ 0,0,0.5,0.5 \\ 0,0,1,0 \\ 0,0,0,1 \end{pmatrix}.$$

Different from any of the pure-strategy Nash equilibria, in $\mathbf{D}^*$, agent 2 delegates half to agent 3 and the other half to agent 4. In any weighted Nash equilibrium, it is obvious that all voting weight is distributed between agents 3 and 4, and therefore all agents have agents 3 and 4 as their gurus. This implies that in any weighted Nash equilibrium, any agent obtains a utility of 0.9.

6.1.2 Group Accuracy in Equilibrium

So what is the truth-tracking quality of equilibria in weighted delegation games? We answer this question by first comparing group accuracy $q_{N,\mathbf{w}(\mathbf{D})}$ when $\mathbf{D}$ is a

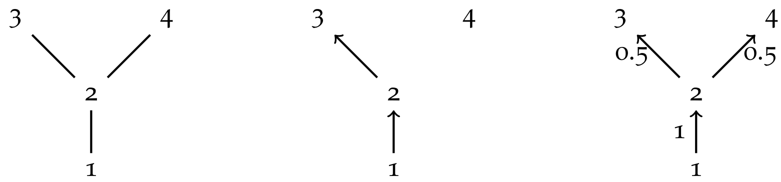


Figure 6.1: Figures in Example 29: left: the underlying network; middle: pure-strategy Nash equilibrium $\mathbf{d}^*$; right: weighted Nash equilibrium $\mathbf{D}^*$.

U-NE with $q_{N,\mathbf{d}}$ when $\mathbf{d}$ is a u-NE. We then establish a bound on how bad group accuracy may be in equilibrium using price of anarchy.

6.1.2.1 Weighted vs. Pure Equilibria

Weighted delegations make it possible to achieve higher group accuracy in equilibrium by balancing weight among maximally accurate agents. As a result, NE in delegation games with weighted profiles can be shown to be never worse than NE with pure delegations and to be better in some cases. Given a delegation game on N, let us denote with $N^* = \{i \in N \mid \forall j \in N, q_i \geq q_j\}$ the set of maximally accurate agents in N. Based on a weighted profile $\mathbf{D}$, for agent $i \in N$, let $(\mathbf{D}_{-i}, \mathbf{D}'_i)$ be the weighted profile where all agents take $\mathbf{D}$ except for i who takes the weighted delegation strategy $\mathbf{D}'_i$.

Theorem 18. *Any weighted profile $\mathbf{D}^*$ of a delegation game G such that for all $i \in N^*$ $w(i, \mathbf{D}^*) = \frac{n}{|N^*|}$ is a U-NE and $\mathbf{D}^* \in \arg\max_{\mathbf{D} \in \mathcal{E}(G)} q_{N,\mathbf{w}(\mathbf{D})}$.*

Proof. We first prove that $\mathbf{D}^*$ is a U-NE by a stronger claim, that if in a weighted profile $\mathbf{D}$, $\sum_{i \in N^*} w(i, \mathbf{D}) = n$, $\mathbf{D}$ is a U-NE. We show this by working towards a contradiction. Let q^* be the maximum accuracy, i.e., the accuracy of any agent in N^*. Assume that an agent $i \in N$ exists such that she obtains higher utility by taking $\mathbf{D}'_i$. That is, $U_i((\mathbf{D}_{-i}, \mathbf{D}'_i)) > U_i(\mathbf{D})$. Since $\sum_{i \in N^*} w(i, \mathbf{D}) = n$, all agents only have gurus in N^* in $\mathbf{D}$, and $U_i(\mathbf{D}) = q^*$. This also implies that all delegation chains point to one of the agents with accuracy q^*. Therefore by taking $\mathbf{D}'$, there are two possible cases: (1) agent i changes part of her delegations to other agents other than herself; (2) agent i changes part of her delegations from other agents to herself and hence she becomes one of her gurus.

In case (1), all delegation chains starting from i still point to agents with accuracy q^*, and then $U_i((\mathbf{D}_{-i}, \mathbf{D}'_i)) = q^*$. Contradiction.

In case (2), agent i becomes one of her own gurus, and her utility becomes $U_i((\mathbf{D}_{-i}, \mathbf{D}'_i)) = (1 - D'_{ii})q^* + D'_{ii}q_i$. Then, if $i \in N^*$, $U_i((\mathbf{D}_{-i}, \mathbf{D}'_i)) = U_i(\mathbf{D})$, but if $i \in N \setminus N^*$, $q_i < q^*$, which results in $U_i((\mathbf{D}_{-i}, \mathbf{D}'_i)) < U_i(\mathbf{D})$. Contradiction.

Then by Theorem 4, the U-NE $\mathbf{D}$ such that for all $i \in N^*$, $w(i, \mathbf{D}) = n/|N^*|$ optimizes group accuracy for N^*. Furthermore, by the Condorcet jury theorem

(Theorem 1), which states that larger group of agents (with homogeneous accuracy higher than 0.5) enhances group accuracy, no U-NE $\mathbf{D}$ in which $Gu(\mathbf{D})$ is a strict subset of N^* has higher group accuracy. □

The following example shows that there exist delegation games in which $\mathbf{D}^*$ has strictly better group accuracy than any equilibrium in pure delegation strategies.

Example 30. *Consider a delegation game where there are* 7 *agents,* 5 *of which have maximal accuracy* $q^* = 0.9$. *By Theorem 4 and Theorem 1, any pure delegation NE with maximal group accuracy would involve a pair of maximally accurate agents who both get a weight of* 2. *These two agents form a winning coalition, but they have a lower group accuracy than the remaining three gurus, i.e.,* $0.00081 = {q^*}^2(1-q^*)^3 < {q^*}^3(1-q^*)^2 = 0.00729$. *So the resulting group accuracy (recall Definition 2),* 0.98496, *is strictly worse than that of* $\mathbf{D}^*$, $q_{\mathbf{D}^*} = 0.99144$, *which is obtained by Theorem 18.*

6.1.2.2 Price of Anarchy

In this section, we define the price of anarchy introduced in Equation 2.14 by using the group accuracy as the social welfare and investigate the weighted version. We define the price of anarchy of a game G as:

$$\mathsf{PoA}(G) = \frac{\max_{\mathbf{D}\in\mathbb{D}} q_{N,\mathbf{w}(\mathbf{D})}}{\min_{\mathbf{D}\in\mathcal{E}(G)} q_{N,\mathbf{w}(\mathbf{D})}}. \tag{6.2}$$

When restricting to the price of anarchy in games with pure delegations (and therefore pure strategy u-NE), it gives rise to PoA^{pure} as defined in Equation 2.14.

So Equation 6.2 gives us a measure of how much group accuracy is 'lost' in equilibrium, in the worst case, with respect to what would be achievable via Algorithm 6 in Chapter 5.

Theorem 19. *When* $|N| \to \infty$, $\mathsf{PoA} \to \frac{1}{q^*}$, *where* q^* *is the accuracy of a maximally accurate agent in* N.

Proof. Let $\mathbf{D}$ be the weighted profile for which $q_{N,\mathbf{w}(\mathbf{D})}$ is maximal and let $\mathbf{D}'$ be the U-NE profile for which $q_{N,\mathbf{w}(\mathbf{D}')}$ is minimal. $\mathbf{D}'$ is the case when all agents delegate to the same guru, which has accuracy q^*. Then, since each $q_i \in (0.5, 1]$, by the law of large numbers as $|N| \to \infty$, $q_{N,\mathbf{w}(\mathbf{D})} \to 1$ and by construction $q_{N,\mathbf{w}(\mathbf{D}')} = q^*$. □

The same argument can be applied to the setting with pure delegations, obtaining the same asymptotic value for PoA^{pure}.

In the non-asymptotic case, since weighted delegations enable optimal group accuracy (Theorem 14) while pure delegations do not (Example 21), the PoA in delegation games with weighted delegations is trivially higher than that in the case of pure delegations:

Corollary 3. $\mathsf{PoA} \geq \mathsf{PoA}^{pure}$.

6.1.3 Delegation Games Under The Limit Weight Approach

Recall that in Section 5.3.2, agents' weight transfer in the limit is denoted as the n-dimensional vector $\mathring{\boldsymbol{t}}_{\mathbf{D}}(i)$. Similarly, we will refer to equilibria under the limit weight approach as $\mathring{U}$-NE, where the utility function is defined as:

$$\mathring{U}_i(\mathbf{D}) = \sum_{j \in N} q_j \mathring{\boldsymbol{t}}_{\mathbf{D}}(i,j), \tag{6.3}$$

by replacing the weight transfer in Equation 6.1 by $\mathring{\boldsymbol{t}}$.

We can then show that NE in the expected weight are also NE in the limit weight.

Theorem 20. *Given a delegation game G, if a weighted profile $\mathbf{D}$ is a U-NE, it is also a $\mathring{U}$-NE of G.*

Proof. First note that there is no delegation cycle in the $\mathbf{D}$, by Lemma 9. Moreover, since all weight in the network is concentrated on all highest-accuracy agents—assume this accuracy is q^*—and the underlying network is connected, by the proof of Theorem 18, all agents have the same utility, which is equal to q^* because their initial weight of 1 is all distributed among the highest-accuracy agents. Subsequently, we show in two possible cases: (i) all loops in $\mathbf{D}$ are with weight of 1; (ii) at least one loop, which is with weight less than 1, exists.

(i) This case satisfies the conditions in Theorem 16. Therefore, for all $i \in N$, $w(i, \mathbf{D}) = \mathring{w}(i, \mathbf{D})$. Thus, no agent has a profitable deviation, which indicates that $\mathbf{D}$ is an $\mathring{U}$-NE.

(ii) First notice that since no cycle exists, $\mathring{\boldsymbol{t}}_{\mathbf{D}}(i)$ converges for all $i \in N$, indicated by the proof of Theorem 16. Then, since $\mathbf{D}$ is a U-NE, any agent who has a loop with weight less than 1 must have accuracy q^*, by the proof of Theorem 18, and each of her out-degrees must end up at agents with accuracy q^*. By the limit weight approach, the weight "flows" to the absorbing agents among those highest-accuracy agents. That is, all agents inherit accuracy from those with q^*. Hence $\mathbf{D}$ is a $\mathring{U}$-NE. □

The other direction of the theorem does not hold, however, as shown by the following example:

Example 31. *Consider $N = \{1, 2\}$ with $\boldsymbol{q} = (0.9, 0.9)$ and $R = N^2$. Profile*

$$\mathbf{D} = \begin{pmatrix} 0.5, 0.5 \\ 1, 0 \end{pmatrix},$$

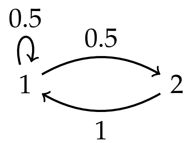


Figure 6.2: The $\mathring{U}$-NE in Example 31

as shown in Figure 6.2, is a $\mathring{U}$-NE. Since each agent is recurrent and the delegation graph is aperiodic, the weight transfers of both agents converge under the limit weight approach. Observe that in the limit, a stable flow of voting weight is retained by agent 2 *and agent* 1 *maintains the same extra amount besides the flow. Therefore, the converging voting weight distribution is* $\mathring{\mathbf{w}} = (\frac{4}{3}, \frac{2}{3})$, *and each agent still obtains the maximal utility of* 0.9. *However, the profile contains a cycle of weight* 0.5 *and cannot therefore be a U-NE by Lemma 9.*

6.2 EXPERIMENTS

The analytical results of the previous sections show that Nash equilibria with weighted delegations can be better in principle under the idealized rationality assumptions of Nash equilibria. We try to get a more fine-grained picture of group accuracy with weighted delegations using a weaker form of equilibrium incorporating a form of bounded rationality. In this section, we approach this aim by proceeding with a set of computational simulations under both the expected and the limit weight approach.

6.2.1 Experimental Setting

Agents are constrained in their delegations by a random network which will be treated as a parameter. Agents try to maximize their own utility under different weight approaches, namely, as per Equation 6.1 and Equation 6.3, respectively. However, they are assumed to be boundedly rational and achieve this maximization only imperfectly. To this aim, we model agents' strategic behavior with the so-called quantal response model [56], which has already been applied successfully to other strategic contexts in social choice (e.g., Meir [57], and an empirical study of storable voting by Casella *et al.* [17]).

LOGIT QUANTAL RESPONSE The quantal response model assumes that agents choose their strategies with noise. The probability (belief distribution) of choosing a pure delegation is positively related to the utility of that delegation, and agents respond to the others' strategies assuming that all agents have the same belief distribution, until an equilibrium is reached.

More precisely, we assume a special case of the quantal response model, known as *logit quantal response* (LQR).

Definition 22 (Logit Quantal Response). *Given weighted profile* $\mathbf{D}$*, an agent* $i \in N$ *responds in that profile by changing her individual weighted strategy to* $\mathbf{D}' = (\mathbf{D}'_i, \mathbf{D}_{-i})$*, such that for any neighbor* $j \in R(i)$*,*

$$D'_{ij} = \frac{e^{\lambda U_i(D_{ij}=1, \mathbf{D}_{-i})}}{\sum_{k \in R(i)} e^{\lambda U_i(D_{ik}=1, \mathbf{D}_{-i})}}, \tag{6.4}$$

where $(\mathbf{D}'_i, \mathbf{D}_{-i})$ *denotes the weighted profile in which each agent takes* $\mathbf{D}$ *except for agent i who takes* $\mathbf{D}'_i$*, and* λ *is a parameter indicating the error level to which agents are subject.*

Observe that, when $\lambda = 0$, for any agent $i \in N$, Equation 6.4 becomes $D'_{ij} = \frac{1}{\sum_{k \in R(i)} 1} = \frac{1}{|R(i)|}$, which corresponds to a uniformly random choice. However, as $\lambda \to \infty$, agents' choices approach optimality and the weighted profile approaches a pure-strategy Nash equilibrium [56]. Note that we replace $U_i(\mathbf{D})$ in Equation (6.4) by $\mathring{U}_i(\mathbf{D})$ for the limit weight approach.

Algorithm 8 Iterated LQR (ILQR)

INPUT: R, $\mathbf{q}$, $\mathbf{w}^\star$, λ, σ, $k^\star$.

INITIALIZE: $\mathbf{D}^0 : \forall i \in N, D^0_{ii} = 1$.

ROUND k $(k \geq 1)$:

1: For $1 \leq i \leq n$:
2: For $j \in R(\sigma(i))$:
3: $D^k_{\sigma(i)j} = \frac{e^{\lambda U_{\sigma(i)}(D_{\sigma(i)j}=1, \mathbf{D}^{k-1}_{-\sigma(i)})}}{\sum_{h \in R(\sigma(i))} e^{\lambda U_{\sigma(i)}(D_{\sigma(i)h}=1, \mathbf{D}^{k-1}_{-\sigma(i)})}}$.
4: If $k \geq k^\star$ or $\mathbf{D}^k = \mathbf{D}^{k-1}$:
5: Exit.

RETURN: $\mathbf{D}$

We implement an iterated LQR (ILQR) model (Algorithm 8) starting with the trivial profile $\mathbf{D}^0$ where for all $i \in N$, $D^0_{ii} = 1$. In the algorithm, besides the underlying network R and accuracy profile $\mathbf{q}$, we have three extra inputs: λ, the parameter for the LQR model, σ, a round-robin sequence permuting N, and $k^\star$, the maximum number of rounds. Note that we also apply the limit weight approach in the algorithm, where we replace U_i by $\mathring{U}_i$. Agents then apply LQR iteratively ordered by the sequence σ until no agent changes strategy (i.e., an equilibrium is found) or the number of rounds reaches the limit. By way of illustration, let agent 1 be the first agent responding to $\mathbf{D}^0$. Her response would be $\mathbf{D}^1 = (\mathbf{D}^1_1, \mathbf{D}^0_{-1})$, such that for all $j \in R(1)$, $D^1_{1j} = \frac{e^{\lambda q_j}}{\sum_{k \in R(1)} e^{\lambda q_k}}$, since for all $k \in R(1)$, $U_1(D_{1k} = 1, \mathbf{D}^0_{-1}) = q_k$. Then agent 2 responds to $\mathbf{D}^1$ by LQR, and so on until no agent changes her strategy any more, reaching a so-called LQR equilibrium. By [56, Th. 2], we know that as $\lambda \to \infty$, this LQR equilibrium converges to one of the Nash equilibria of the delegation game.

As a benchmark, we also implement a one-shot algorithm, called One-shot LQR (OLQR), which sets the parameter $k^\star$ in Algorithm 8 to 1. That is, in the one-shot algorithm, each agent in turn takes a logit quantal response only once.

Note that all agents' weight transfers converge in the limit weight approach, since agents have a loop in the weighted profiles output by the ILQR/OLQR. Therefore, no weight is lost in the limit weight.

6.2.2 Parameter Setting

The simulations are divided into two parts. In experiment A (Section 6.2.4.1), we consider different levels of density in an underlying random network (recall Section 2.1) by varying the probability of any two nodes being connected with $p \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. Then in experiment B, we consider different error levels in the LQR model by varying $\lambda \in \{0, 2, 4, 6, 8, 10, 20, 40, 60, 80, 100\}$. In both of the above parts, the two weight approaches are simulated: the expected weight model underpinning utility U_i of Equation 6.1, and the limit weight model underpinning utility $\mathring{U}_i$ of Equation 6.3. We fix λ to be 20 in experiment A and we fix p to be 0.9 in experiment B, due to the obvious trends of desired properties, e.g., the individual accuracy, the group accuracy, and delegation structure criteria, under these two settings.

6.2.3 Criteria

We study the effects of the above parameters on two properties of weighted delegation profiles: decision quality and delegation structures.

Decision quality is measured by two criteria, namely, group accuracy and average accuracy, which reflect the decision quality of weighted profiles from the levels of the individuals and the group, respectively. Group accuracy is defined in Equation (5.3). The average accuracy is simply the weighted mean of all gurus' accuracies, based on the weight distribution induced by the weighted profiles.

To investigate the delegation structures of weighted profiles, we study four criteria: the maximum and minimum individual weights in the weight distribution, the amount of weight lost in delegation cycles in the expected weight approach, and the Gini coefficient [36]. The Gini coefficient measures the equality of the weight distribution: the higher the index is, the more unequal the distribution is.

The main measures of decision quality and the structure of delegation graphs we focused on are the average accuracy, the group accuracy and the Gini coefficient. To determine whether the qualification of the described trends with respect to the variable on the x-axis (e.g., "increasing" and "decreasing") as well as the described difference between the algorithms and between the weight approaches are appropriate or not, we carry out statistical tests (ANOVA tests) for these three criteria (i.e., Figures 6.3a, 6.3b, 6.3c, 6.4a, 6.4b, and 6.4c). The details are provided in Appendix A.2.

SETUP We set $n = 30$. Agents' accuracies are independently drawn from the same Gaussian distribution ($\mu = 0.7$, $\sigma = 0.075$) and values are forced within the $[0.5, 1]$ range. For each parameter configuration we perform 50 runs to obtain our data. As group accuracy involves exponential-time computations, we estimate it via a Monte Carlo approximation sampling $2^{n-1}/100$, i.e., 5368709 times, random coalitions for each computation. Similar to our approaching, the Monte Carlo method is also used in [2] to compute the group accuracy in various voting methods. As for ILQR (Algorithm 8), we set for each parameter configuration the maximum number of iterations (i.e., $k^\star$) as 100. The experiments have been programmed in Python 3.7 and run on a CPU cluster of the University of Groningen[2] with 1GB memory.

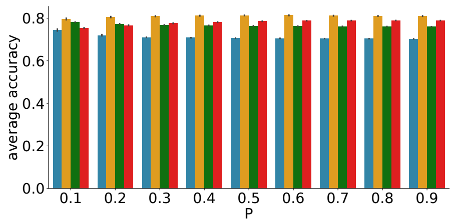
6.2.4 Findings

6.2.4.1 Experiment A

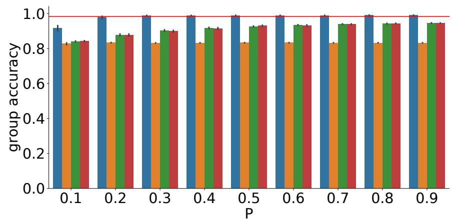
We first study the trends of average accuracy and group accuracy. As for average accuracy (Figure 6.3 (a)), in better connected underlying networks, agents have higher average accuracy in the expected weight approach (■ bars and ■ bars). The reason is straightforward: agents have access to higher-accuracy agents because of the better connectivity. However, in the limit weight approach (■ bars and ■ bars), agents' average accuracy becomes slightly lower as p grows. We conjecture that cycles appear more frequently in the delegation graphs in better connected networks. Thus voting weight is distributed more broadly in cycles by the limit weight, due to the fact that when a weight transfer converges in a delegation cycle, a stabilized amount of weight keeps "flowing" in the cycle. This then leads to lower average accuracy. Another interesting observation is that in the expected weight approach, ILQR (the yellow bars) outputs higher

2 `https://wiki.hpc.rug.nl/peregrine/start`

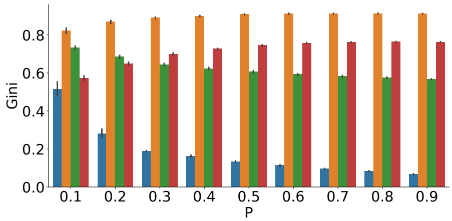
average accuracy than OLQR (the red bars), since agents tend to delegate to higher-accuracy gurus in the iterated response dynamics. This trend is inverse in



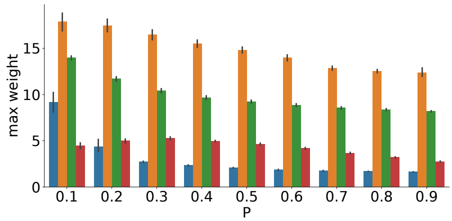
(a) A: average accuracy



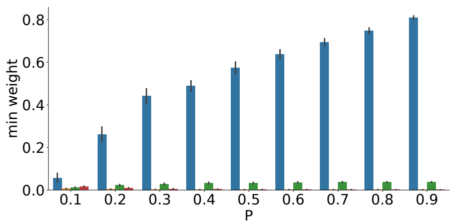
(b) A: group accuracy



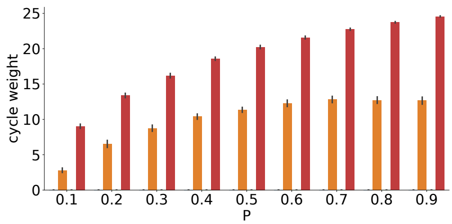
(c) A: Gini coefficient



(d) A: maximum weight



(e) A: minimum weight



(f) A: weight lost in cycles

Figure 6.3: Experiment A: fixed $\lambda = 20$ varying connection density of the underlying network p. Figures show the average accuracy, group accuracy, Gini coefficient, the maximum/minimum individual weight, and weight lost in delegation cycles. Response dynamics and weight approaches: ■ : ILQR with limit weight; ■ : ILQR with expected weight; ■ : OLQR with limit weight; ■ : OLQR with expected weight. The red line in Figure 6.3b denotes the group accuracy of the direct voting, i.e., the trivial delegation profile.

the limit weight, and we conjecture that this happens also because more cycles are formed by ILQR.

Then, as for group accuracy (Figure 6.3 (b)), as p increases, the trends of both dynamics and both weight approaches (weakly) increase since agents have a better chance to delegate to higher-accuracy agents. The group accuracies in the limit weight (■ bars) are (weakly) higher than that in the expected weight, i.e., for each p, the group accuracies of ILQR in the limit weight (■ bars) are significantly higher than those of ILQR in the expected weight (■ bars), and the group accuracies of OLQR in the limit weight are similar to those of OLQR in the expected weight (■ bars), with the group accuracies of ILQR in limit weight the highest, almost reaching 1. This is because the weight distribution in the limit weight is more balanced as shown by the Gini coefficient (Figure 6.3 (c)), where the values in the limit weight (■ bars and ■ bars) decrease as p increases, and between those the value of the ILQR (■ bars) is considerably lower than the others. This observation can be further supported by the maximum and minimum weight statistics (Figures 6.3 (d) and (e)): the individuals' weights of ILQR in the limit weight become roughly identical when $p = 0.9$, with a minimum of more than 0.8 and maximum around 1. We then conclude that in the limit weight, weight distribution is balanced, especially of the ILQR, because a large number of delegation cycles are formed. It is also worth observing that when $p \geq 0.3$, the group accuracy of ILQR in the limit weight approach (■ bars), which has the most balanced weight distribution, outperforms the high group accuracy of direct voting (the red line in Figure 6.3 (b)).

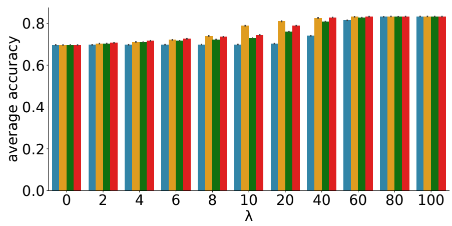
Another interesting finding is that, as shown above, in the limit weight, the weight distribution of ILQR is more balanced than that of OLQR, however, this is reversed for the expected weight approach, as shown in Figures 6.3 (c), (d) and (e). We conjecture that this is also due to delegation cycles. The limit weight approach tends to distribute weight more equally in cycles.

Finally, Figure 6.3 (f) shows that as the connectivity increases, more weight is lost in delegation cycles, in both ILQR and OLQR. Furthermore, ILQR (the red bars) causes increasingly more weight loss than OLQR (the orange bars). Given the fact that agents distribute weighted delegations relatively dispersively when $\lambda = 20$, better connected networks enhance the probability of forming delegation cycles, which aligns with the observed trends. Moreover, by ILQR, in the delegation graphs, more delegations tend to be concentrated on high-accuracy agents. This further increases the probability of forming delegation cycles, which is the reason of higher weight loss in ILQR.

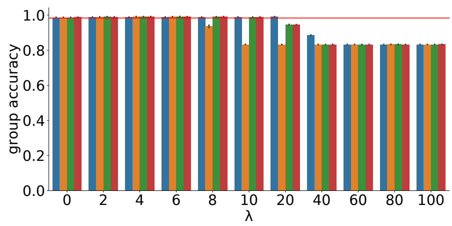
6.2.4.2 Experiment B

By Figures 6.4 (a) and (b), we observe that as agents' weighted delegations are more concentrated (i.e., larger λ), the average accuracies of both dynamics and weight approaches increase (from around 0.7 to more than 0.8), but this does harm to the group accuracy (from almost 1 to around 0.8). The trends of average accuracy are as expected: agents inherit higher accuracy when they delegate more weight to high-accuracy agents.

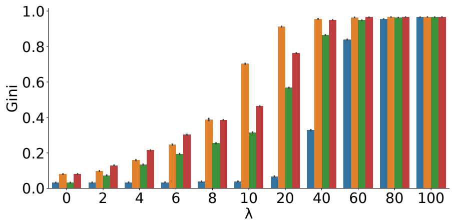
The results on group accuracy suggest that dispersively distributed weight improves group accuracy to some extent. We now investigate group accuracies in more details, combining them with the criteria concerning the delegation structure. For each value of λ, the group accuracy of ILQR in the limit weight (■ bars) is in general weakly higher than the others, and this trend corresponds to a (much) lower Gini coefficient of ILQR in the limit weight for most values of λ, as shown in Figure 6.4 (c). On the other hand, ILQR in the expected weight approach (■ bars) generally has the most unbalanced weight distribution (reflected in the



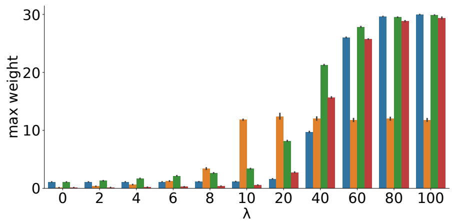
(a) B: average accuracy



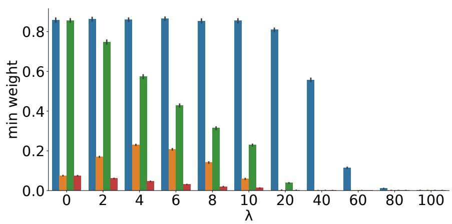
(b) B: group accuracy



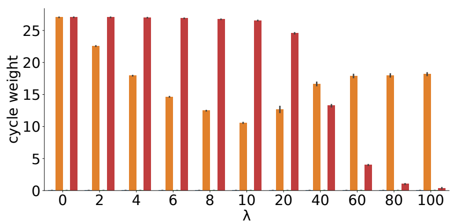
(c) B: Gini coefficient



(d) B: maximum weight



(e) B: minimum weight



(f) B: weight lost in cycles

Figure 6.4: Experiment B: fixed $p = 0.9$ varying the concentration parameter of the LQR model λ. Figures show the average accuracy, group accuracy, Gini coefficient, the maximum/minimum individual weight, and weight lost in delegation cycles. Response dynamics and weight approaches: ■ : ILQR with limit weight; ■ : ILQR with expected weight; ■ : OLQR with limit weight; ■ : OLQR with expected weight. The red line in Figure 6.3b denotes the group accuracy of the direct voting, i.e., the trivial delegation profile.

highest Gini coefficient) when $\lambda \geq 8$. This leads to its overall (weakly) lowest value of group accuracies. The above observations further support the observation that more balanced distributed weight may benefit group accuracy. Observe also that when agents distribute weighted delegations relatively equally among neighbors, i.e., when $\lambda \geq 10$, the group accuracy of most parameter settings outperforms that of direct voting (the red line in Figure 6.4 (b)).

It is also worth observing some features of the two weight approaches with respect to the delegation structure criteria. Similar to experiment A, the limit weight approach also outputs relatively more balanced weight distributions. This can be evidenced by comparing the Gini coefficient in the limit weight approach and that in the expected weight approach by both ILQR (bars and bars) and OLQR (bars and bars), as well as larger minimum weight in the limit weight (bars and bars in Figure 6.4 (e)). However, in the limit weight (bars and bars) high-accuracy agents may accrue more weight, especially when agents' delegations are more concentrated (e.g., $\lambda \geq 40$), as shown in Figure 6.4 (d). This is due to the fact that cycles are formed among high-accuracy agents and it becomes easier for those agents to retain weight in the limit weight: delegation cycles absorb weight and distribute the weight among agents in them.

Lastly, it is interesting to notice that by ILQR in the expected weight (bars), the amount of weight lost in cycles first decreases until $\lambda \geq 10$ and then rebounds (Figure 6.4 (f)). As agents uniformly distribute their weighted delegations among their neighbors (i.e., when $\lambda = 0$), cycles are formed with high probability. This problem of delegation cycles improves when agents start to concentrate their delegations to high-accuracy agents. However, when more delegations are concentrated on smaller groups of high-accuracy agents (when $\lambda \geq 10$), the weight loss again becomes more severe as agents in this group tend to form cycles easily as they have similar accuracies. But by OLQR (bars), many cycles are formed when agents dispersively distribute their delegations (when λ is small). Since they do not change strategies iteratively to avoid cycles as ILQR, the weight loss level keeps unchanged until $\lambda = 10$. The situation is relieved when agents turn to concentrate more delegations on the high-accuracy agents.

CONCLUSION

We incorporated the interpretations of weighted delegations developed in Chapter 5 in the delegation games developed in [11] and studied the features of NE. We showed that the group accuracy of NE with weighted delegation is never worse than that with pure delegation, which translates into a higher price of anarchy for delegation games with weighted delegations. To reveal the link between the two interpretations of weighted delegation, we showed that the NE in the expected weight are special cases of those in the limit weight. We finally complemented these findings with experimental observations showing how weighted delegations may boost group accuracy also in decentralized settings with boundedly rational agents.

Our experiments are based on one specific (and arguably fairly artificial) class of networks. A natural extension of our results would look into other classes of networks. Finally, analytical results about quantal response and group accuracy (e.g., price of anarchy with respect to quantal response equilibria) are worth pursuing.

7 CONCLUSIONS AND OUTLOOK

7.1 SUMMARY OF THE THESIS CONTRIBUTION

Liquid democracy is a young innovation in collective decision making. Much remains to be investigated. In this dissertation, we investigated liquid democracy from two angles: voting power and truth-tracking performance. We developed a method to formally measure the voting power in liquid democracy in Part II. This method provides foundations to understand power accrual in liquid democracy, including thorny issues such as the super voter problem of liquid democracy, i.e., a voting suffers from the risk of being manipulated by a small group of voters when they accrue a large number of votes. In Part III, we then contributed to the ongoing debate on the truth-tracking performance of liquid democracy, by proposing two models of weighted delegations. We have shown that this type of delegation mechanism may allow liquid democracy to achieve better truth-tracking performance than models from the literature.

We now revisit the contributions made in this dissertation in some more detail.

PART I In this part, we introduced the background of liquid democracy, based on which we developed the research in this dissertation. We provided a detailed technical basis in Chapter 2. Our work is based on the setting of collective decision making where a group of agents have to make a decision via binary truth-tracking voting. Agents may freely choose to directly vote or delegate their voting right, while their delegating interactions are restricted by a social network. Delegation cycles, namely, the structures in which no representative exists, may be formed, and all voting weight caught in such a structure is assumed to be equivalent to abstention. Under such a setting, we introduced the key toolbox for the study of power and truth tracking in binary voting. What follows is the introduction of the game-theoretic model of [11] on which we based a large part of our research. In this model, agents' utility is the individual accuracy they inherit from those on the end points of delegation chains. Lastly, we recaptured the existing negative results on liquid democracy's truth-tracking performance, against which we developed the weighted delegation theory to try to recover better truth-tracking performance for liquid democracy.

PART II In this part, we developed the theoretical and empirical study of voting power in liquid democracy.

In Chapter 3, we defined a novel voting power index, which measures agents' probability to influence the voting result in liquid democracy, for both gurus and delegators. This index strictly generalizes the Banzhaf power index to liquid democracy. An axiomatic characterization of the index was provided by extending an existing characterization of the standard Banzhaf power index with extra special properties characteristic of liquid democracy. We also showed that agents' voting power highly depends on the structure of the delegation graph. For example, along a delegation chain, agents closer to the guru have higher voting power, since their delegations are less likely to be redirected by other agents ahead. Another important feature of our index is that agents gain more voting power by obtaining more direct delegations than delegations via long delegation chains. This is because when agents accrue weight via long delegation chains, it is easier to lose a large amount of voting weight once some agent on the delegation chains changes her delegation strategy. However, such a risk is much lower for direct delegations.

According to the method in [49], the authors saw the voting activity of the gurus as a weighted voting and therefore studied the voting power of gurus only. In contrast, by our voting power index, both gurus and delegators have positive voting power, since delegators are also able to influence the voting result by changing their delegation strategies due to the instant recall component of liquid democracy. Moreover, delegators with a large number of direct delegations might have more voting power than gurus with fewer delegations accrued via long delegation chains.

In Chapter 4, we incorporated agents' incentives to retain the above voting power into the delegation games defined in Part I, and studied agents' behaviors both theoretically and empirically. We first showed that pure strategy Nash equilibria cannot be guaranteed to exist in delegation games in general. However, pure strategy Nash equilibria always exist in several subclasses of delegation games, such as in quota voting when the voting quota is less than half of the entire weight, in unanimity voting, and in voting instances for which the social network is complete.

We then bounded agents' rationality and studied their behavior by computational simulations. In the simulation algorithm, agents are assumed to iteratively choose a proxy randomly among those agents who are able to improve their utility. We showed that as the social networks are better connected, the average individual accuracy level is higher, but the voting power distribution becomes less equal. This is because agents have better access to high-accuracy agents, which results in the situation where a small number of high-accuracy agents accrue a large amount of voting weight. Then, when agents have more incentives to retain voting power, delegations tend to be prevented since agents lose voting power by delegating, especially via long delegation chains. As delegations happen less frequently, the voting power distribution becomes more equal, but the average individual accuracy level also decreases. Finally, we studied agents' behavior in different quota voting rules. We mainly observed trends in a benchmark algorithm, where agents in turn choose their best response only once, since the iterated algorithm has poor convergence on many quota settings. Agents tend to delegate more when the quota becomes higher, since agents are less sensitive to voting power as it is harder for agents to influence the voting result. As a result, the average individual accuracy level becomes higher while the voting power distribution becomes slightly less equal.

PART III In this part we mainly investigated the truth-tracking property of liquid democracy, and studied whether the performance is improved by allowing agents to split their voting weight and delegate to multiple proxies, namely, by weighted delegations as per our definition.

In Chapter 5, we provided two different interpretations of weighted delegations, namely, the expected weight approach and the limit weight approach, and we studied and compared the truth-tracking performance of the two interpretations.

In the expected weight approach, we considered weighted delegations as the probability that agents delegate to each neighbor. We thus computed each agent's accrued voting weight as the expected amount of her accrued weight in each possible pure delegation profile. We showed that the optimal voting weight distribution for truth tracking, where each agent's weight is proportional to $\log(\frac{q_i}{1-q_i})$, is achievable by coordinating agents' delegation strategies when the underlying network is connected.

In the limit weight approach, weighted delegations are interpreted as a direct split of voting weight, and this process can be modelled as a Markov process. We showed that these two weight approaches coincide when: (1) all delegators fully delegate their weight or, (2) each agent caught in a delegation cycle in some induced pure profile is caught in a delegation cycle in every induced pure profile. This is because, in the limit weight approach, the weight transfers in delegation cycles or non-full loops, i.e., loops with weight less than 1, may converge considerably differently from those in the expected weight approach. We further proved that the optimal voting weight distribution is also achievable when all agents are involved in an elaborately constructed delegation cycle and retain specific parts of delegations.

Then, in Chapter 6, we incorporated weighted delegations into the delegation games introduced in Part I and studied the Nash equilibria in both weighted delegation approaches. We showed that, even though a pure Nash equilibrium always exists, a weighted Nash equilibrium in the expected weight approach is never worse than a pure Nash equilibrium in terms of truth-tracking performance. As a consequence, weighted Nash equilibria come with a higher price of anarchy than pure Nash equilibria. We also showed that the set of the Nash equilibria in the expected weight approach is a subset of that in the limit weight approach.

Lastly, we empirically studied agents' behavior in delegation games with weighted delegations. In the experiments, we modelled agents' bounded rationality by the logit quantal response model [56], which had already been successfully applied to social choice scenarios. We observed that in better connected social networks, the truth-tracking performance can be weakly improved. However, better connectivity also results in more delegation cycles, such that more voting weight is lost.

In the second simulation study, we showed that as agents concentrate more delegations on high-accuracy agents, the average individual accuracy level becomes higher, with more unequal voting weight distribution. This, on the other hand, does harm the truth-tracking performance.

7.2 OUTLOOK

Our research can be extended in numerous directions. We sketch a few such directions below.

The research on voting power in liquid democracy in Part II moved from the observation that agents have much **flexibility to transfer and retain voting power** in liquid democracy, since agents' voting power may vary due to any change on the delegation graph. However, this also imposes enormous complexity for agents to arrange and manipulate delegation strategies in order to optimize various objectives. Future works can be developed in different directions for different optimization objectives. For example, we may consider two dichotomous types of agents in terms of decision-making quality: self-interested agents and far-sighted agents.

If agents are selfish and concentrate on their own decision-making quality, they aim at maximizing their individual accuracy, which gives rise to the game-theoretic model studied in Chapter 4. We showed in this chapter that pure strategy Nash equilibria are not guaranteed to exist in general, but they do exist in several subclasses of delegation games, e.g., delegation games with a complete underlying social network. However, based on this strong assumption of complete social network, our results leave out all the other classes of social networks. As the structure of social networks can significantly influence the behavior of agents, in future work, it is worth studying **agents' behavior in delegation games under other social network classes**. For example, in a star social network, the central agent may receive a large number of delegations even though her accuracy is at average level, since she has access to high-accuracy agents. However, as she accrues a large amount of voting weight, she might choose to be a guru as she would lose much voting power if she delegates. Such situations may also give rise to the voting power bribery problem as studied by D'Angelo *et al.* [25].

On the other hand, when agents' optimization objective is the overall collective decision-making quality, that is, collective truth tracking, then the setting is the one studied in Part III. In Part III, we showed that the optimal weight distribution with respect to truth tracking is achievable by liquid democracy. However, our results are based on strong centralized delegation mechanisms, which are not realistic in a context where decision making is distributed. To better understand the truth tracking performance of liquid democracy, in future work, we may **link truth-tracking performance with voting power measurement** in general delegation mechanisms. This link has been studied in the setting of weighted voting by Kalai [47], where the author showed that the group accuracy asymptotically converges to 1 if any agent's voting power is bounded. Therefore, in liquid democracy, in order to optimize the truth-tracking performance, agents do not only aim at enhancing the average individual accuracy level, but also try to restrict the voting power of the others, in order to achieve convergence of the group accuracy. For instance, delegators may choose more indirect delegations instead of direct ones in order to restrict the voting power of the gurus. Moreover, using liquid democracy as a method to increase individual accuracy, we may expect a better truth-tracking performance of it as compared to other voting methods, such as direct voting. Therefore, we could investigate the properties *Do No Harm* and *Positive Gain* defined by Kahng *et al.* [46] to compare liquid democracy and direct voting under the above aim.

Following up on the above directions of research, since liquid democracy captures the characteristic of direct voting and representative voting, we would also like to **compare liquid democracy with other benchmark representative voting methods** in terms of truth-tracking performance, e.g., elective representative voting and randomized representative voting. Traditional representative voting usually uses the elective representative voting method, where a set of representatives is elected by the voters. However, the randomized representative voting method, e.g., sortition [33], selects the set of representatives at random. In selective and randomized representative voting methods, the voting power of each representative depends on the size of their set. However, the individual accuracy level of the randomized voting method might be lower than that of the selective one. It is then interesting to conduct similar comparisons between liquid democracy and these two methods, taking into consideration individual accuracy and voting power.

Next, we propose a direction in terms of delegation mechanisms. In Chapter 5, we showed the possibility of optimal truth-tracking performance of liquid democracy. However, this is achieved by strong centralized delegation mechanisms. It is then a natural generalization in future work to **explore the possibility of decentralized delegation mechanisms**, or so-called local delegation mechanisms in [46]. When the underlying social network is complete, Algorithm 5 in Chapter 5 captures some features of decentralized delegation mechanisms: Each agent applies a specific delegation scheme according to the information of her neighbors. However, it is still challenging to verify the situations in which the network is not complete due to the complexity of social networks. For example, we conjecture that the optimal weight distribution (Theorem 4) is still possible by decentralized delegation mechanisms in some social networks which are not complete, even though we showed that no mechanism works in disconnected networks by Example 23. We would then like to extend our work to study in which classes of social networks the above conjecture still holds. Inspired by the experiments of agents with bounded rationality in Chapter 6, it would also be interesting to empirically study the performance of more decentralized delegation mechanisms in general underlying social networks.

Lastly, in Chapter 4, our experiments suffer from poor convergence of the iterated algorithm for several parameter settings of the quota rule. We would like to investigate the reason of such poor performance of the iterated algorithm under these settings of quota voting rules in the future work.

We hope that the theory developed in this dissertation will benefit follow-up research to better understand liquid democracy and settle crucial problems in this young voting method.

A APPENDIX

A.1 STATISTICAL TESTS FOR SECTION 4.3

In this section, we provide statistical tests for the main results of Section 4.3. Specifically, we conduct ANOVA tests for three criteria, including the ratio of delegators, the Gini coefficient of the DB's, and the average accuracy for Experiments A, B and C. That is, we conduct statistical tests corresponding to the results shown in Figures 4.9a, 4.9g, 4.9h, 4.10a, 4.10g, 4.10h, 4.11a, 4.11g, 4.11h. To test the significance of the results, we conduct two types of tests:

1. For each algorithm between IBRD and OSI (Algorithm 4 and the corresponding one-shot algorithm described in Section 4.3), in order to test the significance of the trends with respect to the variable on the x-axis, we conduct pairwise ANOVA tests. For example, for the trend of IBRD varying p shown in Figure 4.9a, we test whether the difference between each pair of adjacent values of p is significant or not, so as to determine whether the description of "increase" or "decrease" is valid or not.

2. For each parameter on the x axis, we test the significance of the difference between the two algorithms.

The details of the statistical tests are as follows.

Note that in the following tables (Table A.4, A.6, A.7 and A.9), the results of several tests are NANs. This is because both of the two data sets being tested have all elements identical, resulting in both within-group variances being zero. Therefore, the statistics cannot be computed (because the denominator is 0).

Based on the tests we performed, the trends observed are significant (by assuming that we reject the original hypothesis if p-value is less than 0.05).

A.1.1 Experiment A

p's	statistic	p-value
0.1 v.s. 0	11.918918918918914	0.0008218136713670221
0.2 v.s. 0.1	6.433893684688775	0.012775301225903556
0.3 v.s. 0.2	2.3007372914241366	0.1325318166017588
0.4 v.s. 0.3	0.8000666444451849	0.37326498425035703
0.5 v.s. 0.4	7.181455633100703	0.008641525849559915
0.6 v.s. 0.5	0.44822539333406515	0.5047531989663581
0.7 v.s. 0.6	1.6896551724137931	0.19669557387883027
0.8 v.s. 0.7	0.550561797752809	0.4598628737645788
0.9 v.s. 0.8	0.34937611408199637	0.5558273671731995

Table A.1: Pairwise statistical tests for the ratio of delegators for IBRD by varying p.

p's	statistic	p-value
0.1 v.s. 0	11.730179282214571	0.0008995284553868939
0.2 v.s. 0.1	6.4367780043339954	0.0127558789054175573
0.3 v.s. 0.2	2.584547911895945	0.11112840648084796
0.4 v.s. 0.3	0.5877655926437918	0.44512736657605234
0.5 v.s. 0.4	7.724114876302524	0.0065324518750908862
0.6 v.s. 0.5	0.531569230667027	0.46768675185554163
0.7 v.s. 0.6	1.3317506237704053	0.2513005631272605
0.8 v.s. 0.7	0.36310583302280339	0.548178213979232
0.9 v.s. 0.8	0.3077337232475393	0.5803380940959126

Table A.2: Pairwise statistical tests for the Gini coefficient for IBRD by varying p.

p's	statistic	p-value
0.1 v.s. 0	11.965276504636225	0.0008038057243390304
0.2 v.s. 0.1	6.854679386252455	0.010243526216546802
0.3 v.s. 0.2	2.3639008164150757	0.12739390805892525
0.4 v.s. 0.3	0.8085618814681148	0.3707499103444768
0.5 v.s. 0.4	7.18243724468699	0.008637128186298066
0.6 v.s. 0.5	0.37765448887627323	0.5402853653439399
0.7 v.s. 0.6	1.6950489215851 82	0.19598864202369062
0.8 v.s. 0.7	0.5055826751449164	0.47874629854651396
0.9 v.s. 0.8	0.3448670023432293	0.5583838578654481

Table A.3: Pairwise statistical tests for the average accuracy for IBRD by varying p.

p's	statistic	p-value
0.1 v.s. 0	30.032258064516128	3.298564922367793e-07
0.2 v.s. 0.1	19.253438113948913	2.88797118552975 74e-05
0.3 v.s. 0.2	3.9168026101141917	0.05061175709798265
0.4 v.s. 0.3	0.7	0.404819225222192 16
0.5 v.s. 0.4	0.33793103448275863	0.5623605782990677
0.6 v.s. 0.5	1.0	0.31977328750858 53
0.7 v.s. 0.6	NAN	NAN
0.8 v.s. 0.7	NAN	NAN
0.9 v.s. 0.8	NAN	NAN

Table A.4: Pairwise statistical tests for the ratio of delegators for OSI by varying p.

p's	statistic	p-value
0.1 v.s. 0	30.040425465888255	3.287953898429905e-07
0.2 v.s. 0.1	18.02066961227148	4.970078284646743e-05
0.3 v.s. 0.2	4.229841830630905	0.04237649128989634
0.4 v.s. 0.3	0.4420443383408732	0.5076989601212056
0.5 v.s. 0.4	0.32578632159690535	0.5694573564017935
0.6 v.s. 0.5	0.4543842868885774	0.5018471627511145
0.7 v.s. 0.6	1.1833514670099692	0.27934289101732485
0.8 v.s. 0.7	2.2000573202663993	0.14121456846635574
0.9 v.s. 0.8	1.4983702260051142	0.22385697055839684

Table A.5: Pairwise statistical tests for the Gini coefficient for OSI by varying p.

p's	statistic	p-value
0.1 v.s. 0	30.015646202162973	3.320255058834433e-07
0.2 v.s. 0.1	18.860630100504697	3.4306533372135195e-05
0.3 v.s. 0.2	3.9179188245321095	0.05057951231551284
0.4 v.s. 0.3	1.0019846454183021	0.3192959748130856
0.5 v.s. 0.4	0.428066761851765	0.5144716239619078
0.6 v.s. 0.5	1.069581887808635	0.30358528274843755
0.7 v.s. 0.6	0.9635314109999913	0.32871586799314567
0.8 v.s. 0.7	1.0	0.3197732875085853
0.9 v.s. 0.8	NAN	NAN

Table A.6: Pairwise statistical tests for the average accuracy for OSI by varying p.

p's	statistic	p-value
0	NAN	NAN
0.1	0.5178335535006606	0.4734799255965564
0.2	1.4082111436950149	0.23822236952511366
0.3	0.2969696969696969	0.5870264485335026
0.4	0.109131403311804014	0.7418403823693323
0.5	17.46391752577321	6.366403027828661e-05
0.6	23.261235955056193	5.1850551956509405e-06
0.7	64.89189189189197	1.9308335774011577e-12
0.8	33.92307692307693	7.290016682853988e-08
0.9	60.004237288135559	8.789059766348315e-12

Table A.7: Statistical tests for ratio of delegators of IBRD and OSI by varying parameter p.

p's	statistic	p-value
0	0.7478440102951036	0.389271586648035596
0.1	0.7798539095481827	0.37934698858583216
0.2	1.7212668436652159	0.19259498574959574
0.3	0.4794864498733933	0.4902925793821019
0.4	0.004657357555142044	0.9457297855079441
0.5	16.072127811109763	0.00011903935854757729
0.6	22.833095832097836	6.2086908526126 15e-06
0.7	58.1487607028792	1.582586966060409e-11
0.8	29.9731374079682	3.3764252045070473e-07
0.9	53.7445918336945 64	6.584828901679882e-11

Table A.8: Statistical tests for the Gini coefficient of IBRD and OSI by varying parameter p.

p's	statistic	p-value
0	NAN	NAN
0.1	0.7203742645224687	0.3980896274569464
0.2	1.5993768339191372	0.2089908034599571
0.3	0.397500632972938886	0.5298503474339566
0.4	0.033624437474190554	0.8548868195864499
0.5	15.90796189472944	0.00012824484660111063
0.6	20.636545361192265	1.5839295077049324e-05
0.7	59.96423284217482	8.900541825662807e-12
0.8	31.6224192098408	1.7690830629556046e-07
0.9	56.164148745545546	2.9929113841626645e-11

Table A.9: Statistical tests for the average accuracy of IBRD and OSI by varying parameter p.

A.1.2 Experiment B

α's	statistic	p-value
0.25 v.s. 0	1613.1019169329115	1.1410966349037825e-62
0.5 v.s. 0.25	32.77139066990937	1.1337831698857927e-07
0.75 v.s. 0.5	22.119898880462262	8.396142814199391e-06
1 v.s. 0.75	1440.3862227325008	2.1042210721459553e-60

Table A.10: Pairwise statistical tests for the ratio of delegators for IBRD by varying α.

α's	statistic	p-value
0.25 v.s. 0	676.6965125611798	8.6415272188374655e-46
0.5 v.s. 0.25	27.068993992308893	1.0783903149669859e-06
0.75 v.s. 0.5	53.519619543196214	7.090481461390791e-11
1 v.s. 0.75	856.9158691218703	3.0201228028622884e-50

Table A.11: Pairwise statistical tests for the Gini coefficient for IBRD by varying α.

α's	statistic	p-value
0.25 v.s. 0	8.136719239798605	0.005291487666277194
0.5 v.s. 0.25	14.12507065984858	0.00029075118135904135
0.75 v.s. 0.5	3.533161027491973	0.06312369549881953
1 v.s. 0.75	240.52855210588743	3.9590328497775264e-28

Table A.12: Pairwise statistical tests for the average accuracy for IBRD by varying α.

α's	statistic	p-value
0.25 v.s. 0	739.9830508474546	1.8331743853950136e-47
0.5 v.s. 0.25	4234.999999999967	1.894603306985394e-82
0.75 v.s. 0.5	3792.820224719053	3.704308065138286e-80
1 v.s. 0.75	1234.6882129277558	2.3979447275126244e-57

Table A.13: Pairwise statistical tests for the ratio of delegators for OSI by varying α.

α's	statistic	p-value
0.25 v.s. 0	17.97921228983098	5.0622941636590357e-05
0.5 v.s. 0.25	66.35260362687832	1.2388124134097925e-12
0.75 v.s. 0.5	817.5698292851629	2.3794628468316579e-49
1 v.s. 0.75	1275.08997805985	5.543600285026786e-58

Table A.14: Pairwise statistical tests for the Gini coefficient for OSI by varying α.

α's	statistic	p-value
0.25 v.s. 0	190.16437003973746	1.0989248854863765e-24
0.5 v.s. 0.25	1199.3319309661858	8.963843866473961e-57
0.75 v.s. 0.5	1502.650327223156	3.0075054146576734e-61
1 v.s. 0.75	278.24991499528915	2.1925867001575185e-30

Table A.15: Pairwise statistical tests for the average accuracy for OSI by varying α.

α's	statistic	p-value
0	3821.5778461538835	2.582013016420025e-80
0.25	93.1773444753946	6.863082090682601e-16
0.5	1218.293464858202	4.4001255412308975e-57
0.75	3351.0008771116057	1.3628792665779992e-77
1	0.29123328380386304	0.5906554066692713

Table A.16: Statistical tests for the ratio of delegators of IBRD and OSI by varying parameter α.

α's	statistic	p-value
0	736.0993673357708	2.3024457511937 31e-47
0.25	3.3033101029132097	0.07219613183411208
0.5	6.226704844088865	0.014254685115983864
0.75	613.3044071802934	5.702638127266943e-44
1	0.50643179732146 94	0.47837818621524963

Table A.17: Statistical tests for the Gini coefficient of IBRD and OSI by varying parameter α.

α's	statistic	p-value
0	16.715875604898663	8.900934579116132e-05
0.25	23.047085675421055	5.673490067469565e-06
0.5	72.94307111233813	1.7573262124657275e-13
0.75	206.57380897184 91	7.188418476763892e-26
1	0.37363055323950906	0.5424472041690601

Table A.18: Statistical tests for the Gini coefficient of IBRD and OSI by varying parameter α.

A.1.3 Experiment C

For the pairwise tests in Experiment C, we only provide those for the OSI, since there is no converged instance by IBRD for $\beta = 21$ and 24. Due to the same reason, we also only provide the tests of the two algorithms for $\beta = 21$ and 27.

α's	statistic	p-value
21 v.s. 18	2063.917372881346	1.1968749911276396e-67
24 v.s. 21	4156.822068965525	4.6245103615193 94e-82
27 v.s. 24	2345.329994107255	2.970848212572 09e-70

Table A.19: Pairwise statistical tests for the ratio of delegators for OSI by varying β.

α's	statistic	p-value
21 v.s. 18	97.931561963354586	2.0348273390122846e-16
24 v.s. 21	558.3545181227464	2.9503027687339763e-42
27 v.s. 24	7793.29667635564	3.2937122609340 69e-95

Table A.20: Pairwise statistical tests for the Gini coefficient for OSI by varying β.

α's	statistic	p-value
21 v.s. 18	1154.1033178980504	5.1080886633043 88e-56
24 v.s. 21	793.2771955165471	8.900797492660102e-49
27 v.s. 24	439.8067523413944	5.2159119054334 14e-38

Table A.21: Pairwise statistical tests for the average accuracy for OSI by varying β.

β's	statistic	p-value
18	2444.9085619284765	4.1922825964316 85e-71
27	2196.403688954098	6.4827660161423 79e-69

Table A.22: Statistical tests for the ratio of delegators of IBRD and OSI by varying parameter β.

β's	statistic	p-value
18	111.27444458657305	7.8274027964111 74e-18
27	81.62012880032862	1.5069916793304207e-14

Table A.23: Statistical tests for the Gini coefficient of IBRD and OSI by varying parameter β.

β's	statistic	p-value
18	68.71280449874074	6.099467693434161e-13
27	71.79587118786283	2.4549312750817777e-13

Table A.24: Statistical tests for the average accuracy of IBRD and OSI by varying parameter β.

A.2 STATISTICAL TESTS FOR SECTION 6.2

In the following, we provide the statistical tests for the significance of the difference of values shown in Section 6.2. Note that we skip the test results for some parameter settings which do not show obvious trends. For example, for the group accuracies shown in Figure 6.4b, when $\lambda \leq 6$ or $\lambda \geq 60$, the values do not show obvious trend. However, most information is revealed by the other parameter settings (i.e., when $\lambda = 8$, 10, 20 and 40). Therefore, we only provide test results for those important parameter settings. Note also that for some parameter settings, the p-values are rounded to 0.0 due to their smallness.

Based on the tests we performed, the trends observed are significant (by assuming that we reject the original hypothesis if p-value is less than 0.05).

A.2.1 Experiment A

For Experiment A, we conduct two types of statistical tests.

For each setting of weight approach and interaction algorithm (i.e., ILQR and OLQR), we test whether the trend is significant or not, when varying p. In order to verify this, we conduct statistical tests for all adjacent p's, for each setting of the other parameters, in Section A.2.1.1.

Then, for each p, we test whether the difference is significant between interaction algorithms and weight approaches, for each criterion, in Section A.2.1.2, by a two-way ANOVA test.

Additionally, we also test whether liquid democracy's group accuracy is significantly different than that of direct democracy for ILQR of the limit weight, i.e., the blue bars in Figure 6.3b.

We use ANOVA tests for all of the above statistical tests. These tests are conducted by Python on MacBook Pro.

A.2.1.1 Pairwise Tests for Different p'S

p's	statistic	p-value
0.2 v.s. 0.1	166.5397783634625	7.433703058956776e-23
0.3 v.s. 0.2	47.8638831087753	4.735223271012755e-10
0.4 v.s. 0.3	23.91453684963575	3.944431198328 98e-06
0.5 v.s. 0.4	13.334446409867066	0.00042045280122 77543
0.6 v.s. 0.5	13.023451894236535	0.00048661483953794276
0.7 v.s. 0.6	10.802011576795007	0.0014079003679302296
0.8 v.s. 0.7	4.955071590286428	0.028305339241508104
0.9 v.s. 0.8	8.427140275946762	0.004566795433766429

Table A.25: Pairwise statistical tests for the group accuracy of parameter p's, OLQR and the limit weight approach.

p's	statistic	p-value
0.2 v.s. 0.1	123.93567646066671	4.298462408652272e-19
0.3 v.s. 0.2	32.26303471147467	1.3797023715398688e-07
0.4 v.s. 0.3	20.689295619330842	1.548322326620974e-05
0.5 v.s. 0.4	28.35415703946293	6.42662262426275e-07
0.6 v.s. 0.5	0.93351056491011	0.33633066018148 84
0.7 v.s. 0.6	10.740277833718434	0.0014508020622763 93
0.8 v.s. 0.7	12.233685746340656	0.0007072317784877887
0.9 v.s. 0.8	1.6742302409643715	0.19873401257257492

Table A.26: Pairwise statistical tests for the group accuracy of parameter p's, OLQR and the expected weight approach.

p's	statistic	p-value
0.2 v.s. 0.1	56.717191311394615	2.5038481356604112e-11
0.3 v.s. 0.2	8.29838431663491	0.004874466046953932
0.4 v.s. 0.3	0.006674611013906675	0.9350532621761203
0.5 v.s. 0.4	1.8653165494527333	0.17513881903583361
0.6 v.s. 0.5	0.00038087393427298417	0.9844691593595195
0.7 v.s. 0.6	0.26097921461763046	0.6105971528259573
0.8 v.s. 0.7	3.8637350976478713	0.05217053400884316
0.9 v.s. 0.8	0.026657330433067485	0.8706415038655109

Table A.27: Pairwise statistical tests for the group accuracy of parameter p's, ILQR and the limit weight approach.

p's	statistic	p-value
0.2 v.s. 0.1	4.783725909893994	0.031108553248157 25
0.3 v.s. 0.2	8.195704857989204	0.005135200232083897
0.4 v.s. 0.3	0.9512097371190559	0.3318128859151591
0.5 v.s. 0.4	0.501177924823 07	0.4806633618783578
0.6 v.s. 0.5	0.0008437315234920584	0.9768861317449898
0.7 v.s. 0.6	0.4953103055772458	0.48323687424265604
0.8 v.s. 0.7	0.029534474249846666	0.8639051460797447
0.9 v.s. 0.8	0.16310147597541333	0.6871968891294142

Table A.28: Pairwise statistical tests for the group accuracy of parameter p's, ILQR and the expected weight approach.

p's	statistic	p-value
0.2 v.s. 0.1	84.63236141287486	6.608676972304395e-15
0.3 v.s. 0.2	65.84972557949106	1.4426433530836054e-12
0.4 v.s. 0.3	18.864788334799815	3.424392385075335e-05
0.5 v.s. 0.4	15.20312952170269	0.00017686807957719978
0.6 v.s. 0.5	14.018784554942242	0.0003054657264744267
0.7 v.s. 0.6	9.753995902738094	0.0023528696059906228
0.8 v.s. 0.7	7.03861422156798	0.009307010039519193
0.9 v.s. 0.8	17.244964179810474	7.020477223595789e-05

Table A.29: Pairwise statistical tests for the Gini coefficient of parameter p's, OLQR and the expected limit approach.

p's	statistic	p-value
0.2 v.s. 0.1	112.68126362303727	5.621006990832714e-18
0.3 v.s. 0.2	88.53053298879304	2.321104853184 18e-15
0.4 v.s. 0.3	49.25344499762507	2.9488125133694385e-10
0.5 v.s. 0.4	48.85027781685804	3.3815458751443366e-10
0.6 v.s. 0.5	40.22095493344449	7.002000403059658e-09
0.7 v.s. 0.6	8.525868061892409	0.004344560645718378
0.8 v.s. 0.7	8.28649608967821	0.004903937524124969
0.9 v.s. 0.8	4.586395714312812	0.034705904832439 59

Table A.30: Pairwise statistical tests for the Gini coefficient of parameter p's, OLQR and the expected expected approach.

p's	statistic	p-value
0.2 v.s. 0.1	108.8274627066418	1.399914031516551e-17
0.3 v.s. 0.2	70.66417364643337	3.421826831854012e-13
0.4 v.s. 0.3	55.331780435230684	3.919805993698668e-11
0.5 v.s. 0.4	78.98238905961016	3.138266821489785e-14
0.6 v.s. 0.5	51.27506161096944	1.492854178998599e-10
0.7 v.s. 0.6	82.73522873587588	1.1088479356446503e-14
0.8 v.s. 0.7	53.268460240173425	7.702047101572619e-11
0.9 v.s. 0.8	131.72982134207976	7.820561182843862e-20

Table A.31: Pairwise statistical tests for the Gini coefficient of parameter p's, ILQR and the limit approach.

p's	statistic	p-value
0.2 v.s. 0.1	34.43118657253161	6.0074882486393325e-08
0.3 v.s. 0.2	23.466302755746106	4.757637978436083e-06
0.4 v.s. 0.3	12.894134615075288	0.00051719466784891 86
0.5 v.s. 0.4	18.449120422044643	4.1120854888978577e-05
0.6 v.s. 0.5	3.3613607426648446	0.06977830651483 74
0.7 v.s. 0.6	0.020358528479130354	0.8868332419727378
0.8 v.s. 0.7	0.4399249269568965	0.5087158816116875
0.9 v.s. 0.8	0.016792359983213316	0.8971601034785985

Table A.32: Pairwise statistical tests for the Gini coefficient of parameter p's, ILQR and the expected approach.

p's	statistic	p-value
0.2 v.s. 0.1	180.33231346473858	6.075673865856836e-24
0.3 v.s. 0.2	65.91569746628959	1.4140619636420263e-12
0.4 v.s. 0.3	17.09680596569194	7.501808712162509e-05
0.5 v.s. 0.4	10.625548009693947	0.0015341516947878118
0.6 v.s. 0.5	9.898375759011046	0.0021910397388396835
0.7 v.s. 0.6	6.9730021384387066	0.009630389405240754
0.8 v.s. 0.7	3.1169030920828207	0.08059862756205427
0.9 v.s. 0.8	7.955601238144938	0.005802973419173339

Table A.33: Pairwise statistical tests for the average accuracy of parameter p's, OLQR and the limit approach.

p's	statistic	p-value
0.2 v.s. 0.1	353.6597616313316	2.7634758912104417e-34
0.3 v.s. 0.2	148.6966474065831	2.324936961628038e-21
0.4 v.s. 0.3	146.2012683872805	3.8386849544050825e-21
0.5 v.s. 0.4	97.13910710218056	2.4866934638756686e-16
0.6 v.s. 0.5	77.63082588778710	4.5900865474079884e-14
0.7 v.s. 0.6	93.860441533214340	5.752246681875576e-16
0.8 v.s. 0.7	127.41254590838	1.9951421071062741e-19
0.9 v.s. 0.8	171.79779903512284	2.8182645838852330e-23

Table A.34: Pairwise statistical tests for the average accuracy of parameter p's, OLQR and the expected approach.

p's	statistic	p-value
0.2 v.s. 0.1	73.26961841250122	1.5984856151942656e-13
0.3 v.s. 0.2	39.4879347958012	9.141994481330327e-09
0.4 v.s. 0.3	30.919074147187413	2.3279188812886616e-07
0.5 v.s. 0.4	31.132465351920946	2.1415136837388408e-07
0.6 v.s. 0.5	10.987807724636683	0.0012865043744678625
0.7 v.s. 0.6	22.621167947236035	6.789752524820291e-06
0.8 v.s. 0.7	3.3738484747454134	0.06926978697994199
0.9 v.s. 0.8	23.349704170845637	4.996039031258842e-06

Table A.35: Pairwise statistical tests for the average accuracy of parameter p's, ILQR and the limit approach.

p's	statistic	p-value
0.2 v.s. 0.1	111.2370821523629	7.896783011122896e-18
0.3 v.s. 0.2	37.6229968002405	1.8144059797531448e-08
0.4 v.s. 0.3	29.613508481152458	3.8922765733525847e-07
0.5 v.s. 0.4	12.681313245479357	0.0005718930729590695
0.6 v.s. 0.5	10.971241299729803	0.0012968762732144482
0.7 v.s. 0.6	3.587367672173392	0.0611693630966966606
0.8 v.s. 0.7	0.07036657370588842	0.7913609794590181
0.9 v.s. 0.8	0.022715038219193389	0.88051025617271112

Table A.36: Pairwise statistical tests for the average accuracy of parameter p's, ILQR and the expected approach.

p's	statistic	p-value
0.2 v.s. 0.1	111.2370821523629	7.89678301122896e-18
0.3 v.s. 0.2	37.6229968002405	1.8144059797531448e-08
0.4 v.s. 0.3	29.613508481152458	3.8922765733525847e-07
0.5 v.s. 0.4	12.681313245479357	0.0005718930729590695
0.6 v.s. 0.5	10.971241299729803	0.0012968762732144482
0.7 v.s. 0.6	3.587367672173392	0.061169363096696606
0.8 v.s. 0.7	0.07036657370588842	0.7913609794590181
0.9 v.s. 0.8	0.022715038219193 89	0.88051025617271112

Table A.37: Pairwise statistical tests for the average accuracy of parameter p's, ILQR and the expected approach.

A.2.1.2 Tests for Different Algorithms and Weight Approaches, for Each p.

p's	variable	statistic	p-value
0.1	algorithm	52.73161	8.784988e-12
	weight approach	110.451787	8.925789e-21
	mixed	118.650620	6.567179e-22
0.2	algorithm	127.186458	4.668018e-23
	weight approach	836.415126	1.212913e-72
	mixed	825.150574	3.558876e-72
0.3	algorithm	29.509790	1.637654e-07
	weight approach	2090.217695	1.667584e-106
	mixed	1917.742687	3.648821e-103
0.4	algorithm	12.491996	5.098308e-04
	weight approach	2612.265143	2.917188e-115
	mixed	2441.290496	1.378686e-112
0.5	algorithm	222.854880	3.708810e-34
	weight approach	4549.485397	1.348699e-137
	mixed	5004.602836	1.703522e-141
0.6	algorithm	378.657637	1.157994e-47
	weight approach	5018.6373182	1.308049e-141
	mixed	4837.393729	4.192957e-140
0.7	algorithm	1162.848184	2.393625e-84
	weight approach	9093.251194	3.458018e-166
	mixed	8892.110180	2.955479e-165
0.8	algorithm	2260.911194	1.432350e-109
	weight approach	14207.893565	7.383587e-185
	mixed	14496.652002	1.055409e-185
0.9	algorithm	3241.709061	7.154750e-124
	weight approach	17930.574118	1.212686e-194
	mixed	17431.716657	1.868712e-193

Table A.38: Statistical tests for the group accuracies of ILQR with the limit weight, ILQR with the expected weight, OLQR with the limit weight, and OLQR with the expected weight varying parameter p's.

p's	variable	statistic	p-value
0.1	algorithm	541.955203	2.486891e-58
	weight approach	1651.312367	1.992946e-97
	mixed	1165.829093	1.930694e-84
0.2	algorithm	340.393589	1.010321e-44
	weight approach	2817.187176	2.925816e-118
	mixed	992.476706	1.218906e-78
0.3	algorithm	474.194396	3.183433e-54
	weight approach	5682.474424	1.036829e-146
	mixed	1423.536894	8.012347e-92
0.4	algorithm	898.793089	3.838589e-75
	weight approach	12187.286488	2.003835e-178
	mixed	2463.631038	6.029831e-113
0.5	algorithm	1129.127000	2.815109e-83
	weight approach	15388.591031	3.272828e-188
	mixed	2893.024282	2.558289e-119
0.6	algorithm	967.197006	1.004518e-77
	weight approach	13925.673244	5.134683e-184
	mixed	2372.439288	1.844239e-111
0.7	algorithm	1321.791847	4.647890e-89
	weight approach	18453.299502	7.476472e-196
	mixed	3087.341597	6.465541e-122
0.8	algorithm	1769.834795	4.480329e-100
	weight approach	20467.562217	3.223241e-200
	mixed	3802.165859	2.659475e-130
0.9	algorithm	1565.684907	2.092782e-95
	weight approach	15830.048618	2.118387e-189
	mixed	3182.073690	3.977625e-123

Table A.39: Statistical tests for the average accuracies of ILQR with the limit weight, ILQR with the expected weight, OLQR with the limit weight, and OLQR with the expected weight varying parameter p's.

p's	variable	statistic	p-value
0.1	algorithm	2.063452	1.524623e-01
	weight approach	44.796822	2.242124e-10
	mixed	452.620561	7.914925e-53
0.2	algorithm	207.965985	1.308265e-32
	weight approach	1885.167734	1.673053e-102
	mixed	2420.964872	2.943655e-112
0.3	algorithm	2135.462763	2.441343e-107
	weight approach	17332.951787	3.241061e-193
	mixed	12658.610188	5.150890e-180
0.4	algorithm	3488.035864	8.097745e-127
	weight approach	29866.596857	3.557646e-216
	mixed	17018.659643	1.908830e-192
0.5	algorithm	5169.832784	7.941656e-143
	weight approach	45047.381401	1.421971e-233
	mixed	21812.327787	6.679522e-203
0.6	algorithm	10472.311504	4.435690e-172
	weight approach	92209.912243	5.731036e-264
	mixed	39757.359369	2.787318e-228
0.7	algorithm	14667.139183	3.407006e-186
	weight approach	128202.550126	5.725188e-278
	mixed	52517.384918	4.455589e-240
0.8	algorithm	23451.098818	5.860183e-206
	weight approach	205512.789094	4.986306e-298
	mixed	81151.045528	1.525375e-258
0.9	algorithm	55040.438054	4.560390e-242
	weight approach	485850.180509	0.0
	mixed	189644.046310	1.301870e-294

Table A.40: Statistical tests for the Gini coefficient of ILQR with the limit weight, ILQR with the expected weight, OLQR with the limit weight, and OLQR with the expected weight varying parameter p's.

A.2.1.3 *Tests for The Group Accuracy of Liquid Democracy and Direct Democracy, for ILQR of The Limit Weight*

p's	statistic	p-value
0.1	73.52540063100427	1.4843542321180185e-13)
0.2	0.7994393699767813	0.37345164698004174
0.3	213.55453015945778	2.3559655039091344e-26
0.4	252.3493374295719	7.3166421953031165e-29
0.5	320.84862465047166	1.124762850395545e-32
0.6	407.82889092931816	1.0588738498555122e-36
0.7	224.96724706664287	4.0103169492453565e-27
0.8	337.7445525333721	1.6106492090757784e-33
0.9	321.9059495743476	9.93679951173922e-33

Table A.41: Statistical tests for whether liquid democracy's group accuracy is significantly higher than that of direct democracy for ILQR of the limit weight, by varying p.

A.2.2 Experiment B

For Experiment B, we also conduct two types of statistical tests.

For each setting of weight approach and interaction algorithm (i.e., ILQR and OLQR), we test whether the trend is significant or not, when varying λ. In order to verify this, we conduct statistical tests for all adjacent λ's, for each setting of the other parameters, in Section A.2.2.1.

Then, for each λ, we test whether the difference is significant between interaction algorithms and weight approaches, for each criterion, in Section A.2.2.2, by a two-way ANOVA test.

We further test whether liquid democracy's group accuracy is significantly different from that of direct democracy when varying λ. We test for both ILQR and OLQR, and for both limit and expected weight approaches when $\lambda \leq 20$, since direct democracy's group accuracy is significantly higher when $\lambda \geq 40$.

We use ANOVA tests for all of the above statistical tests. These tests are conducted by Python on MacBook Pro.

A.2.2.1 *Pairwise Tests for Different λ's.*

λ's	statistic	p-value
2 v.s. 0	33.6974778670765 86	7.946101700252189e-08
4 v.s. 2	0.16819827976258037	0.6826132315605582
6 v.s. 4	0.3875272882046523	0.5350478645151406
8 v.s. 6	2.816772068870739	0.09647008253997547
10 v.s. 8	10.419220171091021	0.0016967025128810749
20 v.s. 10	2825.079596940522	4.532730211581297e-74
40 v.s. 20	19641.13288677056	1.0122329981378235e-114
60 v.s. 40	2.4185554876369604e-07	0.9996086095746533
80 v.s. 60	1.2763697938929606	0.26133277186465625
100 v.s. 80	0.2963536700150953	0.5874139674834l4

Table A.42: Pairwise statistical tests for the group accuracy of parameter λ's, OLQR and the limit approach.

λ's	statistic	p-value
2 v.s. 0	9.185425887162024	0.003120312782120451
4 v.s. 2	4.117874246953836	0.04514398791867437
6 v.s. 4	0.6963500810900743	0.4060424127271359
8 v.s. 6	0.8035971039030514	0.372216853110097
10 v.s. 8	8.740566493094823	0.003899117072624748
20 v.s. 10	2856.7192374997394	2.674099474469802e-74
40 v.s. 20	16963.36342498823	1.2818868402077l6e-111
60 v.s. 40	0.7036003374112373	0.4036179601536374
80 v.s. 60	0.45722157909118183	0.5005180436112836
100 v.s. 80	2.3512919571563335	0.12840129954422397

Table A.43: Pairwise statistical tests for the group accuracy of parameter λ's, OLQR and the expected approach.

λ's	statistic	p-value
2 v.s. 0	4.936579334842027	0.0285945623796507
4 v.s. 2	0.47639936927665755	0.49168918815236196
6 v.s. 4	1.5140779063848664	0.2214632300022637
8 v.s. 6	0.002523340638107158	0.9600390640834481
10 v.s. 8	2.870474348645114	0.09339413814244243
20 v.s. 10	1.997298263891961	0.16074829889427572
40 v.s. 20	8553.583053603601	3.6310930903052493e-97
60 v.s. 40	2184.2010361432144	8.419607483091502e-69
80 v.s. 60	2.152043678555343	0.1455821020803066
100 v.s. 80	0.9969393739349762	0.3205112362508542

Table A.44: Pairwise statistical tests for the group accuracy of parameter λ's, ILQR and the limit approach.

λ's	statistic	p-value
2 v.s. 0	14.995611962700002	0.00019452709360650832
4 v.s. 2	6.346968744471554	0.013375385219743944
6 v.s. 4	0.8712060117710392	0.3529156737152048
8 v.s. 6	266.86517574854065	9.934545741349914e-30
10 v.s. 8	950.1298026564227	3.1320426664060774e-52
20 v.s. 10	0.8128906484490336	0.36947752639071607
40 v.s. 20	0.5758440565459478	0.4497672301752814
60 v.s. 40	0.002691715980522462	0.9587285222456762
80 v.s. 60	0.49198797642733977	0.48470417782576625
100 v.s. 80	3.0399507200818996	0.08437441721337217

Table A.45: Pairwise statistical tests for the group accuracy of parameter λ's, ILQR and the expected approach.

λ's	statistic	p-value
2 v.s. 0	1517.041457414907	1.9391427766025834e-61
4 v.s. 2	2292.3401026467295	8.703151065741204e-70
6 v.s. 4	2172.3675115653855	1.086352014992814e-68
8 v.s. 6	2097.221933107771	5.656762473677674e-68
10 v.s. 8	1794.1756890771048	8.231836662736442e-65
20 v.s. 10	31612.73378001771	8.25747418088154e-125
40 v.s. 20	63973.74719816083	8.891208725709813e-140
60 v.s. 40	11313.095664279248	4.65684969697804e-103
80 v.s. 60	1616.1571758582281	1.0455288238422296e-62
100 v.s. 80	1008.7723402870596	2.168462711525211e-53

Table A.46: Pairwise statistical tests for the Gini coefficient of parameter λ's, OLQR and the limit approach.

λ's	statistic	p-value
2 v.s. 0	3172.386013585435	1.846789816800291e-76
4 v.s. 2	8030.789207076138	7.702123345344262e-96
6 v.s. 4	7346.919409379285	5.714474039735905e-94
8 v.s. 6	6502.32642187234	2.0882113419178038e-91
10 v.s. 8	4279.655207209466	1.1462458282947723e-82
20 v.s. 10	71882.01613617723	2.966208929060779e-142
40 v.s. 20	67576.31250130241	6.093074227619333e-141
60 v.s. 40	1963.794389943116	1.2235810341387132e-66
80 v.s. 60	86.2660046375037	4.250893194764263e-15
100 v.s. 80	0.009481826631807739	0.9226277308812322

Table A.47: Pairwise statistical tests for the Gini coefficient of parameter λ's, OLQR and the expected approach.

λ's	statistic	p-value
2 v.s. 0	1.321057633336573	0.2531996860021503
4 v.s. 2	0.3250725715656805	0.5698798800942273
6 v.s. 4	0.009831921588099427	0.9212169178594561
8 v.s. 6	21.6813340995141	1.0119258712255448e-05
10 v.s. 8	8.653968915224144e-05	0.9925965606279419
20 v.s. 10	649.8237118732212	4.8863411310102726e-45
40 v.s. 20	29121.19479909663	4.552687331014183e-123
60 v.s. 40	106857.06862829272	1.1082168947562314e-150
80 v.s. 60	18776.759503668647	9.08269388623569e-114
100 v.s. 80	12256.182841887934	9.508331822688943e-105

Table A.48: Pairwise statistical tests for the Gini coefficient of parameter λ's, ILQR and the limit approach.

λ's	statistic	p-value
2 v.s. 0	395.2663253537091	3.6419203317140286e-36
4 v.s. 2	3619.7999901908834	3.443221854981776e-79
6 v.s. 4	3750.394312702926	6.339954175106029e-80
8 v.s. 6	1979.2783921554467	8.478822008512142e-67
10 v.s. 8	10186.570545131302	7.58948409965021e-101
20 v.s. 10	34335.01518105878	1.4592366190495527e-126
40 v.s. 20	3749.4511680130536	6.416577125121589e-80
60 v.s. 40	328.2477530509702	4.7566008625515914e-33
80 v.s. 60	232.83157396987306	1.2275218795198537e-27
100 v.s. 80	206.46571918521704	7.3151358511182084e-26

Table A.49: Pairwise statistical tests for the Gini coefficient of parameter λ's, ILQR and the expected approach.

λ's	statistic	p-value
2 v.s. 0	2685.8229351892155	4.960001750637973e-73
4 v.s. 2	2441.0753598383403	4.514045385742642e-71
6 v.s. 4	2081.864477988122	7.98060747247188e-68
8 v.s. 6	1572.596287078475	3.6935984559733007e-62
10 v.s. 8	1517.4769890983464	1.913674720741711 8e-61
20 v.s. 10	25665.09103779997	2.1737390912583185e-120
40 v.s. 20	81997.6813851716	4.719062817809857e-145
60 v.s. 40	27533.7728083320464	7.0311148551621 37e-122
80 v.s. 60	3225.743936598063	8.35436610460555e-77
100 v.s. 80	1718.6791987840668	6.0597868015300 95e-64

Table A.50: Pairwise statistical tests for the average accuracy of parameter λ's, OLQR and the limit approach.

λ's	statistic	p-value
2 v.s. 0	19.749945778036597	2.3254857209021276e-05
4 v.s. 2	54.6901659061 64644	4.831010024066825e-11
6 v.s. 4	114.28925085575138	3.860441287142883e-18
8 v.s. 6	197.73075255465898	3.066571151345921 7e-25
10 v.s. 8	224.59325574105858	4.245628682084058e-27
20 v.s. 10	2857.6244949586317	2.634246990646779 7e-74
40 v.s. 20	17078.156653206734	9.228706047661161e-112
60 v.s. 40	12720.667658309472	1.5582186029268303e-105
80 v.s. 60	8178.623273717061	3.18438169104 75e-96
100 v.s. 80	201.5900396442289	1.619563929931945e-25

Table A.51: Pairwise statistical tests for the average accuracy of parameter λ's, OLQR and the expected approach.

λ's	statistic	p-value
2 v.s. 0	8.99150018672182	0.0034378606996870297
4 v.s. 2	10.542422350781557	0.001597616098971112
6 v.s. 4	29.168170354788174	4.644486704172686e-07
8 v.s. 6	8.219458012188372	0.005073624244997605
10 v.s. 8	11.082393634009161	0.0012289020981254473
20 v.s. 10	1120.200847028509	1.9629230138835463e-55
40 v.s. 20	46676.003516705816	4.421000106647259e-133
60 v.s. 40	153850.16517645994	1.9682534711657252e-158
80 v.s. 60	22219.018380916154	2.4720711766134 86e-117
100 v.s. 80	14239.210092481466	6.456371822660651e-108

Table A.52: Pairwise statistical tests for the average accuracy of parameter λ's, ILQR and the limit approach.

λ's	statistic	p-value
2 v.s. 0	77344.885188897237	8.2313603599995172e-144
4 v.s. 2	56599.36895531823	3.5569897003762766e-137
6 v.s. 4	29286.01498580673	3.4559070232639999e-123
8 v.s. 6	10457.691267331198	2.1204806076879408e-101
10 v.s. 8	12341.88320719503	6.776118143805385e-105
20 v.s. 10	40.550162196224456	6.214718321938043 4e-09
40 v.s. 20	173.569388282307 27	2.0413696104548 11e-23
60 v.s. 40	42.4494941523106	3.141355808629567e-09
80 v.s. 60	0.11614836185544074	0.7339795761001308
100 v.s. 80	1.4186805041551653	0.23649755799451747

Table A.53: Pairwise statistical tests for the average accuracy of parameter λ's, ILQR and the expected approach.

A.2.2.2 *Tests for Different Algorithms and Weight Approaches, for Each λ.*

λ's	variable	statistic	p-value
8	algorithm	279.164929	1.512060e-39
	weight	237.7201318	1.198384e-35
	mixed	249.286910	8.992955e-37
10	algorithm	37013.638638	2.973404e-225
	weight	37475.788408	8.869170e-226
	mixed	38100.799977	1.768204e-226
20	algorithm	3241.709061	7.154750e-124
	weight	17930.574118	1.212686e-194
	mixed	17431.716657	1.868712e-193
40	algorithm	1708.936310	9.803792e-99
	weight	1635.649555	4.593175e-97
	mixed	1720.197495	5.500323e-99

Table A.54: Statistical tests for the group accuracy of ILQR with the limit weight, ILQR with the expected weight, OLQR with the limit weight, and OLQR with the expected weight varying parameter λ's.

λ's	variable	statistic	p-value
2	algorithm	5.217394e+04	8.455049e-240
	weight	7.191158e+06	0.0
	mixed	6.609895e+04	7.796567e-250
4	algorithm	2.534262e+05	6.111730e-307
	weight	6.866635e+06	0.0
	mixed	3.199818e+05	7.381529e-317
6	algorithm	3.617392e+05	4.495997e-322
	weight	4.479112e+06	0.0
	mixed	4.643772e+05	0.0
8	algorithm	3.368646e+05	4.798366e-319
	weight	2.585992e+06	0.0
	mixed	4.438866e+05	0.0
10	algorithm	4.127015e+05	0.0
	weight	1.836149e+06	0.0
	mixed	5.506983e+05	0.0
20	algorithm	1565.684907	2.092782e-95
	weight	15830.048618	2.118387e-189
	mixed	3182.073690	3.977625e-123
40	algorithm	1447.483395	1.899464e-92
	weight	28361.0525665	5.469179e-214
	mixed	38.774471	2.838955e-09
60	algorithm	13362.585211	2.769964e-182
	weight	28713.455901	1.644006e-214
	mixed	11690.381508	1.109327e-176

Table A.55: Statistical tests for the average accuracy of ILQR with the limit weight, ILQR with the expected weight, OLQR with the limit weight, and OLQR with the expected weight varying parameter λ's.

λ's	variable	statistic	p-value
0	algorithm	0.035847	8.500281e-01
	weight	5116.991777	2.095054e-142
	mixed	0.004152	9.486911e-01
2	algorithm	2621.845334	2.089120e-115
	weight	7470.586359	5.139491e-158
	mixed	37.374589	5.176190e-09
4	algorithm	9832.592566	1.901413e-169
	weight	17098.559058	1.212474e-192
	mixed	813.530639	1.093925e-71
6	algorithm	15174.051023	1.273302e-187
	weight	33083.355277	1.676803e-220
	mixed	3532.666908	2.487169e-127
8	algorithm	4187.121639	3.246379e-134
	weight	21236.659919	8.971656e-202
	mixed	4373.542062	5.473334e-136
10	algorithm	372.990325	3.066431e-47
	weight	182467.784945	5.683862e-293
	mixed	73720.477641	1.820874e-254
20	algorithm	55040.438054	4.560390e-242
	weight	485850.180509	0.0
	mixed	189644.046310	1.301870e-294
40	algorithm	112213.473376	2.618440e-272
	weight	202595.982339	2.021084e-297
	mixed	117174.346335	3.801811e-274
60	algorithm	13521.116745	8.865201e-183
	weight	22917.629804	5.484089e-205
	mixed	13286.940104	4.793020e-182

Table A.56: Statistical tests for the Gini coefficient of ILQR with the limit weight, ILQR with the expected weight, OLQR with the limit weight, and OLQR with the expected weight varying parameter λ's.

A.2.2.3 *Tests for Whether Liquid Democracy's Group Accuracy Is Significantly Different from That of Direct Democracy When Varying λ.*

λ's	statistic	p-value
0	124.94858741590238	3.4328803768511157e-19
2	231.06564337119408	1.5973595059623873e-27
4	241.639184799514	3.3698156597007435e-28
6	232.70429964932276	1.2509841762114946e-27
8	309.7857815905298	4.192246971954736e-32
10	481.89793591507726	1.289405960085353e-39
20	321.9059495743476	9.93679951173392e-33

Table A.57: Statistical tests for whether liquid democracy's group accuracy is significantly different from that of direct democracy for ILQR of the limit weight approach by varying λ.

λ's	statistic	p-value
0	128.000176306112388	1.7544895587311982e-19
2	238.666419641131614	5.193320659867455e-28
4	420.7386848204703	3.070972124249652e-37
6	376.134583925767 97	2.542561885010698e-35
8	195.40324977666975	4.525122138081867e-25
10	94383.5282914068	4.82524231650703 24e-148
20	140224.50742225765	1.8458909513839925e-156

Table A.58: Statistical tests for whether liquid democracy's group accuracy is significantly different from that of direct democracy for ILQR of the expected weight approach by varying λ.

λ's	statistic	p-value
0	137.14343052631392	2.477508442951973e-20
2	358.752002141059	1.5930112628740303e-34
4	236.40461586600216	7.235718043275045e-28
6	371.26109947255884	4.2238859644490043e-35
8	388.6273807393792	7.086134710617155e-36
10	272.8769719295507	4.44752638643974e-30
20	2655.8686869445114	8.429210458735259e-73

Table A.59: Statistical tests for whether liquid democracy's group accuracy is significantly different from that of direct democracy for OLQR of the limit weight approach by varying λ.

λ's	statistic	p-value
0	147.26170082103727	3.1000247428827045e-21
2	310.45464676564734	3.8677729257736274e-32
4	409.6044303696373	8.9145489090137e-37
6	532.1341763247573	2.1826687933756497e-41
8	346.93550872185 97	5.7749247958316930-34
10	327.8113945334829	5.0021628184350130-33
20	2587.0223262118034	2.9158098169545693e-72

Table A.60: Statistical tests for whether liquid democracy's group accuracy is significantly different from that of direct democracy for OLQR of the expected weight approach by varying λ.

14. Brill, M. & Talmon, N. *Pairwise Liquid Democracy* in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18* (2018), 137–143.

15. Butterworth, G. & Booth, R. *A Contribution to the Defense of Liquid Democracy* 2022. `https://arxiv.org/abs/2206.13904`.

16. Caragiannis, I. & Micha, E. *A Contribution to the Critique of Liquid Democracy* in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19* (2019), 116–122.

17. Casella, A., Gelman, A. & Palfrey, T. R. An experimental study of storable votes. *Games and Economic Behavior* **57,** 123–154. ISSN: 0899-8256 (2006).

18. Center, P. R. Majority of public disapproves of supreme court's decision to overturn Roe v. Wade. *Report* (2022).

19. Chalkiadakis, G., Elkind, E. & Wooldridge, M. *Computational Aspects of Cooperative Game Theory* (Morgan & Claypool, 2012).

20. 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'17)* **251** (EPTCS, 2017), 134–150.

21. Cohensius, G., Mannor, S., Meir, R., Meirom, E. & Orda, A. *Proxy Voting for Better Outcomes* in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems* (2017), 858–866.

22. Colley, R., Delemazure, T. & Gilbert, H. *Measuring a Priori Voting Power - Taking Delegations Seriously* in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (2023), 2658–2660. ISBN: 9781450394321.

23. Colley, R. & Grandi, U. *Preserving consistency in multi-issue liquid democracy* in *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)* (2022).

24. Condorcet M.J.A.N. de C., M. d. *Essai sur l'Application de l'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix* (Imprimerie Royale, Paris, 1785).

25. D'Angelo, G., Delfaraz, E. & Gilbert, H. *Computation and Bribery of Voting Power in Delegative Simple Games* in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (2022), 336–344.

26. DeGroot, M. H. Reaching a Consensus. *Journal of the American Statistical Association* **69,** 118–121 (1974).

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

28. Doob, J. L. & Doob, J. L. *Stochastic processes* **2** (Wiley New York, 1953).

29. Dubey, P. & Shapley, L. S. Mathematical Properties of the Banzhaf Power Index. *Mathematics of Operations Research* **4,** 99–131. ISSN: 0364765X, 15265471 (1979).

30. Elkind, E. & Slinko, A. in *Handbook of Computational Social Choice* 169–196 (Cambridge University Press, 2016).

31. Escoffier, B., Gilbert, H. & Pass-Lanneau, A. *The Convergence of Iterative Delegations in Liquid Democracy in a Social Network* in *Proceedings of the International Symposium on Algorithmic Game Theory (SAGT'19)* (2019), 284–297.

32. Feld, S. L. & Grofman, B. The Accuracy of Group Majority Decisions in Groups with Added Members. *Public Choice* **42,** 273–285. ISSN: 00485829, 15737101. (2022) (1984).

33. Flanigan, B., Gölz, P., Gupta, A., Hennig, B. & Procaccia, A. D. Fair algorithms for selecting citizens' assemblies. *Nature* **596,** 548–552 (2021).

34. Ford, B. Delegative Voting. `http://www.brynosaurus.com/deleg/deleg.pdf` (2022).

35. French, J. A Formal Theory of Social Power. *Psychological Review* **61,** 181–194 (1956).

36. Gini, C. *Variabilità e mutabilità: contributo allo studio delle distribuzioni e delle relazioni statistiche.[Fasc. I.]* (Tipogr. di P. Cuppini, 1912).

37. Gölz, P., Kahng, A., Mackenzie, S. & Procaccia, A. D. *The Fluid Mechanics of Liquid Democracy* in *Proceedings of the 14th Conference on Web and Internet Economics (WINE'18)* (2018), 188–202.

38. Green-Armytage, J. Direct Democracy by Delegable Proxy. `http://web.archive.org/web/20050428192452/http://fc.antioch.edu/~james_green-armytage/vm/proxy.htm` (2005).

39. Green-Armytage, J. Direct voting and proxy voting. *Constitutional Political Economy* **26,** 190–220 (2015).

40. Grofman, B., Owen, G. & Feld, S. L. Thirteen theorems in search of the truth. *Theory and Decision* **15,** 261–278 (1983).

41. Halpern, D., Halpern, J. Y., Jadbabaie, A., Mossel, E., Procaccia, A. D. & Revel, M. In Defense of Liquid Democracy. *arXiv preprint arXiv:2107.11868* (2021).

42. Hu, X. & Shapley, L. On Authority Distributions in Organizations: Controls. *Games and Economic Behavior* **45,** 153–170 (2003).

43. Hu, X. & Shapley, L. On Authority Distributions in Organizations: Equilibrium. *Games and Economic Behavior* **45,** 132–152 (2003).

44. Jackson, M. O. *Social and Economic Networks.* (Princeton University Press, 2008).

45. Jain, P., Sornat, K. & Talmon, N. *Preserving Consistency for Liquid Knapsack Voting* in *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems* (2021), 1542–1544. ISBN: 9781450383073.

46. Kahng, A., Mackenzie, S. & Procaccia, A. *Liquid Democracy: An Algorithmic Perspective* in *Proc. 32nd AAAI Conference on Artificial Intelligence (AAAI'18)* (2018).

47. Kalai, G. Social Indeterminacy. *Econometrica* **72,** 1565–1581. ISSN: 00129682, 14680262. (2023) (2004).

48. Kavitha, T., Király, T., Matuschke, J., Schlotter, I. & Schmidt-Kraepelin, U. *Popular Branchings and Their Dual Certificates* in *Integer Programming and Combinatorial Optimization* (Springer International Publishing, Cham, 2020), 223–237. ISBN: 978-3-030-45771-6.

49. Kling, C. C., Kunegis, J., Hartmann, H., Strohmaier, M. & Staab, S. *Voting Behaviour and Power in Online Democracy: A Study of LiquidFeedback in Germany's Pirate Party* in *Ninth International AAAI Conference on Web and Social Media* (2015).

50. Kotsialou, G. & Riley, L. *Incentivising Participation in Liquid Democracy with Breadth-First Delegation* in *Proceedings of the International Foundation for Autonomous Agents and Multiagent Systems (AAMAS '20)* (2020), 638–644. ISBN: 9781450375184.

51. Koutsoupias, E. & Papadimitriou, C. *Worst-Case Equilibria* in *STACS 99* (eds Meinel, C. & Tison, S.) (Springer, 1999), 404–413. ISBN: 978-3-540-49116-3.

52. Lanphier, R. A Model for Electronic Democracy? `http://web.archive.org/web/20051027051842/http://robla.net/1996/steward/` (1995).

53. Lehrer, E. An axiomatization of the Banzhaf value. *International Journal of Game Theory* **17,** 89–99 (1988).

54. Matsui, T. & Matsui, Y. A survey of algorithms for calculating power indices of weighted majority games. *Journal of the Operations Research Society of Japan* **43,** 71–86 (2000).

55. Matsui, Y. & Matsui, T. NP-completeness for calculating power indices of weighted majority games. *Theoretical Computer Science* **263.** Combinatorics and Computer Science, 305–310. ISSN: 0304-3975 (2001).

56. McKelvey, R. & Palfrey, T. Quantal Response Equilibria for Normal Form Games. *Games and Economic Behavior* **10,** 6–38 (1995).

57. Meir, R. *Strategic Voting* (Morgan & Claypool, 2018).

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

59. Nisan, N., Roughgarden, T., Tardos, E. & Vazirani, V. V. *Algorithmic game theory* (Cambridge university press, 2007).

60. Nowak, A. S. On an axiomatization of the banzhaf value without the additivity axiom. *International Journal of Game Theory* **26,** 137–141 (1997).

61. Osborne, M. J. & Rubinstein, A. *A Course in Game Theory* (MIT Press, 1994).

62. Paulin, A. *An Overview of Ten Years of Liquid Democracy Research* in *Proceedings of the 21st International Conference on Digital Government Research* (2020).

63. Penrose, L. The Elementary Statistics of Majority Voting. *Journal of the Royal Statistical Society* **109,** 53–57 (1946).

64. Shapley, L. & Shubik, M. A method for Evaluating the Distribution of Power in a Committee System. *American Political Science Review* **48,** 787–792 (1954).

65. Shapley, L. S. & Grofman, B. Optimizing group judgmental accuracy in the presence of interdependencies. *Public Choice* **43,** 329–343 (1984).

66. Tullock, G. Proportional representation. *Toward a mathematics of politics,* 144–157 (1967).

67. Tullock, G. Computerizing politics. *Mathematical and Computer Modelling* **16,** 59–65 (1992).

68. Zhang, Y. & Grossi, D. Power in Liquid Democracy. *Proceedings of the AAAI Conference on Artificial Intelligence* **35,** 5822–5830 (2021).

69. Zhang, Y. & Grossi, D. *Tracking Truth by Weighting Proxies in Liquid Democracy* in *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems* (2022), 1482–1490. ISBN: 9781450392136.

positive perspective on the issue of so-called supervoters in liquid democracy, because power concentration does not appear to occur under our assumptions.

To address question (ii), we provide two weighted delegation models that represent how voters split their votes and delegate to multiple delegatees: one is probabilistic and models such behavior as mixed strategies, i.e., distributions on the space of possible delegations; the other one models such behavior as a split of votes viewed as shares to different delegatees. Compared to the pure delegation setting (i.e., each voter can only delegate to one delegatee), we demonstrate that it is possible to optimize the decision-making quality through the weighted delegation scheme. However, this requires centralized coordination.

We then investigate, both theoretically and empirically, voters' behavior in delegation games where weighted delegation is allowed. The results show that the Nash equilibria in weighted delegations, that is, delegation structures where no agent has an incentive to change their delegation, are always weakly better in terms of decision-making quality than those in pure delegation. However, this comes with a higher price of anarchy, i.e., the fraction between the optimal welfare and the welfare of the worst equilibrium. Empirically, our simulations show that when voters are boundedly rational, weighted delegation reaches a better decision-making quality than pure delegation.

Overall, in this dissertation, we study liquid democracy, this young collective-decision-making method, in terms of the above two questions. We contribute formal methods to analyze it, and show liquid democracy's potential to enhance the quality of collective decisions.

de stemkracht van een kiezer in vloeibare democratie wordt niet alleen bepaald door het aantal verzamelde delegaties, maar ook door hoe zij delegaties precies verzamelt: directe delegaties versterken degenen die de delegaties ontvangen beter dan indirecte.

Vervolgens vinden we, met behulp van methoden uit de speltheorie, dat kiezers die gevoelig zijn voor macht de neiging hebben om niet te veel te delegeren, vooral niet via indirecte delegatie, om hun invloed op het stemprobleem te behouden. Deze resultaten bieden een positievere kijk op het vraagstuk van de zogenaamde 'superkiezers' in vloeibare democratie, omdat machtsconcentratie onder onze aannames niet lijkt voor te komen.

Om vraag (ii) aan te pakken, bieden we twee gewogen delegatiemodellen, die weergeven hoe kiezers hun stemmen opsplitsen en delegeren aan meerdere gedelegeerden: de ene is probabilistisch en modelleert zulk gedrag als gemengde strategieën, dat wil zeggen, verdelingen in de ruimte van mogelijke delegaties; de andere modelleert zulk gedrag als een opsplitsing van stemmen die wordt beschouwd als aandelen bij verschillende gedelegeerden. Vergeleken met de context van pure delegatie (waarbij elke kiezer slechts kan delegeren aan één gedelegeerde), laten we zien dat het mogelijk is om de besluitvormingskwaliteit te optimaliseren door middel van het gewogen delegatieschema. Echter, dit vereist gecentraliseerde coördinatie.

We onderzoeken vervolgens het gedrag van kiezers in delegatiespellen waarbij gewogen delegatie is toegestaan, met zowel theoretische als empirische methoden. De resultaten tonen aan dat in de context van gewogen degelatie de Nash-evenwichten, dat wil zeggen, de delegatiestructuren waarin geen agent een prikkel heeft om zijn delegatie te veranderen, altijd minstens zo goed zijn qua besluitvormingskwaliteit als de Nash-evenwichten in de context van pure delegatie. Echter, dit gaat gepaard met een hogere prijs van anarchie, dat wil zeggen, het verschil tussen de optimale welvaart en de welvaart van het slechtste evenwicht. Empirisch laten onze simulaties zien dat wanneer kiezers begrensd rationeel zijn, gewogen delegatie leidt tot een betere besluitvormingskwaliteit dan pure delegatie.

Over het geheel genomen bestuderen we in dit proefschrift vloeibare democratie, de jonge methode voor collectieve besluitvorming, met betrekking tot de bovengenoemde twee vragen. We dragen formele methoden bij om het te analyseren en laten zien dat vloeibare democratie daadwerkelijk het potentieel heeft om de kwaliteit van collectieve beslissingen te verbeteren.