Livos
Liquid-Federated Voting System

École Polytechnique Fédérale de Lausanne

LIVOS
a Liquid Voting System

by Etienne Boisson and Guillaume Tabard

Bachelor Project Report

Prof. Bryan Alexander Ford
Project Advisor

Noémien Kocher
Project Supervisor

EPFL IC INFCOM DEDIS
BC 233 (Bâtiment BC)
Station 14
CH-1015 Lausanne

January 6, 2022

Contents

"Modern European democracies are completely technologically outdated. In France we still have to drive to distant physical voting booths, only to fill out a paper ballot while we could have secure and convenient online vote system."

Acknowledgments

Thank you to everyone, special thanks to Noémien Kocher for the supervision and the precious help you brought us. Thank you Mr.Ford for the opportunity and for your time. Thank you to those who spent time answering our survey.

Lausanne, January 6, 2022 Etienne Boisson and Guillaume Tabard

Abstract

LIVOS stands for Liquid Voting System, this is a research project based upon Liquid Democracy consisting in creating an e-voting system with the concepts of liquidity of vote and delegation. The main goal of the project is to show concrete numbers that comfort the fact that Liquid Democracy is more accurate in taking into account the voters' opinions than any other non-liquid democratic system and also to provide an intractable system where simulations can be run. This is indeed a proof of concept based upon tangible experiments.

Liquid Democracy is a subject that has already been deeply explored in theory, but hasn't been compared to other systems to be used in our every-day life. In practice, the Liquid Democracy is quite different, we need to solve Liquid Democracy related problems (for example the circular delegation or the tyrant problems, that we explore and for which we offer a possible solution). Then it needs to be compared to an actual working democratic system such as Direct Democracy or Representative Democracy in order to show if Liquid Democracy is more accurate, represents more efficiently the voters' opinion and thus may influence our leaders into rethinking our actual political systems. We always have to keep in mind that Liquid Democracy is a complex structure that needs to be easily accessible, understandable and usable to be incorporated in our everyday life.

To explore the different paths of the Liquid Democracy, we implemented a web application and an environment where simulations can be run. That allowed us to work with collected data in order to show the relevance, the problems and the challenges of Liquid Democracy.

Chapter 1

Introduction

1.1 Context and definitions

First of all let's define the terms that compose LIVOS (LI-quid VO-ting S-ystem).

Liquid means that you can split your vote: in other terms imagine you have a glass of water and two Candidates A and B to vote for. Your preference goes to A and you want to give him 60% of your voting power leaving 40% to B. The voting power is a reference to the total amount of water you have in your glass, in our project we choose it to be 100 as it is easier to manipulate.

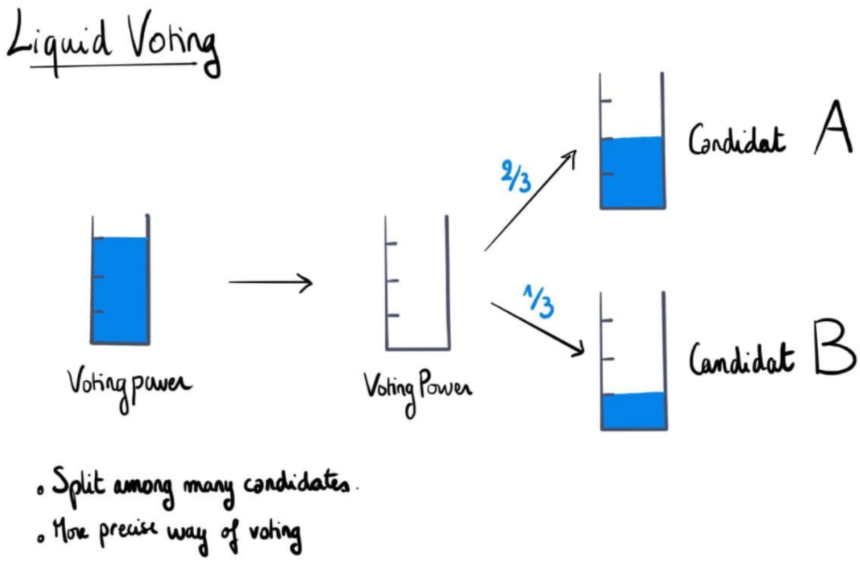


Illustration of the liquidity of the system

This system also comes with the possibility of vote delegation : it's the combination of Representative and Direct Democracy. The idea is that people can cast a vote themselves or if they don't have time, are not interested in the subject or are not confident in the given "votation"'s field ("votation" is a Swiss word that we transposed into an English version), they can easily delegate their vote to people they deem more experienced, more knowledgeable... Then we have the Voting system, which you can interact with through a website. This design allows us to interact with the votes and to conduct parts of our experiments. We want to explore the possibility of an E-voting system based upon liquid principles (delegation, liquidity of the votes) and to highlight interesting aspects, namely usability, accessibility.

1.2 Usefulness of the project

If we search on the Internet, we find many different projects linked to e-voting and Liquid Democracy; for instance, there is Adhocracy [1] (belonging to a platform that makes virtual democracy more accessible) : you only need to register on a website, and join "channels" in order to participate in the vote. Digital Liquid Democracy is another project aiming at understand the barriers that prevents the UK from moving forward in the direction of e-voting and offers an e-voting system at Lancaster University [10].

There are also theoretical researches such as Bryan Ford Paper [3], for example we can find a complete discussion about the different liquid systems as Quadratic Voting. An important section is devoted to transferable voting, with a complete argumentation about Instant Runoff voting, single transferable vote, cumulative or quadratic transferable vote. It develops the pros and cons of each system and expands the theory behind them.

A lot of paths, advantages, drawbacks have been explored but people need to be educated about the fact that Liquid Democracy exists, works and needs to have an interface to interact with.

In the end, Liquid Democracy is the democracy model fit for today's society. The technology is ready, what is missing is more effort into working on real implementations researching this area in more details. Most importantly, we need to determine which models are applicable for the actual governance of a country (directive or administrative). The urge of having an implementation of this Liquid Democracy system is a huge strength of our study.

1.3 Goal of the project

This project consists in implementing a proof of concept of a liquid democratic system. One of the goals is to analyse and discuss the different results, counting methods (such as Liquid Quadratic Voting or Markov's chain resolution) and the parameters of liquid democracy (the possibility to delegate votes and adjusting different parameters such as the voting power of the minorities, the weight of the delegated votes and so on) and to offer the fairest possible system. Indeed in theory, the liquidity and delegation should offer a better view of the opinions among the population, as participation should be increased with the redistribution of the votes and each vote should be as close as possible to the voter's opinion.

With this implementation of Liquid Democracy, with various parameters such as the type of voter, the type of question (Referendum or Election), the way of counting the results (normal way, Liquid Quadratic Voting or Markovien way), Liquid Democracy could be imported in our every-day life, at least on the school ground.

In the end, we also build a basis for future studies. There is already a master project that has been offered for next semester that will continue our work.

1.4 Problematic

It comes with many interrogations: How much weight does a delegated vote have? What happens if you delegate your vote to Bob and Bob is not confident enough and thus decides to delegate back to you (circular voting)? How to balanced, in case of a minority, their voting power and the way it could influence the vote ? For example, how to moderate the weight of their votes so that they will not impose their ideas to the majority? What if everybody delegates his vote ?

And finally, could Liquid Democracy be better than other non Liquid systems such as Direct Democracy?

1.5 Challenges

At the beginning of the semester we had mighty thoughts about the project, great ambition with three main axes: federation, security and liquidity. It rapidly appeared that it was too much.

Therefore a big challenge was to restart on week 4-5 almost a new version of the project, because the amount of work needed to accomplish this would have been like the twelve labors of Hercules. So we chose to focus on implementing the liquid voting system with delegation, the website application, the possibility of running simulations and on some research about liquid democracy and it's problems and challenges.

In the end, a major challenge was the implementation of the liquid democracy. It's a very complex structure with a lot of possibilities for the voter: delegate, vote (either a referendum or a candidate election: the big difference is that in a referendum you cannot vote for the two opposite sides, whereas for the candidate you can split your vote to different candidates) and partial vote and delegate. This leads us to circular delegation problems, rounding problems. Simulating the behavior of a voter was also an issue as it gets more and more complex the closer we get to reality. This section is going to be more developed in the Design choice.

1.6 Results of the simulation

Through our simulations we have seen that in specific situations a Liquid System could be up to 20% more accurate than without liquidity (only in the more realistic setups that we defined). In general, with few voters (< 100), the liquid system is 5-6% more precise than a traditional system. In most of our simulation, the additional liquidity precision tends to fade as the numbers of voters grow.

If you would like to see the code of the project, it's here : LIVOS Github

Chapter 2

Design

Individual decision is very important in a project because it allows us to quickly bypass, solve a problem. In our case, the fact that we were two to work on the project allowed us to split tasks but also to discuss on different implementation choices for the simulation (we discussed the different possible behaviours of the users for example), or design for the website (the parameter page), anything about the code or the orientation of our project. It's easier to take decision when you can directly talk to someone that understands quickly your questions and elaborates with you a correct answer. In the end, thanks to all the feedback received during our semester presentations, we were able to self-guide into the "right" direction.

2.1 The project at the beginning

Since the federated part creates institutions, entities that you can delegate to (for example you can delegate your vote to the EPFL directly), it creates discussions and debates at each group level. It should ensure participation for everyone and should lead to better, enlightened decision making.

2.1.1 Federation

But we didn't implement the federated part even if it's an interesting subject that comes with a lot of interrogations that we thought during the first part of the project. (What happens if someone delegates his vote to an entity, how to fairly split the vote ? Who takes the decision for the entity, what happens if too many people delegate their vote to an entity ?). It would still be a good improvement to the project since it introduces the notion of group. The objective of the creation of groups is once a proposition of vote has been cast, you can, before voting, discuss or debate with the other members and as you will be in a small committee, you will be more implicated, more open-minded to suggestions… and the participation rate would probably rise.

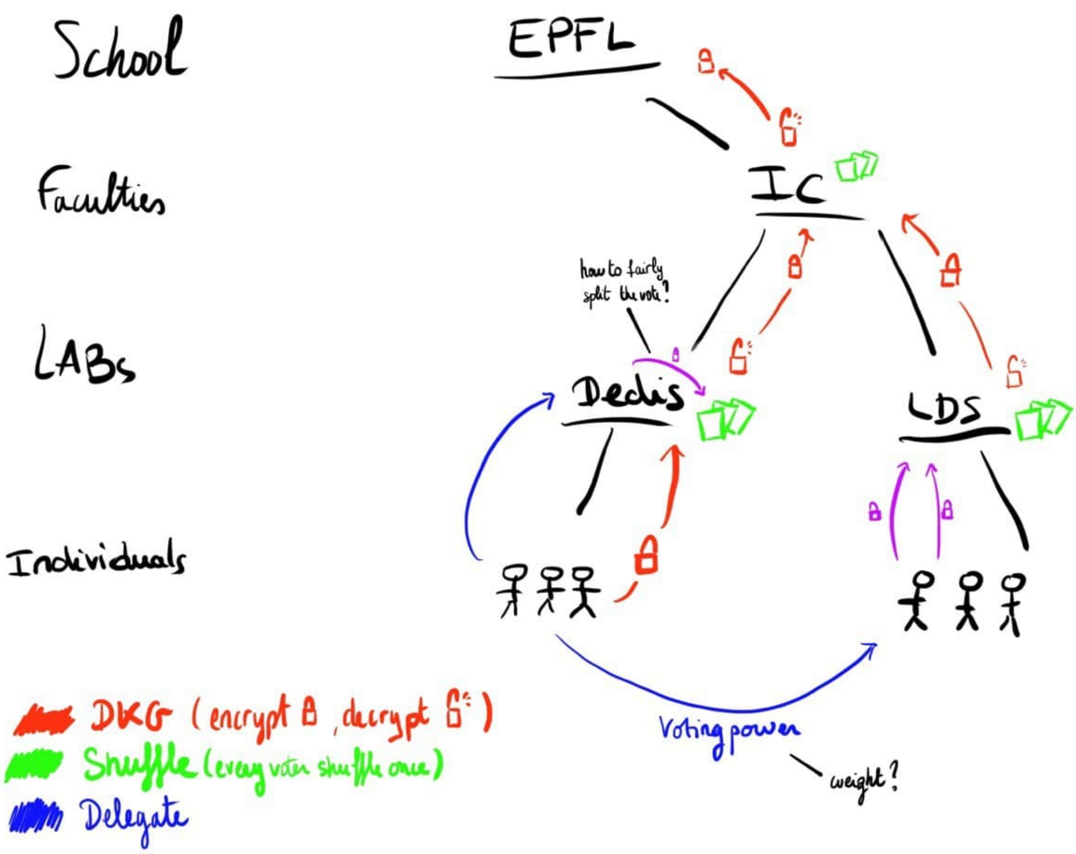
2.1.2 Secures systems

We also had a secure part through Distributed Key Generation (DKG) that ensures the encryption of the votes with Pederson Algorithm and it's anonimity with Neff-shuffling. [8] But we couldn't afford to implement this secure system mainly to simplify and narrow the project. We also thought of using the other project of the DEDIS lab (D-voting). We did spent some time learning about how it works even if we didn't implemented it.

For DKG we would have used Pedersen's algorithm, it works with the participation of people. The N people are represented as N nodes, and they all contribute to the generation of one private key and one public key (collective in case of federated architecture). If the threshold is obtained (T nodes out of N), then the secret can be computed using the Lagrange interpolation.

Neff-Shuffling means that after casting a vote, every node gets it's round of shuffling, we would use Kyber library to implement it. As in our project, secrecy by encryption and anonymity are not satisfied, we couldn't also use the DELA Blockchain for traceability and auditability of the votes. In the end, our approach was more practical than theoretical, therefore we won't develop it more.

A federated architecture (votes at different levels in a hierarchical way) and secure systems are great improvements that deserve to be further explored.



DKG, Neff-Shuffling and Federated architecture example on the EPFL system.

2.1.3 Database

For our website to be completely effective, it would need to be accessible online hosted by a server. Moreover each user should have different access rights (user, admin...), thus a user identification, this leading to a database storing the user, the password and more about the votation like the description, the list of voters, candidates if needed. It should also contain the voting results. This database even if very effective and useful was not implemented, but using the BBolt library could be a good upgrade of this part of the study.

2.2 Overcome the challenges

2.2.1 Circular delegation

Circular delegation is a process of Liquid Democracy where someone delegates parts of his voting power to someone else and after a while, some of his own voting power comes back to him. (Example: Alice delegates to Bob and Bob delegates back to Alice). At first, we found a solution for circular delegation in our simulation involving preference delegation list. Therefore as a voter, you would have to fill in a list of other voter that you would like to delegate to, but this problem is complex to solve and doesn't guarantee a situation where it always work (as the preferences list are finite). Let's take an example with 3 voters and 3 people in their delegation list:

- Voter A: B,C,D
- Voter B: A,C,D
- Voter C: A,D,B
- Voter D: A,C,B

Here we can clearly see a circle with A and B involved, in order to brut force a solution we would have to try all solutions until one is correct but it's complex. We would have for example as a first try: VoterA => B, VoterB => A, VoterC => A, VoterD => B; but we detect a cycle therefore VoterA will now delegates to his next person on his preference list. We see again a cycle with C and A, continuing the process this would give us 3^4 possibilities $O(n^(n+1))$ so with a large number of voters this solution is not viable.
In the end we choose to resolve this problem by taking the solutions directly into the data survey that we launched during the project, namely if you detect a cycle you delegate to another person or you choose to vote.

2.2.2 Optimisation of Delegation

In case of a partial circular delegation, meaning that you received back an inferior percentage of what you delegated, Mr. Ford introduced us to a solution. Instead of doing loops until your voting power expires, we should spot that there is a circular delegation and balance the amount you receive by changing the amount you sent at the beginning. Let's take an example: Alice delegates

100 to Bob, Bob decides to vote for 100 and delegates 100 to Charlie who is also voting for 100 and delegating 50 to Alice and 50 to Eve. In the end of this cycle, Alice received back half of what she invested.

In the same way we continue this loop, Alice delegates back 50 to Bob, who in the same percentage as before votes for 25, (50% of his voting power) and delegates 25 (50% of his voting power) and so on. But with Mr. Ford's idea, we could break this circular motion much sooner by increasing the amount Alice delegates the first time in order for Bob to directly vote 128 (100 first time, 25 second time, 3,5 third time and so on), thus she would need to delegate $100+(28*2) = 156$ to Bob, so that he can skip casting n times his votes and thus only having one iteration.

2.3 Simulation Design

2.3.1 Blank vote and abstention-vote

The blank vote or the absenteeism rate is very important and is present in any democracy, even in Liquid Democracy ? We decided to not implement them for our Liquid Democracy simulations as they don't bring any advantage or drawback of the Liquid Democracy into play. We considered blank vote or abstention-vote as a deep personal conviction, then you shouldn't split your vote between blank vote and delegating or between voting for yes and making an abstention vote, in either case you should invest yourself with 100% into it and thus either doing nothing with your voting power or voting 100% for the blank. Admitting this hypothesis, adding theses types of vote to our simulation doesn't seem relevant.

2.3.2 Maximum number of actions

On top of that, we decided to give voters a maximum amount of actions: three (that you can split between voting, delegating or both). Because it seems logic and realist that someone is not going to delegate to ten different people (except for fun) as you don't have time and desire to find more than 3 people. It's the same if you vote for more than 3 candidates, it seems a bit weird (but not impossible) as it would mean that you support 4 different points of view. We are aware that it's a limiting factor but it helps with the simulations and we think that a majority of the population is satisfied with 3 actions.

Later, we will see that we have done a survey and one of the results is that 35% of the people won't even use the liquidity of the system to split their vote.

2.3.3 Simulation rounding problems

Each voter is assigned a category which influence his actions and behavior in a proper manner. We encountered a problem when wanting to reproduce the previous actions of a user.

Let's say he voted for candidate A for 30%, for B for 40% and delegated the rest to Bob. When another user delegates some of his voting power, we need to split it in the same way as his first choice. But if he only received 40 of voting power via delegation then 30% of 40 is 12, which is an integer but then he will delegate this 12 to Bob and if Bob needs to split again we might ending up with fractions. We choose the integer type to represent the voting power (100 initially), even if loosing a little precision with rounding. We tried with the float64 type but it would go into too many iterations.

In the end the choice between integer and float64 isn't very relevant, since we choose as a solution to "get around" the problem by allowing our simulation to stop when the voting power of each user is less than 1%. This leaves unused some of their voting power. But as we are working on a great numbers of simulations and we think that the variance of these simulations would stabilize the results, so it won't be too relevant.

Additionally, to run our simulations a bit faster, we decided to loose some precision. When a voters has 10 or less of voting power, he stops splitting them and invest them in the category he voted the most. For example Alice delegated 60%to Bob, voted for 40% then Alice receives 5 of voting power, she will delegate all her voting power to Bob as delegation to Bob has the highest value in her history. This simplification doesn't seem that impact-full, since we are dealing with very small quantities of voting power.

2.4 Transitivity of delegation

In our project we choose to make the delegation process transitive, this means that when you delegate 100 voting power to Bob, Bob will receive 100.

First this allows Bob to delegate also himself without losing any power, thus it allows the creation of a chain of delegation. We thought of modifying the voting power when you delegate (you still delegate to Bob 100 but he only receives 90 for example), but this would discourage people to use the delegation process and therefore loosing all of the advantages of the delegation.

Furthermore, this would create a situation of "1 man / 0,9 vote" which contradicts the fundamental democratic principle "1 man / 1 vote". In our choice of design, the people who received delegations are not treated differently than anyone else: we could see delegation as copying the vote of the end node of the chain.

Chapter 3

Implementation

3.1 Implementation of the system in GoLang

We develop our voting system project using GoLang. Here is our most important structures :

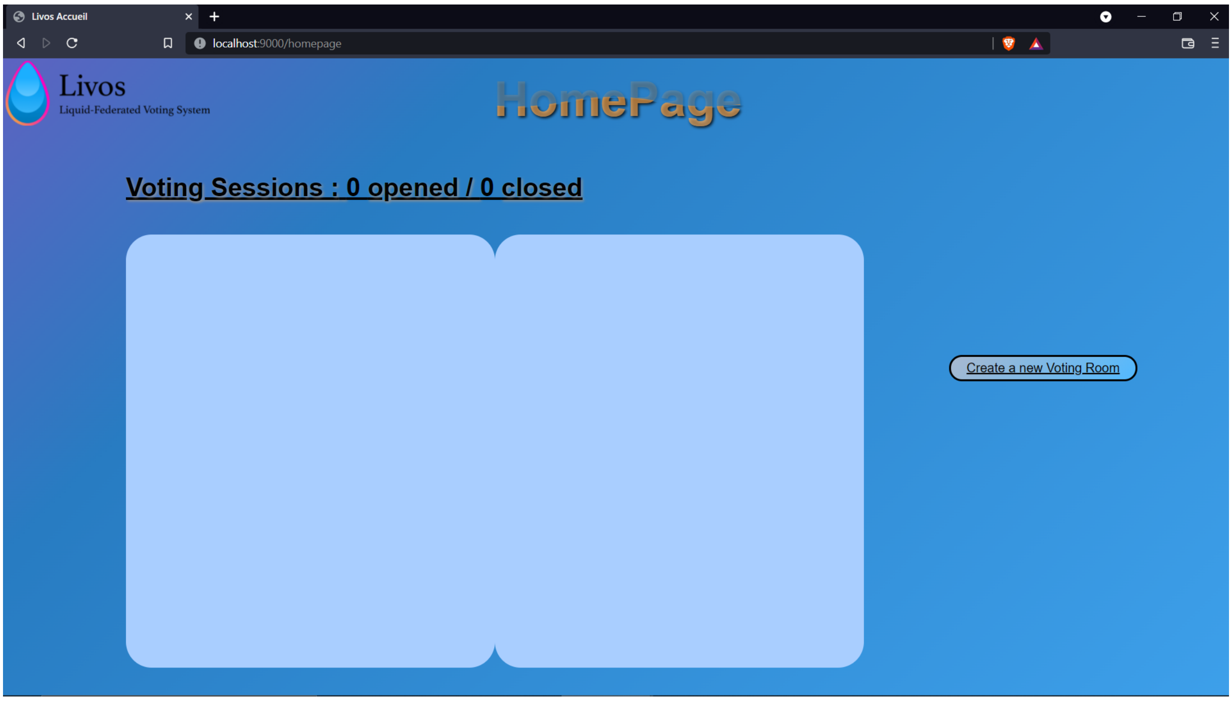
- VotingSystem : a structure that contains the all the different Voting Instances. It has his proper interface in order to interact with.
- VotingInstance : a structure that represent a votation to be resolved. It possess a VotingConfig. It has it's proper interface to interact with.
- VotingConfig : a struct that contains all the information necessary to a VotingInstance (list of voter, list of candidate, a description, Referendum or Election...).
- Users : a struct with a proper ID and can vote or delegate (part of) their voting power.

We also choose to interact with interfaces instead of objects. This allows us to interact with the structures more easily and more safely.

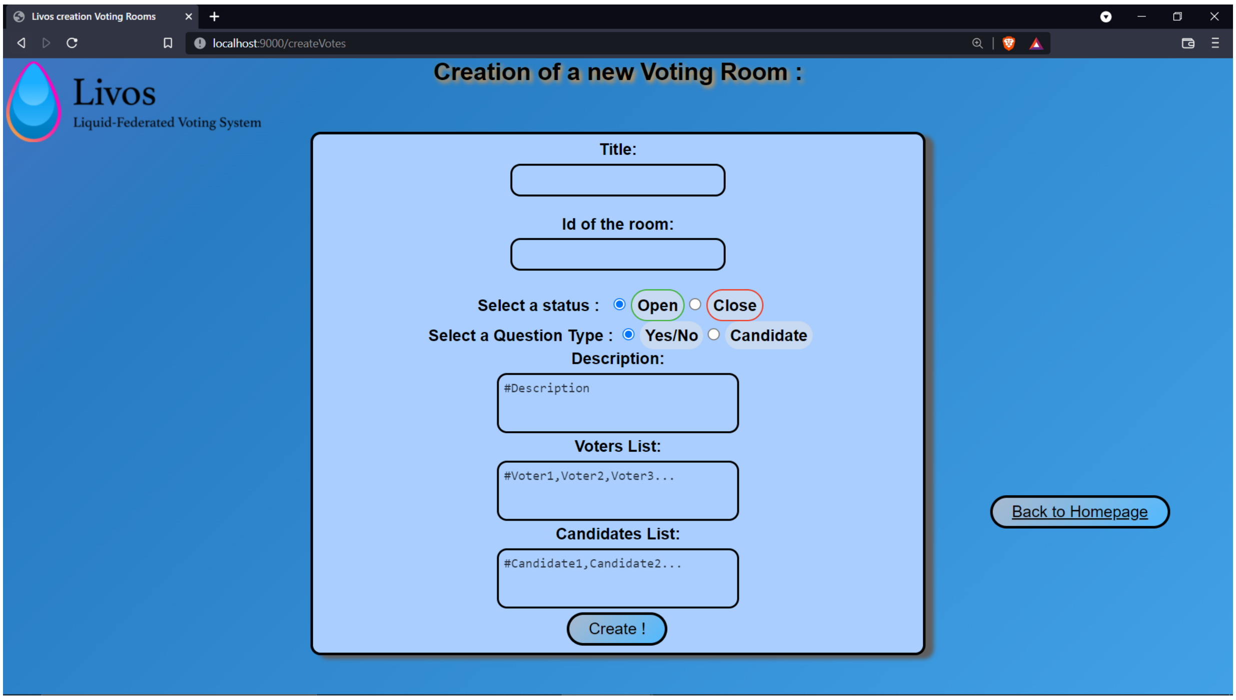
3.2 Server and web application

We have GoLang local server linked to web pages. The controller enables us to communicate with our back-end implementation by triggering some code (handle functions that is triggered by the main.go) when specific actions are done on the website. We use a combination of HTML and CSS to build most of our pages. We have a short JavaScript part in order to display a graph that represents the current state of the vote. We are using Viz.js, that allows use to render a graph from a well parsed string.

Because our project isn't focus on the security aspect of the implementation, we don't have proper sessions for each user. For now, we have a centralized entity which manipulates the website, allowing us to simulate small Election or Referendum.

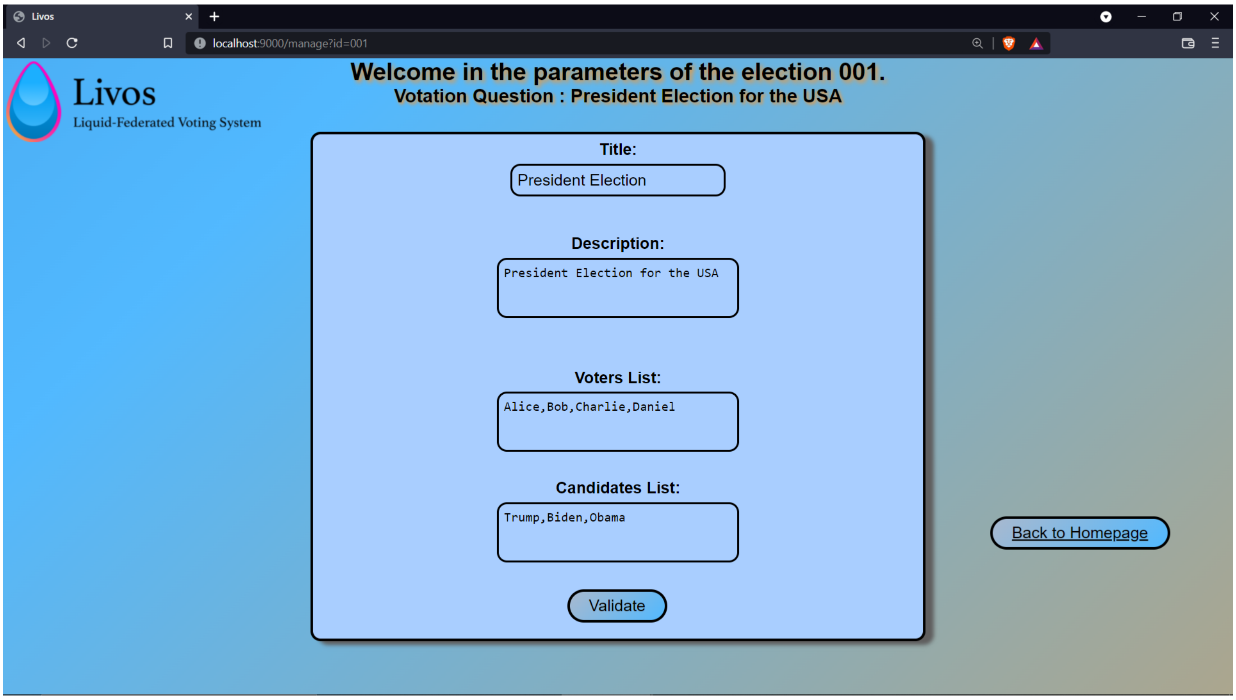


Homepage

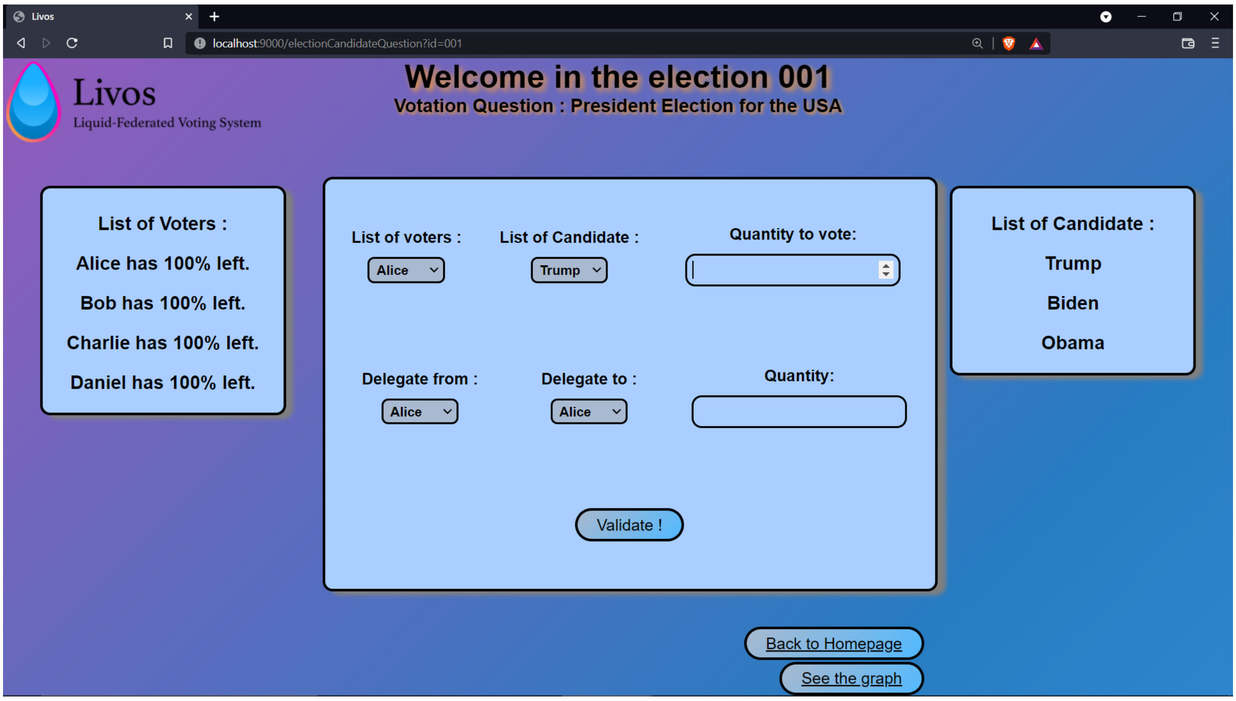


Creation of a Voting Room

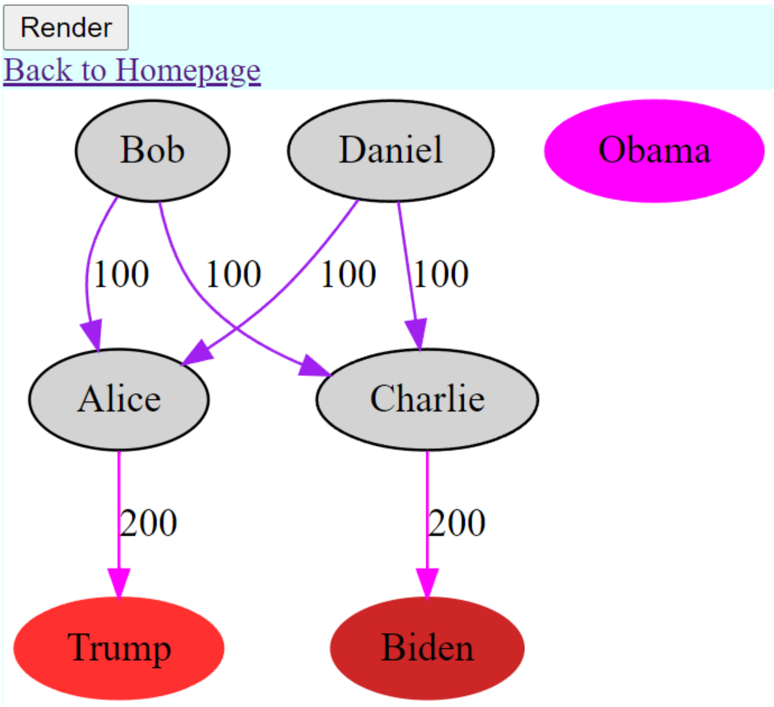
Homepage with Voting Rooms created



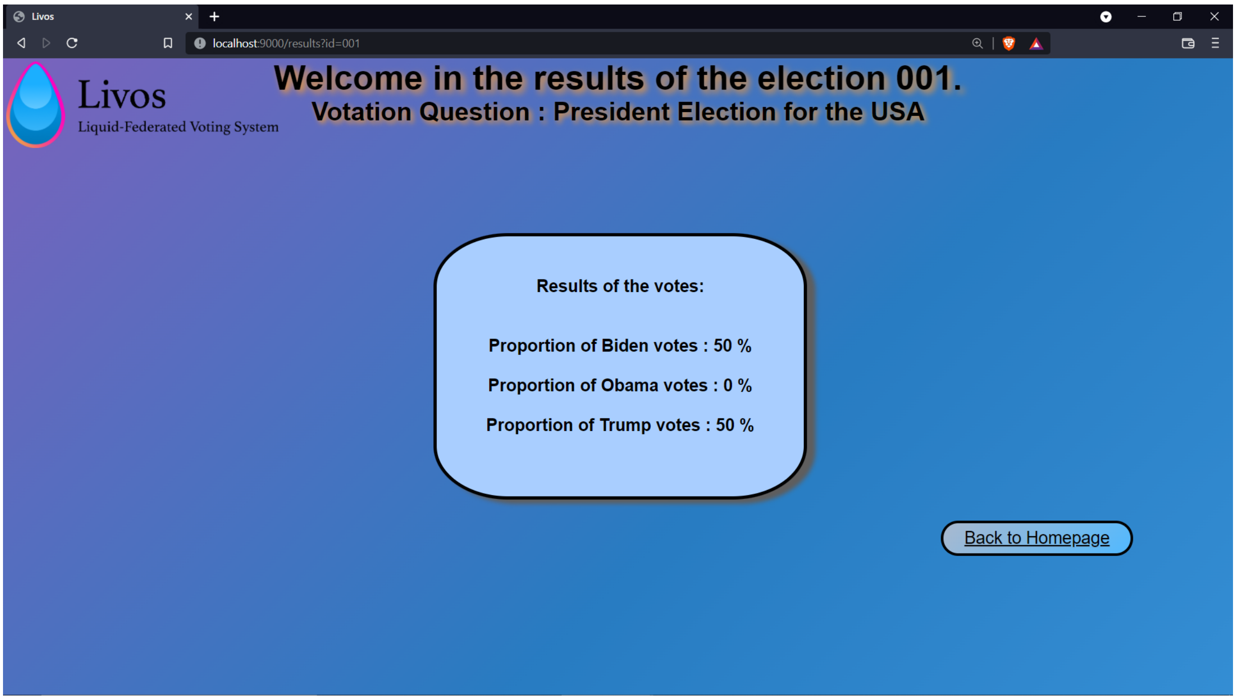
Managing the parameters of a Voting Room



The Voting Room, where voters can cast their votes (we can see their voting power in real time on the left side, and the list of candidates on the right side)



Graphic representation the current state of the Election



Display the Results of the Election

In the end, our website isn't used for a real experiment but it's a useful tool that is implemented, ready to be used. In the future of the project, we can put this website online and conduct a real experiment.

We also choose the fact that the user needs to come back to the website to perform actions (the different reasons are explained in 4.2 circular delegation); consequently it's not very user-friendly and in order to conduct a real experiment to make it more acceptable for a voter we could increase the votation time, for important questions it could last a month for example.

To maintain order, the voters have one week to vote or delegate the first time, then we repeat this process but decrease the time they have to vote or delegate their new voting power to four-five days for example, repeat this process until a month is passed.

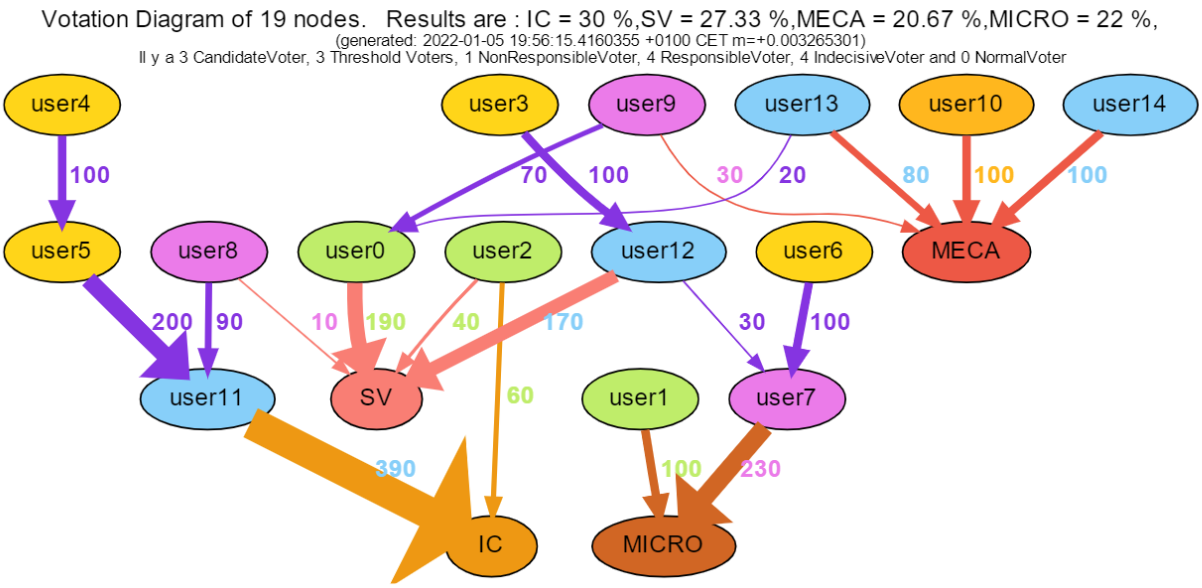
Then you have multiple solutions, either all the P have been voted, then give the results, or their is still voting power left then you can, based on the actions voting power of each user, determine a category in which he belongs to compute the result, by distributing it according the each voter type. Alternatively you can give a last votation where you cannot delegate. This part might need to be more researched as this is a suggestion to be worked on.

3.3 Simulations

We have multiple simulations we can run with our project. Coding them in their proper package and running them via their testing functions allows us to run them one at a time. One of the side effects of theses test functions is to write text in a provided file. Then, using the GraphViz library, we can transform it into a graphic representation of an election/referendum and display the results. The pipeline of a simulation looks like this :

- Creates the Voting System, the Voting Configuration and the Voting Instance and all the voters (which have a proper voting behavior)
- Make the voters vote or delegate until the voting power is almost exhausted (meaning most of the voting power has been cast)
- From the results, write into a file, following the desired format.
- Transform the file text into a graphic representation of the election in pdf format.

Here is a graphic representation of an election graph. In this example, we try to determine which EPFL faculty is the best. We can see the candidates that are in dirt/salmon/orange/red colors which are Computer and Communication Science (IC), Life Sciences (SV), Mechanical engineering (MECA) and Micro engineering (MICRO). We have 11 voters belonging to 5 categories (represented by 5 colors) that will dictate their behavior.



Graph of the election example

We will go deeper into the explications of the simulations below in a dedicated chapter.

3.4 Other paths explored

During the semester, we also explored other ideas and questions about Liquid Voting and it's challenges. We know that the Markov Chains is a well used mathematical concept in the representation and simulation of voting system in general. The possibility to delegate a vote comes with numerous theoretical problems such as the Circle Delegation or the Tyrant question (Merging of

the delegated votes to few voters that have huge impact on the total election/referendum).
Liquid Quadratic Voting : a way to encourage people to diverse their choices.

Chapter 4

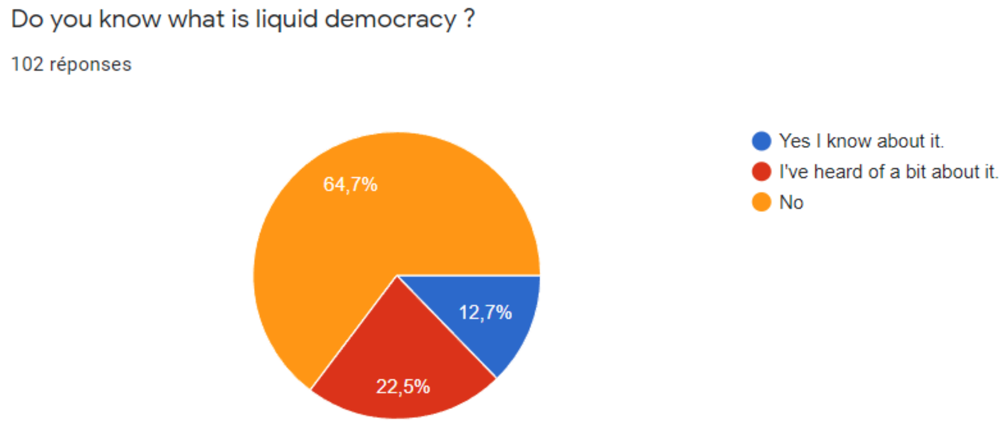
Simulations

4.1 Survey

Our goal being to run simulations to see the challenges of a liquid voting system, we need to run simulations closest to reality. To be able to run the more realistic simulations, we need to categorize voters into user behavior that we can implement. We received yet about a hundreds answers. We used this data to run our simulations but we are aware of many limitations of this. Obviously a hundred responses isn't representative of the real world. Also, a large percentage of the responses comes from French people that don't have the same relation to voting as the Swiss people.
Finally, most of the responses we received comes from people under 30. By asking people of all ages, from different nationalities (people that don't live in a democracy for example), more or less educated people, from different "social class", we would certainly get different results. But even if our scope isn't that large, we consider that the data we collected give us enough information to start building some simulations or asserting hypothesis about circular delegation, Tyrant problem and so on.

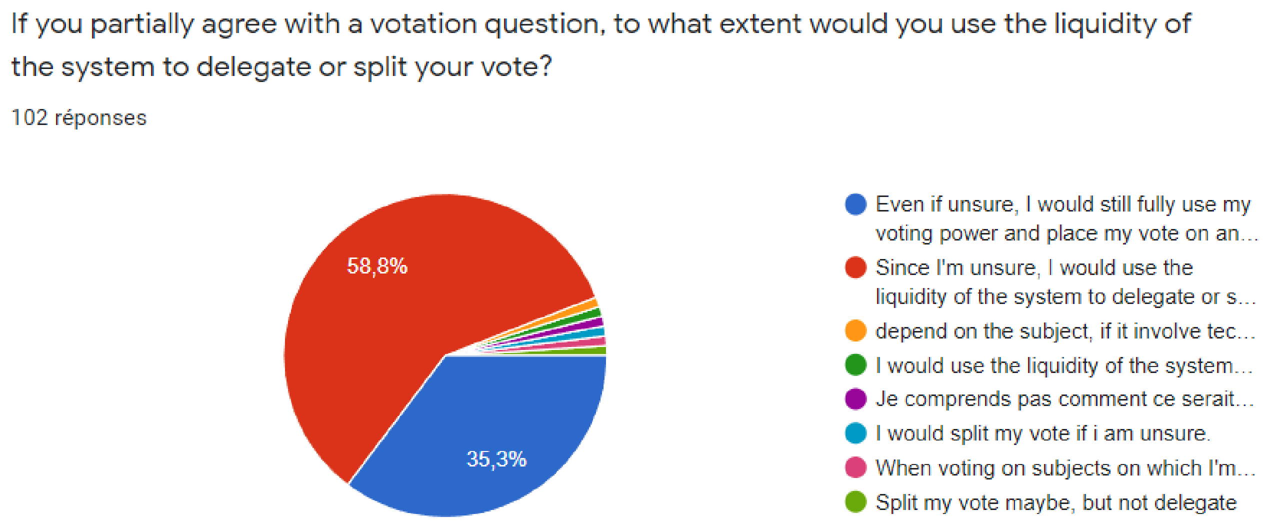
4.1.1 Results of survey questions

As excepted, 64.7% of the asked persons didn't know what Liquid Democracy is and 22.5% only heard about it.



Survey answer about popularity of Liquid Democracy

A more surprising data is that around 38.3% of the people won't use the liquidity to split their vote even if their choice is divided among many candidates. We expected the answers to be more favorable to the liquidity alternative.

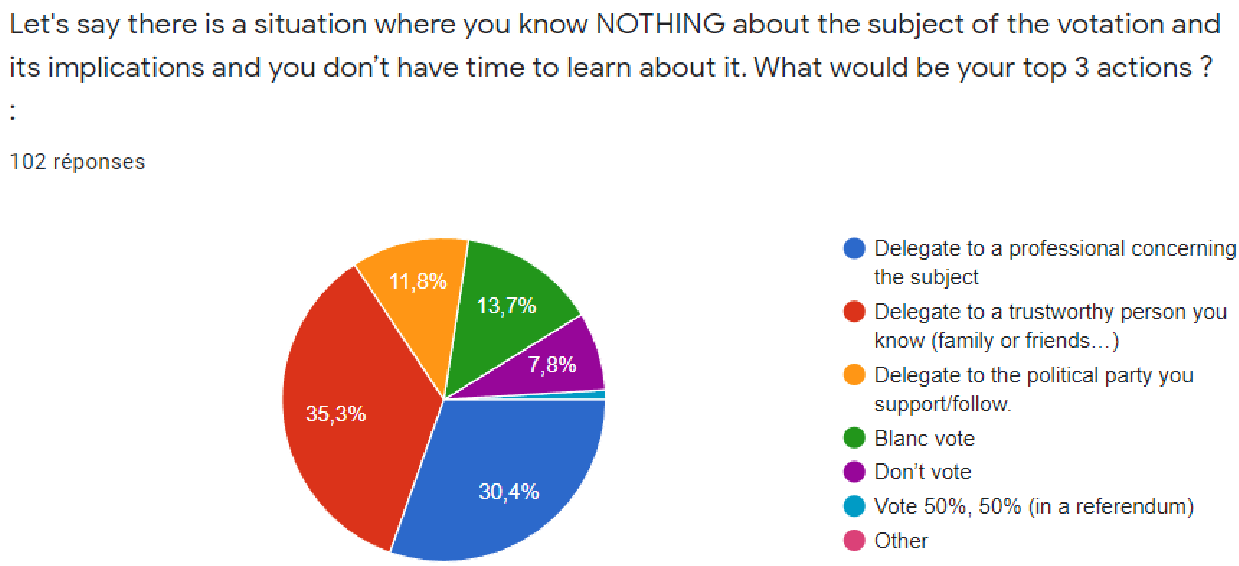


Survey answer about using the Liquidity of the system

The threshold of total voting power at which the people rethink their choices if very disparate. Between the people that possess such a threshold, we have that 11% have a threshold at (1-10%), 13% at (10-20%), 8% at (20-30%), 6% at (30-40%), 7% at (40-50%) and 29% at more than 50% (which is more that the majority, meaning that you have full control over the votation's results). Overall, 25.5% take the vote "seriously" in all cases, meaning that whenever they will receive some voting power through delegation, it will impact their choice (reconsideration, rethink/reevaluate their choice...). Also we can note that 14.7% of the people asked, do not care about how much voting power they own and this particular amount won't change their initial vote.

We ask the people what would they do if by delegating their vote they entered in an infinite circle delegation with no end. An interesting results is that 50% of the people chose to take a decision towards the election (or referendum) in order to break the circle (they renounce to their delegation). Then, 31.4% of the people chooses to delegate to another person, with hope that it will break the circle. Lastly, around 14% decide to vote blank (or not to vote). From this question, we expected more delegation alternative than taking a decision to vote.

Confronted to a situation where the people know nothing about the subject and with the additional condition that they don't have time to learn more about it, the opinions are divided. Around 35,3% prefer to delegate their vote to a trustworthy person they know (like a family member or a friend). About 30.4% prefer to delegate their vote directly to a professional. 13.7% of them prefer to vote blank and 11.8% prefer to delegate their vote to a party they are supporting. We can note that 7.8% of the people prefer simply to avoid voting. Theses are results we were expecting.



Survey answer about delegation preferences

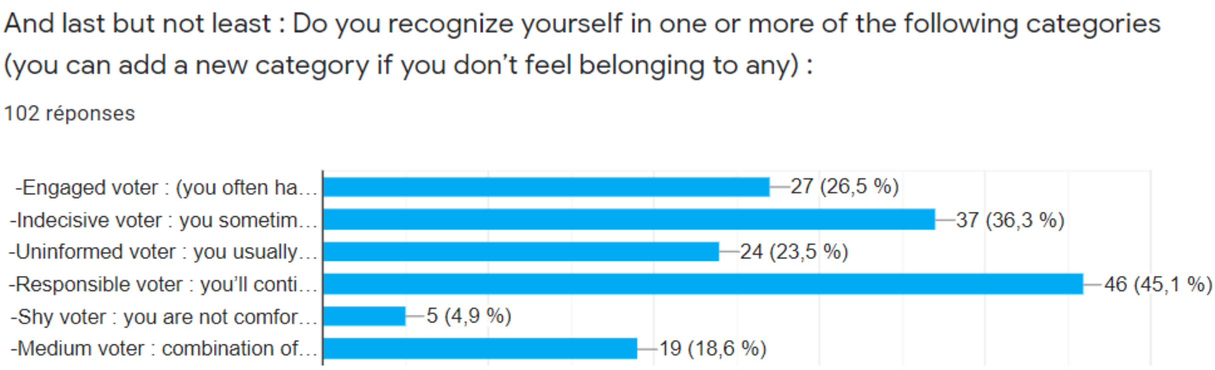
At the end, we ask the people to indicate in which (pre-made) categories they feel they belong to (they can choose as many as they want) Here are the pre-made categories we build from our discussions, reasoning and researches on the web :

- Engaged Voter : a voter that often have strong ideas and it's hard to make them change their mind. In our simulation, they choose a side (referendum) or they split their vote between candidates (election) and continue this way independently to the amount of voting power they receives.

- Responsible Voter : a voter that will continue to vote accordingly to their initial decision (delegation or vote), independently of the quantity of voting power they possess. In our simulation he chooses a side (referendum), split his vote between candidates (election) or delegate his vote and continue this way independently to the amount of voting power he receives.

- Indecisive Voter : a voter that delegates his vote because he struggle to choose a side as he is sensible to the arguments of both sides. In our simulation, an Indecisive voter always delegates.
- Uninformed Voter : a voter that delegates his vote because he doesn't know the subject well.
- Shy (Non responsible) Voter : a voter that refuses to take responsibilities when they receives some additional voting power via delegation. In our simulation, he chooses a side (referendum), split his vote between candidates (election) or delegate his vote and he always delegate the voting power he receives.
- Medium (Threshold) Voter : a voter that is a perfect mix between the Responsible voter and shy voter. He will vote the delegated power he received in the same way as his first choice until a threshold. After this specific amount has been reached, he will always delegate his vote.

The results of this question are :



Proportion of answers to the categories partition.

If you are curious, we invite you to take part in this experience and to see for yourself and answer our survey : " Liquid Voting Survey "

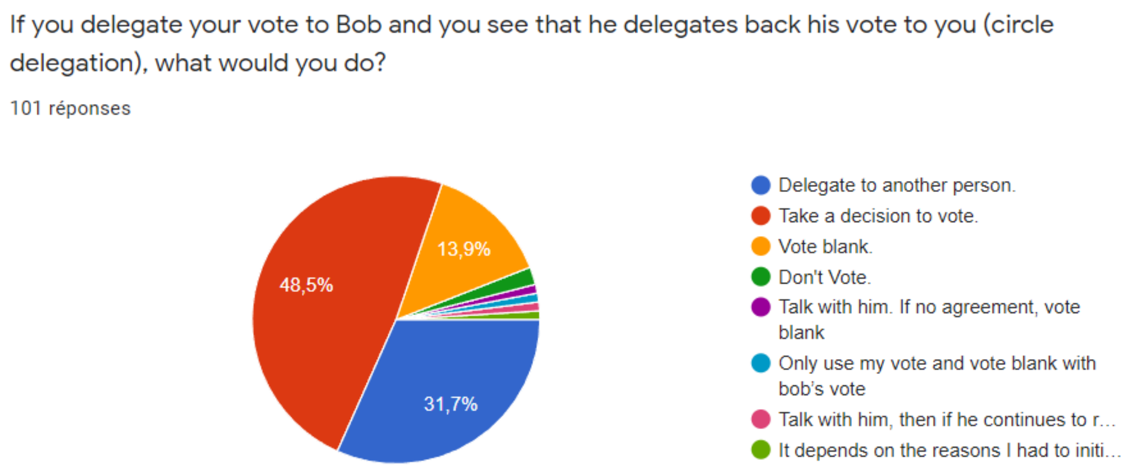
4.1.2 Remarks about corruption of votes

In our survey we had quite a few remarks, questions about the application of a Liquid Democracy system mainly because they thought Liquid Democracy would develop, encourage corruption since it's easier to buy a vote with delegation. These remarks made us think. In our opinion it is not very different from a DD since in either democracy you have many ways of riging an election (impersonation of voters, missrecording of votes and so on). The only difference we found is that the corruptors are sure to have received the vote since in Liquid Democracy you can delegate your vote to the malicious voter. (In other systems if you pay someone to vote for a particular candidate, how can you be sure he does it ?). [2]

4.2 Circular delegation

Our data survey tends to confirm that circular delegation is a false problem in real life application, which is something that is already believed by some circles in the liquid voting community. [9] The reasons why you delegate your vote usually are because you don't have enough time to answer, or you know a trust worthy person, a professional or an entity more competent on the votation theme.

Therefore in real condition, if one node of the cycle knows he belongs to a cycle, he will take a decision to break it : in (50+14)% he will take time to take a decision for the group (the group being all the persons belonging to the circle) to either elect a candidate(50%), vote yes/no if it's a referendum or vote blank(14%). In 30% he will delegate to another person, resulting most of the time in breaking the cycle. So as soon as there is activity inside the cycle, the reality shows that it doesn't last very long.



Survey answer about circle delegation.

This raises another problem : people need to be aware that there is a cycle. Therefore they need to receive a notification alerting them to come back to the voting session and vote again manually.

Could we automate this process to be more user-friendly ?
In theory yes, but if we automate every action so that the user comes only once and then the program does the job after, it raises in our opinion more problems than it solves : First we would need a user to answer a lot of questions that correspond to different cases such as what happens if someone delegates his vote to you. As we see in the survey, depending on the user type (responsible, threshold or shy), the behavior is very different. Therefore we would need to categorize people into known behaviors which doesn't seem very user-friendly. Additionally, it's not a good solution to ask the people to fit themselves into a category because they are biased. For example if we say: "if you appear to be a responsible voter, you will never have to come back to the website because the computer will distribute the power the same way you did with your

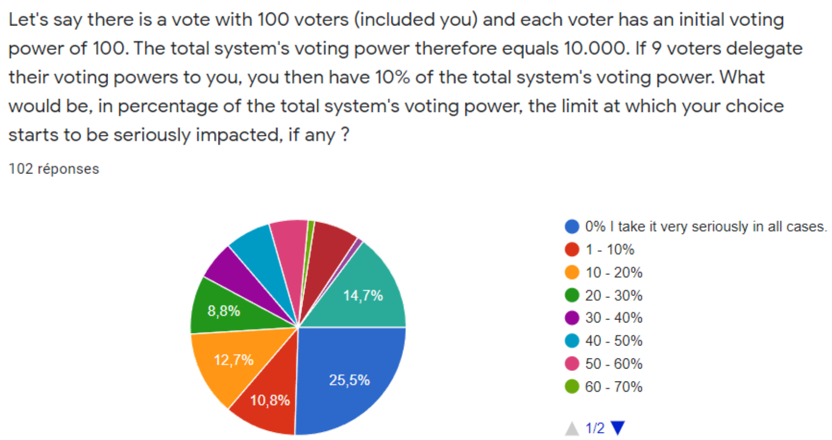
initial power." It will bias the opinion of the voter who will go for the easiest solution therefore be a responsible voter and not use delegation.

We also thought about another challenge with circular delegation : if the person doesn't come back to check if he obtained new voting power, he cannot manually break the circle. A potential solution would consist in creating preference lists of delegation. There is a simulation trying to recreate this situation. For each user, we must create a list of person you would delegate your vote to. In case of a cycle, we could try different combinations of preference list to see if the cycle is broken and it doesn't create another one. But at the end, we implemented a solution from our data survey as we want to be closer to reality.

In the end, as each behavior is different, we can't really force people to automate their decisions because we would loose some accuracy as the models that we have don't completely correspond to the voter's behavior. It it a complicated challenge to model and let the computer work on his own without any loss of precision.

4.3 Tyrant problem

We might also think about the fact that with a delegation process, we could have one man too powerful, with too much voting power. This is indeed a problem if a person, say a well know politician, receives many direct delegation (we can note that in our implementation, we cannot revoke our vote or our delegation); then we obtain a person that has too much power. If this politician receives his power from a chain of delegation, it implies that everyone in this chain is a chance to break it, and therefore reducing the amount of power of the tyrant by becoming himself a smaller tyrant. Thanks to the information received in our survey, we can see that almost 50% of the people are responsible voters, which means that no matter what amount of voting power they get, they will not reconsider their vote/choice. They will take a decision that is independent from the amount of voting power they have. This can be elaborated by the fact (a bit contradictory) that 15% of the people never reconsider their choice even if they have 100% of the votes. 50% reconsider their vote, as soon as they have 25% or more of the voting power.



Survey answer about threshold

Furthermore, it seems unrealistic that everyone chooses to delegate their vote to a single common person, since if you choose to delegate, in our survey, you will delegate to a professional or a trustworthy person (32% of the cases), or either delegate to your political party or vote blank in about 13% of the cases. Therefore the spectrum of people you are delegating to is too large to join back to one unique person (if we consider having a complete population with a large number of voters). So finally, it seems quite complicated to create a tyrant as a majority of people choose to vote independently from their voting power and they seem to delegate to different persons who aren't linked.

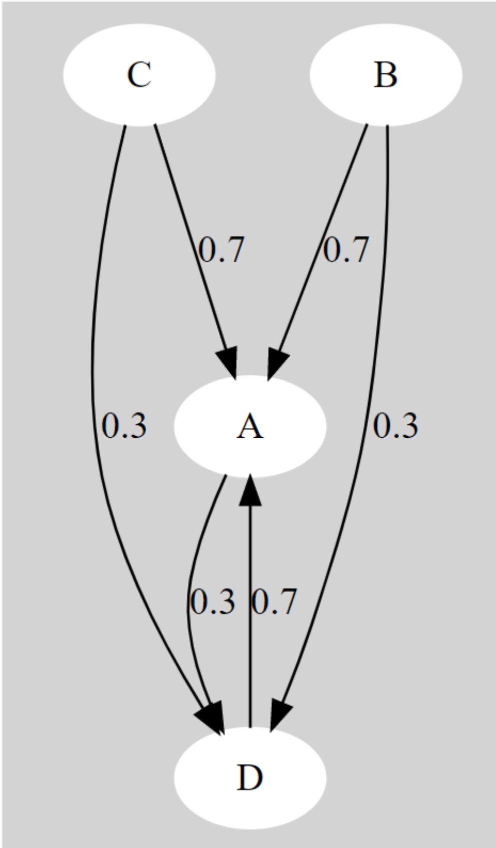
4.4 Markov Chains

In a normal democratic system, the winner of an election is the one who gets the more votes, but using the liquidity of our system, we could elect not the most voted candidate but the one that best suits the most people. This section has only been seen in theory, we haven't implemented a simulation with Markov Chains because it seems sufficient to just explain our reasoning. Let's see an example to see why Markov's chain can be useful in our project. We have 4 voters and 3 candidates (Candidate A => A, Candidate B => B and Candidate C => C). We ask the voters to sort the candidates by preference order :

- Vote 1: A,B,C
- Vote 2: A,B,C
- Vote 3: B,C,A
- Vote 4: C,B,A

From this we can infer that candidate A should be winning as he has obtained the majority of the votes but we can note that Voter3 and Voter4 don't like candidate A (last in their preference list) and this antipathy should count therefore to satisfy the maximum amount of people. We should maybe elect candidate B as it is more in adequation with everyone choices. This simple example offer a simple conclusion, but the theory behind this is very interesting. Indeed, in our system we have a lot more voters and candidates, the liquidity part can be introduced as a tool to sort the candidates. To see this, we need to first construct a graph. The nodes are our candidates and the edges are gonna be weighted (from 0 to 1) by the amount of voting power that this candidate has.

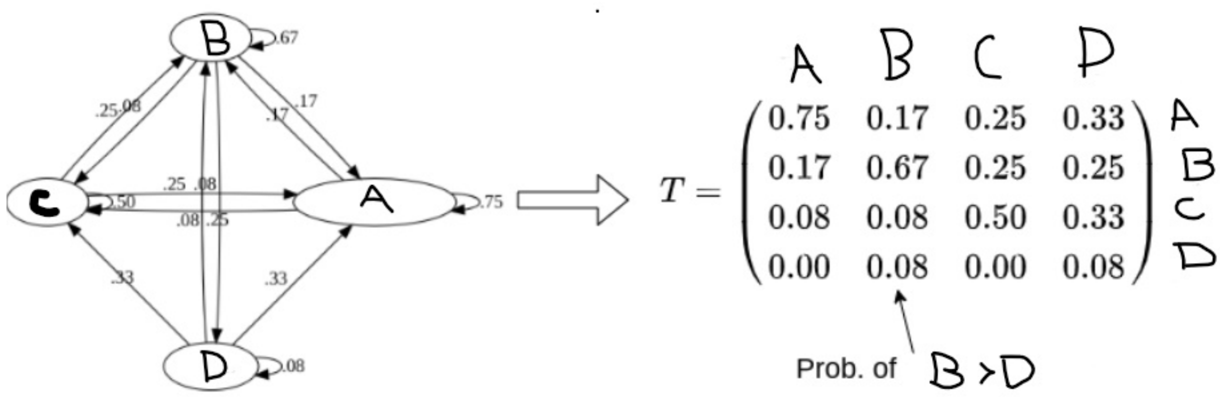
Let's see another example : here is a list of preference of a voter in the election between 4 candidates (A, B, C and D)



Representation of the preference list of a voter.

Here if we transpose this graph, we see that this voter preference list would be: A (in 70%), D(in 30%) then C and B (with 0%). A is preferred over C in 70%. For each voter, we are going to construct one graph like this and then add every graph together. Each edge in the summed graph is assigned a weight which is the sum of the weight of this edge of all the graphs divided by the number of graphs : the weight is normalized. Example : to add two edges that have 0,7 and 0,3 respective weights, we have (0,7+0,3)/2 = 0,5.

Then add a self-edge to each node with a weight equal to the sum of all weights of incoming edges. Finally, transform the graph to normalize all the weights so that the sum of the weights of all outgoing edges from a node is 1. Then we can construct a n*n matrix (n being the number of candidates). This will be our Transition Probability Matrix.



Scheme of the Markov system transformed into the transition matrix

In the end we need to multiply this transition matrix by a n*1 vector(composed by one element replicated everywhere namely 1/n). This product of matrices will give us a stationary distribution and the one that obtains the most results is the winner of the election.

4.5 Liquid Quadratic Voting

We implement a version of the vote counting that differs from the traditional way. In the traditional way, initial voting power being 100, Alice can vote 100 for Candidate A (and it gives him 100 points) or she can split her voting power between candidate A and B (50 each). But in a Quadratic voting system, the votes aren't counted the same way. This system is designed in order to favor the split of votes and encourage liquidity. Here we try this formula : $f(x) = \sqrt{x * 20}$, which maps 100 to 44.72. Note that under this system, if Alice votes less that 20 for a candidate, she actually has a bigger impact and creates some voting power. However, since every voter has the same possibility and initial voting power, it doesn't favor any particular voters.

By implementing the system with the formula : $f(x) = \sqrt{x}$ we get the same results in the simulations. (see below) This system provides a protection from extreme minorities that can, in a liquid system, all support an extreme candidate and since the other voters would usually split their votes, the extreme candidate might have a chance to win, satisfying only the initial minority. There is still work to do on this system and our simulations, but we provide a way to simulate it easily and to conduct experiments concerning the Quadratic Voting System.

4.6 Liquid asset comparing to the classical voting systems

We run simulations to get some results concerning the applications of a liquid voting system in multiple environments. One objective of this is to calculate the precision that a liquid system bring compared to a more traditional (non liquid) system. We modulate the parameters to obtain different setups where we can evaluate the impact of our Liquid System.

4.6.1 How do we measure the difference ?

Let's denote by "traditional system" a voting system with forbid any form of delegation and do not let the voters split their votes between multiples choices. As for an example, the presidential elections in France (without the possibility of proxy) would be a traditional voting system.

To measure the impact of liquidity we look at the differences in the results. This is a simple way to see if a liquid system bring more precision than a traditional one. Actually, in our simulations the reference is the liquid system, and we translate the results of this system into a traditional system by theses rules :
For a voter who voted at least for a candidate, we look at his favorite candidate and vote 100% for him (favorite candidate is the one that receive the most voting power from Alice: it's a majority principle). For a voter who only delegates it's vote, we make him vote blank as he couldn't make a choice. We can then take our Liquid system as a reference and see how much the results of the traditional system deviates from the ones of the liquid system. To do that we evaluate for each candidates, the proportion of votes he gets (the absolute value of the difference) : Difference of candidate A = |Score of A in Liquid system - Score of A in traditional system|. To obtain the overall loss of precision of the traditional system we sum up the differences of each candidates.

In order to simplify our simulations and reduce their running time, (some of them are too long to be run with our material), we stop the simulation when every voter has less than 1% of their initial voting power. This is a choice of design that allows us to run more complex and heavier simulations. This implies that only between 99% and 100% of the total voting power is process in our simulation. We consider this not to be harmful for the results we obtained as we run lots of simulations. But it is useful to note that we have a maximum of 1% margin error in our simulation's results.

4.6.2 Different parameters

Many parameters are relevant for the simulations, we modified them in order to obtain different setups. Firstly, we have the Balanced / Non Balanced parameter. When confronting candidates (or the yes/no choice for a referendum), we can make the voters choose purely at random which implies that every candidates would have a $\frac{1}{number of candidates}$ chance to be picked. (50% each for referendum). This is the Balanced parameter. On the contrary, the Non Balanced parameter implies that there exists a general preference in the choice for the candidates (or the yes/no for the referendum). For the election with 4 candidates, we test 2 partitions for preferences :

-40% for the first, 30% for the second, 20% for the third and 10% for the last
-65% for the first, 20% for the second, 10% for the third and 5% for the last.

For the referendum vote, there is a 80% to vote No and 20% to vote Yes for the ones that have the choice. Theses parameters come from the fact that Liquidity precision could be erased by complementary votes. For example if Alice votes 70% for a candidate and 30% for another one and Bob does the inverse choice (30% for the first and 70% for the second), then when translating this into the traditional way of voting, we get 100% for first candidate and 100% for second candidate (same final results as liquid way). But with our method, we would see 0% difference. However the liquidity allows the voters to specify their opinion. We must note that there is liquidity at work here even if the final result of this example is the same in both cases. We try to unbalance the choices of votes to see if liquidity could have more impact.

Secondly, we have the Real data with or without Indecisive voters. For these Indecisive voters (voters that fully delegates their votes because they don't know the subject or they don't have time to vote), the translation from the Liquid system to the traditional system is done by making them vote blank which is a realistic solution. We are aware that there exist other alternatives solutions that could lead to different results. With this implementation, there is a big difference between the liquid system and traditional one as the proportion of Indecisive voters (20%) will be transformed into Blank Voters that differs from the liquid results (where there is no blank votes possible). To moderate this, we run simulations with Real Data but without Indecisive voters to see the concrete differences. That gives lower differences results as we will see below. From the survey we construct this partition of voters :

- 23% of Engaged Voters (strong mind, one choice)
- 20% of Indecisive Voters (always delegate)
- 40% of Responsible Voters (assume their choice when voting for others)
- 1% of Non Responsible Voters (don't vote for others)
- 16% of Threshold Voters (mix between Responsible and Non Responsible)

We also have a partition without the Indecisive voters that plays an important role in the simulations as we will see below :

- 29% of Engaged Voters (strong mind, one choice)
- 50% of Responsible Voters (assume their choice when voting for others)
- 1% of Non Responsible Voters (don't vote for others)
- 20% of Threshold Voters (mix between Responsible and Non Responsible)

We'll call theses 2 partitions "Real Data" and "Real Data without Indecisive".

Lastly, a crucial parameter is the number of voters. Indeed, we run simulations with 100 voters (= a big laboratory size) to 10.000 voters (= all EPFL) with intermediary steps (mostly 1.000 voters). This parameter allows us to detect how the liquidity impact scales with growing participation (under the conditions of our system).

Chapter 5

Results

5.1 Liquid precision : Concrete results of the simulations

The percentages in the following graphs and tabs represent the differences from the results of the candidate votes (or referendum votes) between the Liquid voting system and a more traditional voting system (without liquidity or delegation).

5.1.1 Liquid precision with candidates

With **4** candidates	**Realist Data**	**Realist Data without Indecisive**
Balanced (25 % each)	100 voters : **19.49 %** (10.000 simulations) 1000 voters : **19.39 %** (1.000 simulations)	100 voters : **5.43 %** (10.000 simulations) 200 voters : **3.86 %** (5.000 simulations) 500 voters : **2.43 %** (2.000 simulations) 1.000 voters : **1.71 %** (2.000 simulations) 10.000 voters : **0.49 %** (100 simulations)
Non Balanced (40, 30, 20, 10%)	100 voters : **19.54 %** (5.000 simulations) 1000 voters : **19.24 %** (1.000 simulations)	100 voters : **5.14 %** (10.000 simulations) 1.000 voters : **1.66 %** (2.000 simulations)
Non Balanced (65, 20, 10, 5%)	100 voters : **19.76 %** (5.000 simulations) 1000 voters : **19.32 %** (1.000 simulations)	100 voters : **4.37 %** (5.000 simulations) 1.000 voters : **1.52 %** (2.000 simulations)

Precision difference between Liquid voting and Traditional voting with 4 candidates

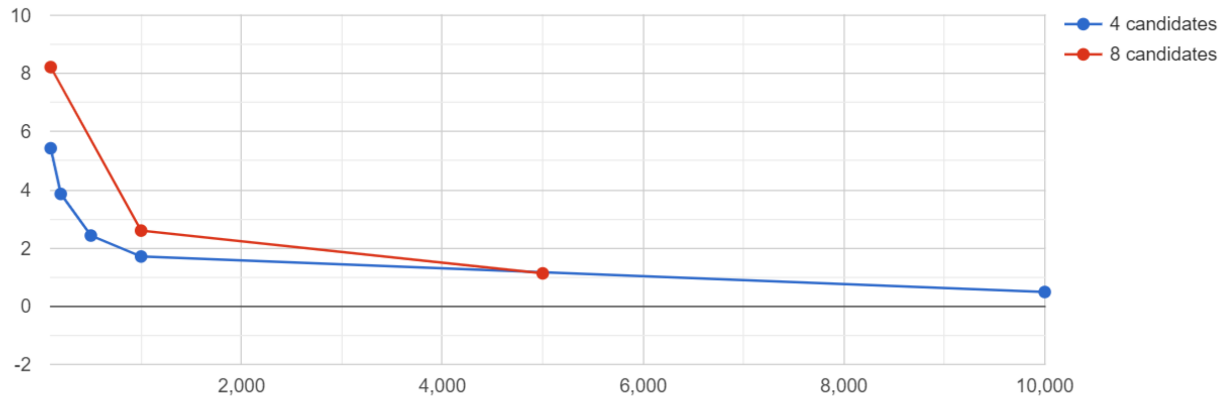
We see that in the most realistic scenario (based on our liquidity survey), we can get near 20% more accuracy with a Liquid voting system compared to a traditional one. We can observe that if we remove the Indecisive voters from the pool of voters, the percentage of difference drops to 5.43% with 100 voters and it decreases when the number of voters increases. This implies that the liquidity impact tends to fade when dealing with high number of voters (> 10.000). [see graph below] We can notice that having a election that isn't balanced doesn't have much of an impact on the results with the Real Data (either with Indecisive and without Indecisive).

With **8** candidates	**Realist Data**	**Realist Data without Indecisive**
Balanced (12.5 % each)	100 voters : **20.47 %** (10.000 simulations) 1000 voters : **19.22 %** (500 simulations)	100 voters : **8.22 %** (5.000 simulations) 1.000 voters : **2.60 %** (2.000 simulations) 5.000 voters : **1.13 %** (100 simulations)

Precision difference between Liquid voting and Traditional voting with 8 candidates

We see that having more candidates options with the Real Data without Indecisive increases the impact of the liquidity. (4 candidates => 5.43% 8 candidates => 8.22%). On the contrary, with the Real Data we notice a constant impact of the liquidity (slightly decreases : 19.49% to 19.29% which means a loss of 0.20% by multiplying by 10 the amount of voters. We can consider it negligible). This is due to the fact that Indecisive Voters vote blank in the traditional system which differs completely from the Liquid system and it brings a notable difference.

Here is a graph that represent this difference of precision between 4 and 8 candidates in terms of number of voters :



Graph representing the % of difference between liquid voting and traditional voting in terms of number of voters between setups with 4 and 8 candidates

We can observe a quick drop of the impacts of liquidity in a negative exponential way.

5.1.2 Liquid precision with referendum

Referendum	**Realist Data**	**Realist Data without Indecisive**
Balanced (50 % each, and equal number of YesVoter and NoVoter)	100 voters : **17.26 %** (5.000 simulations)	100 voters : **3.87 %** (3.000 simulations) 1.000 voters : **1.20 %** (500 simulations) 10.000 voters : **0.49 %** (100 simulations)
Non Balanced (still 50% each but twice amount of YesVoter and 0 NoVoter)	100 voters : **17.96 %** (3.000 simulations) 1000 voters : **18.03 %** (500 simulations) 5.000voters : **16.70 %** (10 simulations)	100 voters : **7.01 %** (3.000 simulations) 1.000 voters : **6.71 %** (1.000 simulations) 10.000 voters : **6.00 %** (6 simulations)

Precision difference between Liquid referendum and Traditional referendum

We observe that the difference between the Liquid system and traditional system remains constant (or slightly decreases but it's almost negligible) independently of the balanced parameter.

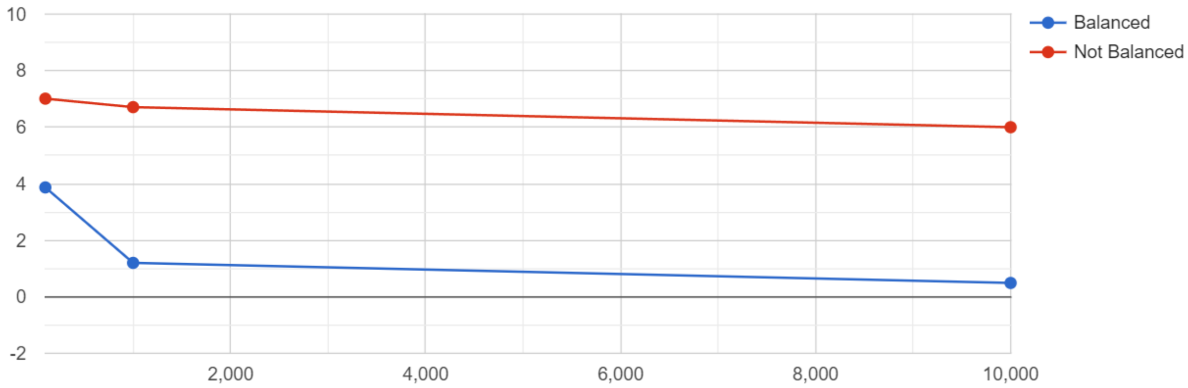
With the Real Data without Indecisive and the balanced setup we can observe the same drop as the election vote. The impact of liquidity seems to fade when there are lots of voters.

We also run simulations with a unbalanced preference for No for the referendum : same number of YesVoter and NoVoter but all the others have 80% chance to vote for No and only 20% to vote for Yes, without Indecisive voters. This gives us the following results : 100 voters : 4,6% of difference (500 simulations), 1.000 voters : 4.19% (100 simulations), 10.000 voters : 4.00 (1 simulation - 86 seconds).

More surprisingly, without Indecisive voter and a non balanced simulation we notice a very slow loss of precision compared to the balanced version of the election equivalent setups. The impact of liquidity is still 6.00% with 10.000 voters participating. We couldn't do heavier simulation due the resources we have but it would be interesting to see if there is a threshold value where the Liquidity impact fades in this particular simulation.

We can note that the biggest simulation (10.000 voters with non balanced setup) runs in 58 seconds. We run 6 of them. In addition, when running the simulations, the program stops if the simulation last longer than 10 minutes. This restrains our choice and variance for the simulation's parameters.

Here is a graph that represents this difference of accuracy between balanced and non balanced referendum in terms of number of voters :



Graph representing the % of difference between liquid voting and traditional voting in terms of number of voters between balanced and non balanced setups.

We see that it's much harder for the non balanced simulations to have an important loss of precision.

5.2 Quadratic Voting impact

Quadratic Impact 4 Candidates	**Realist Data**	**Realist Data without Indecisive**
Balanced (50 % each, and equal number of YesVoter and NoVoter)	100 voters : **4.27 %** (1.000 simulations) 1000 voters : **1.40 %** (100 simulations) 10.000 voters : **0.41%** (4 simulations)	100 voters : **2.74 %** (1.000 simulations) 1000 voters : **0.86 %** (200 simulations)

Result difference between Quadratic voting and Linear voting

We can observe that the differences between the results of Quadratic Voting System and the classical Liquid voting system tend to fade when the number of voters increases.

5.3 Awareness of the specificity of our simulations

We are aware that our results are subject to our design choices and that different choices, more profound data analysis, different data and parameters could lead to different results.

Chapter 6

Related Work

A great book that was very useful in our researches: The Principles of Liquid Feedback, 1st edition since it evokes the circular delegation and the tyrant problems. But it only scratch the surface as it doesn't speak about the automation of the process and assume that each voter is gonna resolve those conflicts by being aware of the problem.

Markov chains is also a widely explored subjects because it's easy to make a link with democratic systems. As it's easy to represent candidates with nodes and votes with arrows. But the researches we found were mainly focused on Direct Democracies and we had to reshape it to Liquid Democracy. [7]
For the types of voters we mainly invented the different categories. In other analysis projects, people tends to directly take political parties and therefore have prefabricated categories.

We haven't made a lot of researches about minorities because this has been an explored path, there exists already algorithms to make noisy minorities have an equal speak time as other minorities (Harmonic Weight) [9]. There also exists the Proportional Runoff Algorithm [5] that works pretty well. (a type of ranked preferential vote counting method used in single-seat elections with more than two candidates). On top of that, minority is a subject deeply inspected by machine learning experiments [6]. As in machine learning, you often have an isolated cluster that represents a minority, different and numerous researches were made about those as they dilute the solutions.

A question that we skipped in the survey results is also about minorities: in a minority, they usually can't express their opinion because they have too little impact. 80% of the people feel that if they had more voting power (with delegation) they would go back to their minority opinion and vote for this minority. 20% would still vote for the majority idea most in agreement with them. We can conclude that a liquid system in most cases promotes the emergence and the support of minorities.

Another way of improving our system is for the website to respect the golden rule of Liquid Democracy implemented by Google in 2015 [4]. It consists in making a trade-off between the privacy of the votes (no one can see what you have done with your vote) and trust because when you voluntary delegate to someone (say Bob), you trust Bob; therefore Google decided to let people (Alice for example) see what Bob does with her vote. "If I give you my vote, I can see what you do with it". If Bob is doing something that doesn't suit Alice, then she can retract her delegated vote and delegates to another person. But this Google research bases its further improvement on social media and Liquid Democracy. They wanted to create a social media that uses Liquid Democracy, a social media where you could vote, take advise from people, delegate your vote on many different subjects (foods, elections...).

Chapter 7

Conclusion

To conclude our proof of concept, we can say that Liquid Democracy could and should be used in every democratic systems. It allows people to be more involved, as they can delegate their vote even if they don't have time or desire to participate. Through our simulations, we have seen that in specific situations a Liquid System could be 20% more accurate than one without liquidity.

In most of our simulations, the additional liquidity accuracy tends to fade as the number of voters grows.

There are still some improvements to be made, specially about the user friendliness of the e-voting system. There should be a solution that keeps a balance between automation of the process and precision of the results through a non stop solicitation of the user.

With our survey, we obtained answers that tends to assert that circular delegation and tyrant problems could be, in reality application, false problems.

However a Liquid system comes with problems and limitations such as corruption, or if we want a full automation of the voting process, we should first see how to handle these problems if the user is not aware of them.

The project, being a from scratch project, has plenty of ways to be improved. We can upgrade the implementation to add a federated architecture to the system, improve the security and usability of the web application thought proper authentication and remote access (hosting the website). We can improve the website to be more user friendly, display additional information and options for the votation creation. We can also improve the simulations by adding more parameters such as the age of the voters.

Overall, we have seen that a implementation of a Liquid System is plausible and that there exists many solutions to interact with it.
If you would like to see the code of the project, it's here : LIVOS Github

Bibliography

[1] "Adhocracy". In: *https://adhocracy.plus/* ().

[2] "Electoral Fraud". In: *https://simple.wikipedia.org/wiki/Electoral*$_{fraud}$ ().

[3] Bryan Ford. "A liquid perspective on democratic choice". In: *arXiv preprint arXiv:2003.12393* (2020).

[4] Steve Hardt and Lia CR Lopes. "Google votes: A liquid democracy experiment on a corporate social network". In: (2015).

[5] "Instant Runoff voting". In: *https://medium.com/codex/better-voting-results-through-markov-chains-d8aa46c79870* ().

[6] Qi Kang, XiaoShuang Chen, SiSi Li, and MengChu Zhou. "A noise-filtered under-sampling scheme for imbalanced classification". In: *IEEE transactions on cybernetics* 47.12 (2016), pp. 4263–4274.

[7] "Markov Chains". In: *https://medium.com/codex/better-voting-results-through-markov-chains-d8aa46c79870* ().

[8] C Andrew Neff. "Verifiable mixing (shuffling) of ElGamal pairs". In: *VHTi Technical Document, VoteHere, Inc* (2003), p. 112.

[9] "The Principles of Liquid Feedback". In: *https://liquidfeedback.com/en/book-the-principles-of-liquidfeedback.html* ().

[10] Dr B Zhang. "Digital Liquid Democracy: Delegative End-to-end Verifiable E-voting". In: *http://gtr.rcuk.ac.uk* (2017).