Proxy Voting Coordination Mechanisms: Determining How Agents Should Coordinate in a Continuous Preference Space

Michael D. Hegerhorst
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Computer Sciences Commons

Recommended Citation
https://digitalcommons.usu.edu/etd/8877

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations, Spring 1920 to Summer 2023 by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
PROXY VOTING COORDINATION MECHANISMS: DETERMINING HOW AGENTS SHOULD COORDINATE IN A CONTINUOUS PREFERENCE SPACE

by

Michael D. Hegerhorst

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

Approved:

Vicki Allan, Ph.D. John Edwards, Ph.D.
Major Professor Committee Member

Shuhan Yuan, Ph.D. D. Richard Cutler, Ph.D
Committee Member Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah
2023
ABSTRACT

Proxy Voting Coordination Mechanisms: Determining How Agents Should Coordinate in a Continuous Preference Space

by

Michael D. Hegerhorst, Master of Science
Utah State University, 2023

Major Professor: Vicki Allan, Ph.D.
Department: Computer Science

Illness, injury, and other impediments are common occurrences of everyday life. Such impediments prevent or deter agents from participating in important parts of the voting process, in particular deliberation, bargaining, and the voting itself. Without participation, the results of the vote may change. There is a need to provide a mechanism by which agents are still able to participate in such processes to ensure their vote is represented. We examine single-vote/single-winner proxy voting in a one-dimension continuous preference space using a combination of $L_p$ aggregation methods. As part of this examination, we develop and examine ‘coordination mechanisms,’ by which proxies and their constituents are able to find a way to combine their preferences while allowing the constituents to still have their voices heard. In exchange, their proxy gains more weight and is able to have a stronger voice in deliberations. We employ a continuous preference space model and determine the result of a vote as a point in the model’s space. This model allows for more options than the vote only passing or failing by allowing the outcome to directly correlate to actions to be taken, such as the amount of money to spend. Such an ability increases the granularity of the results and allows us to better determine how much error is introduced by proxy voting.
The continuous preference space model also provides a more expressive way of casting a vote. We show that proxy voting is effective in many scenarios and that it is consistently better than not allowing inactive agents to vote.
PUBLIC ABSTRACT

Proxy Voting Coordination Mechanisms: Determining How Agents Should Coordinate in a Continuous Preference Space

Michael D. Hegerhorst

Illness, injury, and other impediments are common occurrences of everyday life. Such impediments prevent or deter voters from participating in important parts of the voting process, especially deliberation, bargaining, and the voting itself. Without participation, the results of the vote may change. There is a need to provide a system in which voters are still able to participate in important voting processes to ensure their vote is represented. We explore ‘proxy voting,’ a system in which voters are able to select another individual, or proxy, to vote on their behalf. By choosing a good proxy, a voter can still have their vote represented and cause minimal change to the outcome. We additionally explore different ways a proxy can coordinate with their peers, as well as different ways to count votes, in order to make proxy voting as effective as possible. We show that proxy voting is effective in many scenarios and that it is consistently better than not allowing voters who are unable to fully participate to have their voices heard.
To Danielle, my constant companion and eternal partner.
You are my light and my star, my guiding hope. I love you.

And to Miriam, who joined us in our adventure.
May you lift others up and help them to be as brilliant as you.
# CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT .......................................................... iii</td>
</tr>
<tr>
<td>PUBLIC ABSTRACT .................................................... v</td>
</tr>
<tr>
<td>LIST OF TABLES ...................................................... viii</td>
</tr>
<tr>
<td>LIST OF FIGURES ..................................................... ix</td>
</tr>
</tbody>
</table>

## 1 INTRODUCTION .................................................. 1
1.1 Background ....................................................... 1
1.1.1 What is proxy voting? ........................................ 1
1.2 Previous Work .................................................... 4
1.3 Preliminary Setup ............................................... 8
1.3.1 The Model ..................................................... 8
1.3.2 Voting Mechanisms .......................................... 10
1.3.3 Coordination Mechanisms .................................. 11
1.4 Assumptions ....................................................... 13
1.5 Contribution ..................................................... 14

## 2 ANALYSIS ...................................................... 17
2.1 Results ......................................................... 17
2.1.1 Mechanism Combinations .................................... 17
2.1.2 Inactive Penalty ............................................. 18
2.1.3 Effects of Voter Distribution ............................... 23
2.1.4 Preference Change .......................................... 25
2.2 Conclusions ..................................................... 27
2.3 Future Work ...................................................... 27

## REFERENCES ..................................................... 29

## APPENDICES ...................................................... 31
A Error by Distribution with a different scale .................. 32
B Error by Distribution zoomed in ............................... 34
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Definitions for the voting mechanisms used by [1]. $n$ represents the number of candidates for some vote.</td>
<td>6</td>
</tr>
<tr>
<td>1.2</td>
<td>The distributions to be used to generate preferences. Note how each distribution represents a population type. These types are representative, and any distribution could potentially represent a different population type that shares the same shape as the distribution. Additionally, any skewed distributions can be inverted to create a distribution that is skewed in the other direction (e.g. a distribution skewed in favor can be inverted to create a flipped distribution skewed against).</td>
<td>15</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Example of a 1D continuous preference metric space, where ( \top(x) ) represents the preference of agent ( x ). The x-axis represents some preference space. An agent can have a preference anywhere within this space. One way to interpret the model is to have the leftmost point be the most against some idea and the rightmost point is the most in favor of the same idea. Importantly, points toward the center of the space are the most ambivalent, neutral, or central about the idea. Alternatively, points toward the center may also prefer some type of compromise or alternative solution instead.</td>
<td>8</td>
</tr>
<tr>
<td>2.1</td>
<td>The absolute mean error as a percent of the preference space by coordination mechanism per voting mechanism. ‘Error as a percent of the preference space’ means the error divided by the total size of the preference space. Note All-Agents represents when all agents are voting, meaning no proxy voting is used and all agents are present. Similarly, Active-Only represents when proxy voting is not used and not all agents are present, and when proxies lose their constituents. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>19</td>
</tr>
<tr>
<td>2.2</td>
<td>The difference between all agents having the same weight and inactive agents having only ( \frac{1}{5} ) the weight as their proxies. The difference is in terms of distance as a percent of the preference space from all agents voting. ‘Distance as a percent of the preference space’ means the absolute difference divided by the total size of the preference space. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>22</td>
</tr>
<tr>
<td>2.3</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism and preference distribution. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>24</td>
</tr>
<tr>
<td>2.4</td>
<td>The absolute difference in the error produced between no preference change and with a preference change as a percent of space. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>26</td>
</tr>
<tr>
<td>A.1</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism and preference distribution. Note the y-axis does not have the same scale for each plot. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>33</td>
</tr>
<tr>
<td>A.1</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $B(0.3, 0.3)$. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>35</td>
</tr>
<tr>
<td>A.2</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $B(4, 1)$. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>36</td>
</tr>
<tr>
<td>A.3</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $B(50, 50)$. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>37</td>
</tr>
<tr>
<td>A.4</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $N(\mu, \sigma)$. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>38</td>
</tr>
<tr>
<td>A.5</td>
<td>The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $U_{[-1,1]}$. The results are averaged over 1024 total runs. There are 24 total agents.</td>
<td>39</td>
</tr>
</tbody>
</table>
CHAPTER 1
INTRODUCTION

From determining the best breakfast cereal [2] to electing the next president, voting is an extremely important aspect of modern society. Nevertheless, disease, injury, and other impediments can create difficulties for individuals participating in such democratic processes, preventing them from expressing their voice and participating in deliberation, as well as decreasing the overall quality of the result. Proxy voting is a method by which participants are able to have others vote on their behalf. We examine the ability of proxy voting to decrease the impact of such frustrations. We additionally determine strategies by which proxies and their constituents can cooperate in order to reduce the change their absence would otherwise cause. These strategies include allowing proxies and constituents to aggregate their preference into one result, as well as determining how best to aggregate all votes in a unified-vote/single-winner/single-dimension continuous space model.

1.1 Background

1.1.1 What is proxy voting?

Proxy voting is a group of methods by which individuals who are unable to vote in person can still have their voices heard through the use of a proxy, who is an individual authorized to act for another. Agents are able to delegate another individual to be their proxy. We call the delegating agent a delegator or an inactive voter, and the group of agents that delegate to a proxy its constituents. Proxies are also known as delegates or active voters.

When voting, each individual starts with a certain number of votes that they can allocate to any given option. Typically, this number is one. The number of votes a voter has is called its weight or voting power. For example, an individual with a weight of $n$ who
votes for $x$ has the same impact as if $n$ distinct voters with a weight of 1 picked $x$. When delegating a proxy, the delegator’s weight is transferred to the proxy, increasing their total weight by the amount of power transferred. The more weight a voter has, the more they can swing the vote in their favor.

Upon selecting a proxy, a delegator is no longer able to vote directly. Instead, their delegate votes on their behalf. In contrast, direct voting requires each agent to vote on their own, meaning each agent must incur the costs of voting or not have their voice heard. These costs may be tangible, such as needing to pay for gas, or intangible, such as the time or effort required to vote in person. Such costs are common in voting and are further discussed by [3]. Error is introduced into the result of direct voting when an agent is unable or unwilling to pay these costs since information is lost when agents do not share their preferences via a vote. In votes with discrete options, this error could be anything from a different result (in the worst case) or a slightly different count of the votes (for example, 10 votes in favor instead of 11). In these cases, such as in times of injury or illness, proxy voting provides a potential avenue through which the agent can still have its voice heard and reduce the error in the system.

Proxy voting is beneficial for the delegates as well. By working on behalf of their constituents, a proxy has a larger voice in discussions and deliberations since they are representing more agents. This allows them to have a larger influence and achieve a greater impact as topics are debated.

Proxy voting systems are not, however, without flaws. When agents do not participate in person, they are not able to participate in deliberation. Such discussions are vital to the voting process, since as agents confer, they gain access to new information which may change their preference. Inactive agents do not benefit from this deliberation; only active agents do. As such, proxies have to participate in deliberation and change their preference on behalf of their constituents.

There are multiple ways a proxy voting system can allow its proxies to vote. Specifically, they can allow proxies to make only one selection or multiple. When the proxy is only
allowed to make one selection on behalf of all its constituents, those constituents will have their vote applied in a way they find less preferable than simply paying the costs. In this ‘unified-vote’ model, an agent must choose a proxy they are confident would change their preference in the same way they would ‘if only [they] had the time and knowledge to participate directly’ [4], or risk having their vote misallocated. A vote is misallocated when the proxy changes its preference in such a way that some of its constituents would not, or when the proxy and its constituents choose a preference that is different from an individual’s preference. When a vote is misallocated, error is again introduced into the system, and may arguably be worse than if the inactive agent hadn’t voted at all since it may change the result of the vote from what it would with all agents participating.

As an alternative to only casting one vote on behalf of all constituents, the proxy could be allowed to vote once per constituent, and can even be required to vote precisely as each constituent requests. This is the system the 116th United States Congress introduced in May 2020 in an attempt to reduce the risk of COVID-19 in the House of Representatives [5,6]: the proxies effectively ‘relay’ their delegators’ votes individually and for each inactive voter. As such, information about the inactive voters’ preferences is not lost. The process is effective in that every voter will have their vote allocated exactly as they want, but comes with the obvious cost that each proxy has to keep track of how each individual constituent wants their vote relayed. This naturally increases the complexity and work required by the proxy since they not only need to know for whom they vote, but also what their preferences are and keep up-to-date with those preferences. Additionally, the inactive voters still need to be active in the deliberation process in order to properly participate in the process. This makes ‘relay-voting’ only effective for occasions when agents have been able to participate in the full process except for the voting itself and know exactly how they want their vote allocated.

In this study, we will tackle some of the problems associated with unified-vote proxy voting. By determining methods to minimize the error produced by the model, the system will yield the benefits of unified-vote deliberation while still producing a near-perfect result.
1.2 Previous Work

Proxy voting is a well-discussed topic, and we build off many authors’ previous work.

Jonas Degrave [7] implemented a simple model to calculate the weight of a proxy. The model even allows agents to delegate multiple proxies. He treated delegations as a digraph where nodes are voters and edges represent a delegation. He then created two methods to determine the weights of each proxy. The first method constructs a linear equation for each agent, consisting of the agent’s original weight plus the weight of the agents that delegate to it. The second employs an adjacency matrix and uses linear algebra to calculate the solution. Ultimately, both methods simply sum the total weight allocated to a proxy. We will employ the first technique since it allows for a straightforward way to delegate voting power that would not be confusing to voters. However, we will not take full advantage of it by disallowing proxies to delegate to other proxies, as well as only allowing each agent to delegate one proxy. Such proxy-to-proxy delegation is known as liquid democracy, which has its own advantages and challenges. Unfortunately, it can have the effect that delegates might not know who wields their voting power, and so we will focus on enhancing deliberation in unified-vote/single-winner/single-dimension continuous space scenarios.

Related to weight, Zhang and Grossi [8] explore treating weight as a probability instead of a count, in addition to simply transferring voting power to the proxy. In the probability model, inactive agents spread their weight out amongst several proxies. This probability represents the odds the agent would delegate their full weight to that proxy. The probability model allows Zhang and Grossi to probabilistically predict the outcome of a vote, and use that model to see if agents can correctly identify some world state. They assume that any agent is able to delegate to any other agent. We share this assumption, though we will only focus on transferring voting power directly instead of treating it as a probability.

Anurita Mathur and Arnab Bhattacharyya [1] looked at several voting mechanisms applied on a single-winner election vote and determined a ranking for these mechanisms. Voting mechanisms, also known as aggregation mechanisms, are algorithms that take the preferences of all active voters and turn them into the output of the system. They apply
these mechanisms on a dataset while looking only at data points without a Condorcet winner. A Condorcet winner is a candidate that wins in every possible one-on-one election against other candidates. These types of winners are important because they are ubiquitously preferred over all other candidates in an election, and their absence means there is not always a clear winner. In their work, they say a mechanism ‘beats’ another if it has a larger fraction of the population prefers its output over the other’s output. They discover that the maximal lottery rule, also known as the Game Theory (GT) method [9], beats all others. The Schulze method [10] and Minimax voting mechanisms always agree and beat all other mechanisms besides the GT method, while Borda beats Copeland and Plurality, while Plurality comes in last. These mechanisms are briefly described in Table 1.1. This study will also look at voting mechanisms and attempt to determine which mechanism is best suited for proxy voting.

Our work will differ significantly from theirs, however, as we will explore voting mechanisms used in a continuous voting space instead of a discrete space. Additionally, our mechanisms will be different from the ones they employed since their mechanisms either do not work in a continuous preference space or do not work well with proxy voting.

Cohensius et al. [11] explore the use of proxy voting in a continuous metric space using two voting mechanisms: mean and median. They also explore proxy voting in a binary space using the majority mechanism. They discovered proxy voting using any of these mechanisms generally produces lower error than direct voting with active voters alone. This is not too surprising: reintroducing information lost through inactive voters, regardless of the method, ought to help the system. Nevertheless, they were able to show that proxy voting is effective under a number of symmetrical and asymmetrical preference distributions, while under both random and strategic participation. However, the majority of their research focuses on voting with infinite populations. While our work would certainly be applicable to larger populations (since a population of sufficient size will begin to behave like an infinite population),¹ we are more interested in more realistic proxy voting in smaller, finite

¹Naturally, \( \lim_{x \to \infty} x = \infty \).
Table 1.1: Definitions for the voting mechanisms used by [1]. \( n \) represents the number of candidates for some vote.

<table>
<thead>
<tr>
<th>Voting Mechanism</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Borda</td>
<td>Agents rank each candidate, 1 to ( n ). The candidate ranked first receives ( n - 1 ) points, the candidate in second ( n - 2 ), etc. Whichever candidate receives the most points total wins.</td>
</tr>
<tr>
<td>Copeland</td>
<td>Agents rank candidates 1 up to ( n ). Agents are able to leave some blank, all of which will be counted as last place. After all agents have ranked the candidates, they are compared pairwise. If a candidate is more often ranked better than another, it gets one point. If they are more often ranked worse, they get 0 points. When the number of ranked better vs ranked worse is equal, the candidate receives half a point. The candidate with the largest score wins.</td>
</tr>
<tr>
<td>GT</td>
<td>Agents rank each candidate, and a pairwise margin matrix is generated. In the matrix, position ((x, y)) is the number of rankings that prefer ( x ) over ( y ), minus the number of rankings that prefer ( y ) over ( x ). A zero-sum game is defined, and an optimal mixed strategy is computed. A ‘zero-sum game’ is a situation in which for every point gained by one side, the other side loses an equal number of points. A ‘mixed strategy’ is when an agent does not always choose the same move given a situation; instead, they probabilistically choose between the options. An ‘optimal mixed strategy’ is thus the mixed strategy that yields the most utility, or, in other words, achieves the best results. The winner is then chosen randomly using the optimal mixed strategy. See [9] for more details.</td>
</tr>
<tr>
<td>Minimax</td>
<td>Agents rank every candidate from 1 to ( n ). Candidates are compared pairwise, where candidate ( x ) is compared to candidate ( y ). The score of each comparison is the number of votes candidate ( y ) receives more than candidate ( x ). Then, the maximum score for each candidate ( x ) (meaning, its largest pairwise defeat) is determined. The candidate with the lowest (or minimum) maximum score is selected as the winner.</td>
</tr>
<tr>
<td>Plurality</td>
<td>Each agent selects their favorite candidate. The candidate with the most votes wins.</td>
</tr>
<tr>
<td>Schulze</td>
<td>Agents rank the candidates and are allowed to skip rankings, give the same rank twice, or not rank a candidate. Candidates with the same ranking are said to be equally preferred, and unranked candidates are ranked last. Then, a weighted directed graph connecting agents is generated. An edge going from candidate ( A ) to candidate ( B ) with a weight of 10 means 10 more agents prefer ( A ) over ( B ) than ( B ) over ( A ). Using this graph, multiple paths are generated between candidates. The strength of the path is the weight of the weakest edge in the path, and the path with the highest strength from ( A ) is compared to the strongest path from ( B ). The same occurs with paths from ( A ) to ( C ), ( B ) to ( C ), etc. The candidate that has stronger paths to each individual agent than all other agents wins.</td>
</tr>
</tbody>
</table>
populations. As such, we will explore the effects of proxy voting on a finite population, as well as explore other possible voting mechanisms, inside this continuous metric space.

Bulteau et al. [12], develop and experiment with a framework for aggregating preferences in several spaces using $L_p$ mechanisms. $L_p$ aggregation methods work by minimizing the sum of distances to the power of $p$ ($d^p$, where $d$ is a distance) between a possible solution and the voters’ preferences. These mechanisms are particularly useful because they allow fine-tuning of the aggregation method by changing $p$. Specifically, in single-dimension single-winner continuous models, they employ $L_1$ (median), $L_2$ (mean), and $L_\infty$ (mid-range, meaning the point between the highest and lowest agent preference). They additionally treat any possible value in the space as a potential output of the system. They provide a number of remarks and observations for each of these mechanisms, which we will be borrowing. However, they examine these mechanisms without weighted votes, which we will have. As such, we will adapt each of the mechanisms to use weighted votes so they can work with proxies, and examine how they operate in the continuous space.

James Miller [4] imagined a governmental system utilizing proxy voting in 1969 as a more direct form of representative democracy.\(^2\) His work focuses on reworking the current House and Senate systems entirely by using a more directly involved populace, but his ideas can still be relevant under the current system. In particular, he introduces the idea we call expert proxies, those being individuals who would ‘vote as [the delegator] would if only [the delegator] had the time and knowledge to participate directly’ [4]. Additionally, Miller states ‘a representative should be an expert, or at least competent, in each field [on which they are voting]’ [4]. This is true both in the government as well as in other situations where decisions are made by voting. As such, we consider scenarios where a proxy is dubbed, at least according to its constituents, an ‘expert,’ and so the constituents will go with its preference instead of their own.

\(^2\)That is to say, Miller envisioned a system where individuals could directly vote for an issue, or elect a proxy to vote for them. Naturally, any democracy that uses proxy voting is a representative democracy, since the proxy is representing the delegator. Nevertheless, it can be argued that Miller’s proposal could provide a more direct democracy since a voter can directly vote for an issue if they so choose.
1.3 Preliminary Setup

1.3.1 The Model

An important part of any study is the model it uses to represent the system being studied. We employ a model described by Cohensius et al. in their 2017 article [11]. This model places voters’ preferences in a single-dimension continuous metric space $\chi$, such as in Figure 1.1. In this model, two points that are close together in the metric space represent similar preferences, while two points that are far apart represent very different preferences. As a point moves further away from the agent’s preference, the agent likes it less. This model works best when an upper and lower bound is provided, such as only allowing agents to vote in the interval $[-1, 1]$, to prevent agents from voting extremely far in either direction and so potentially biasing the vote. Not all methods of voting are susceptible to such attacks, but it must be a consideration for those that are.

The difference between multiple equidistant points in the model may not be equivalent to the agents. For example, an agent who must spend $30 may prefer spending more money rather than less. As such, even though $29 and $31 are equidistant to $30, an agent may vastly prefer $31 over $29. However, for our model, we will assume agents only care about the distance from their preference, rather than if it is greater or less than some amount.

![Fig. 1.1: Example of a 1D continuous preference metric space, where $T(x)$ represents the preference of agent $x$. The x-axis represents some preference space. An agent can have a preference anywhere within this space. One way to interpret the model is to have the leftmost point be the most against some idea and the rightmost point is the most in favor of the same idea. Importantly, points toward the center of the space are the most ambivalent, neutral, or central about the idea. Alternatively, points toward the center may also prefer some type of compromise or alternative solution instead.](image-url)

Such a model is extremely flexible and can be interpreted in different ways. When applied to binary for-or-against voting problems, options can be placed at either extreme
of the interval. As an example, shareholders at some company are voting on new data collection policies. Using the interval \([-1, 1]\), we can place ‘no collection at all’ at -1, and ‘heavy collection’ at 1. Agents vote according to their preference: -1 for no collection, 1 for heavy collection. So far, everything works the same as a normal vote. However, due to the continuous nature of the voting space, the agents can vote anywhere in the given interval. Therefore, agents who do not care one way or the other can vote at 0 instead of being forced to choose an option. Additionally, those that are only slightly in favor (meaning they would prefer collection but do not really care that much) can choose some value between 0 and 1. Once the votes are aggregated, the result can be rounded to whichever option is closest. The continuous space allows agents to better express themselves according to what their actual preference is, instead of being forcibly binned into one value or the other.

Alternatively, if the majority of the votes are about the center, it may be a sign neither option is satisfactory. Using the data collection example, we can reinterpret 0 to mean the agents do not mind collecting users’ data, but also do not personally care either way. These agents would vote around 0 to avoid imposing. By making the agents aware of how 0 will be interpreted, they can express their dissatisfaction by voting at or around 0. Normally, voting goes through a bargaining and an enforcement phase [13], but if a sufficient number of agents vote close to 0, the group can reopen discussion about the topic and re-bargain before conducting a new vote. This enhances the cooperation aspect of voting, and creates a fundamental change: voting is no longer the end of the negotiation, but rather part of it.

Finally, the continuous space model allows for another, very powerful interpretation: interpreting the votes and result as continuous values. For example, imagine Congress is voting on how much to allocate to the defense budget. In this scenario, Congress can decide to allocate anywhere between $0 and $1,000.\(^3\) A voter can place their vote anywhere between those values, and the result of the vote can be how much Congress allocates. Say there are three voters, each with their own preferences, and all voters are active (no voter delegates their vote). Agent \(a\) prefers $750, \(b\) prefers $600, and \(c\) prefers $250. One way we could aggregate these preferences is by finding their mean. By averaging these preferences,
the system would output \( \frac{8(750+600+250)}{3 \text{ voters}} = \frac{81600}{3 \text{ voters}} = 533.33 \), which would be the amount allocated to the defense budget. Being able to vote on continuous problems and yield a continuous output, as well as working with binary issues, makes the model extremely flexible and allows it to tackle any number of problems.

We are also able to easily calculate error using a continuous space. The error can simply be the distance from the result of the system when all agents are active, and the result under proxy voting.

The flexibility of the continuous model in both the discrete and continuous realms, its ability to use different voting rules, easy interpretability, and easy error calculation are the reasons it is employed in this study. We will focus primarily on the continuous instead of the discrete output of the model, since observing the continuous output allows for more granularity in the differences between proxy and non-proxy voting, as well as in the differences between voting rules.

1.3.2 Voting Mechanisms

This study also makes use of voting mechanisms or voting rules, which are functions that map a set of preferences in \( \chi \) to an outcome that also exists in \( \chi \). For these, we take inspiration from Bulteau et al.’s [12] work in aggregating one-dimensional single-winner elections by using their \( L_p \) aggregation methods, as well as mixing in plurality. \( L_p \) aggregation methods work by minimizing the sum of distances to the power of \( p \) (\( d^p \), where \( d \) is a distance) between a possible solution and the voters’ preferences. Naturally, since [12] did not use weighted proxies, these methods need to be adjusted to allow for weight. Below, \( w(a) \) represents the weight of an agent \( a \), and \( W \) is the set of active voters ordered by preference. Additionally, \( \top(a) \) is the preference of an agent, and \( S \) is the set of all agents. With these notations, the mechanisms we use are

1. Median \( (L_1) \), defined as let \( i = \min i \text{ s.t. } \sum_{j=1}^{i} w(W_j) \geq \frac{w(S)}{2}; \text{md}(W) = \top(a_i) \).

Essentially, the median is the active agent whose additional weight makes the sum of weights become equal to or greater than half the total weight of the system. The sum
will occur in the same order as the ordering of voters in \( W \). There is an edge case where the sum of weight perfectly reaches half the total weight. When this occurs, the preference that reaches half is called the lower weighted median, and the preference that starts from half is called the higher weighted median. In this case, both the lower and higher weighted medians, as well as any value in between, can claim to be the median. This is similar to when there are an even number of preferences in a normal median. In this case, we take the average between the preference of the agent with the highest preference in the lower median half and the agent with the lowest preference in the higher half of the weighted median.

2. Mean (\( L_2 \)), defined as \( \text{mn}(W) = \frac{1}{w(W)} \sum_{a_i \in W} w(a_i) \cdot \mathbb{T}(a_i) \). This is a typical weighted average.

3. Mid-range (\( L_\infty \)), defined as \( \text{mr}(W) = \frac{\mathbb{T}(W_1) + \mathbb{T}(W_n)}{2} \), where \( n \) is the number of proxies
   This equation essentially means the preference of the active agent with the lowest preference plus the preference of the active agent with the highest preference, divided by two.

In addition to applying these voting mechanisms on the set of active agents, we apply these mechanisms to active voters without weights (to simulate if proxy voting was not allowed) and all voters, both active and inactive, (to simulate the best case where all agents vote). These additional calculations serve as baselines to determine how well proxy voting works in these scenarios.

1.3.3 Coordination Mechanisms

There are also several ways an individual proxy can agree to cast its vote on behalf of its constituents. Each has different advantages and disadvantages for the proxy, its constituents, and the system as a whole. We will examine five different ‘coordination’ mechanisms that ‘groups’ (meaning a proxy and its constituents) can use:

1. Expert. The group applies its total weight to the proxy’s preference.
2. *Cooperative Mean.* The group allocates its weight to the mean of the proxy’s and its constituents’ preferences.

3. *Cooperative Median.* The group allocates its weight to the median of the proxy’s and its constituents’ preferences.

These mechanisms are used in an attempt to simulate real-world consequences of proxy voting, as well as identify potential techniques to deter or mitigate a proxy’s ability to swing the system by aggregating a large amount of weight and abusing it.

These mechanisms will additionally be applied after active agents change their preferences in order to determine how well they represent the agents after deliberation has occurred.
1.4 Assumptions

In order to conduct this study, we make a number of assumptions. First, we assume agents vote on only one topic at a time, and each topic is completely independent. In other words, the votes cast, results, and proxies used for one topic do not affect any other topic. This allows agents to select a proxy that works best for the current topic instead of selecting the best proxy for all topics.

Additionally, we only consider scenarios where unified-vote proxy voting is used, where the proxy receives the voting power of the delegating voter, increasing their weight, and makes one selection instead of relaying their constituents’ vote. This type of proxy voting also allows the proxy to update their (and by extension, their constituents’) preferences as new information becomes available. While unified-vote proxy voting gives substantial flexibility to the proxy to operate on behalf of their constituents, this flexibility requires the delegating voter to choose a proxy who they trust to vote as close to how they themselves would. This is a process similar to selecting experts, as described by [4] and [14]. By using unified-vote proxy voting instead of relay-style voting, we hope to exploit the advantages of proxy voting that the relay-style does not provide as described in section 1.1.

We also assume that each voter has reasonable knowledge about potential proxies’ opinions. This will allow them to choose the proxy that has an opinion most similar to their own. While in reality voters will likely not have perfect knowledge of others’ opinions, it is often not difficult to gauge the opinion of others. This is particularly true for individuals with whom a person often associates, and so we believe this assumption is reasonable.

Finally, we assume that there are no factors besides closeness in opinion that affect the choice of proxy. This differs from some systems, such as that presently used by the House of Representatives which includes restrictions such as a proxy can only serve ten voters [5]. However, we feel removing restrictions such as these leads to a more interesting discussion, since it allows the use of different voting mechanisms and more extreme cases.
1.5 Contribution

We explore proxy voting in a unified-vote/single-winner single-dimension continuous space model. We employ three well-known $L_p$ mechanisms, specifically

1. Median ($L_1$)
2. Mean ($L_2$)
3. Mid-range ($L_\infty$)

Each of these mechanisms will additionally be applied to direct voting with all agents and direct voting with only those agents that are present. This is done to show what the output of the system would be with all information (direct voting with all agents), as well as the output with minimum information (direct voting with only those agents that are present). The error will be the distance between the result and direct voting with all agents.

We additionally apply what we’ve dubbed ‘coordination mechanisms,’ which are techniques by which proxies and constituents work together to determine how their weight will be allocated, meaning how the proxy will vote on their behalf. The mechanisms we explore are described in subsection 1.3.3. These mechanisms are used in an attempt to simulate real-world consequences of proxy voting, as well as identify potential techniques to deter or mitigate a proxy’s ability to swing the system by aggregating a large amount of weight and abusing it by misallocating it towards a vote the constituents do not prefer.

Whereas voting on a continuous interval is uncommon and finding a real-world dataset using preferences on an interval currently does not seem possible, these investigations are performed using preferences generated from several statistical distributions. We use various distributions to allow the gathered data to represent different distributions of voters in the real world. These distributions include well-known statistical distributions, such as the uniform distribution, the Gaussian distribution, as well as a few beta distributions. The distributions used and their notations are listed in Table 1.2.

Initial experiments show larger populations behave similarly to those with fewer. As such, each experiment will have a population of 24. We chose this number because prior
Table 1.2: The distributions to be used to generate preferences. Note how each distribution represents a population type. These types are representative, and any distribution could potentially represent a different population type that shares the same shape as the distribution. Additionally, any skewed distributions can be inverted to create a distribution that is skewed in the other direction (e.g. a distribution skewed in favor can be inverted to create a flipped distribution skewed against).

<table>
<thead>
<tr>
<th>Distribution</th>
<th>Notation</th>
<th>Symmetrical?</th>
<th>Population Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform</td>
<td>$U_{[-1,1]}$</td>
<td>✓</td>
<td>Evenly spread</td>
</tr>
<tr>
<td>Normal/Gaussian</td>
<td>$\mathcal{N}(0, 1/3)$</td>
<td>✓</td>
<td>Mostly centrist/indifferent</td>
</tr>
<tr>
<td>Beta(0.3, 0.3)</td>
<td>$\mathcal{B}(0.3, 0.3)$</td>
<td>✓</td>
<td>At either extreme</td>
</tr>
<tr>
<td>Beta(50, 50)</td>
<td>$\mathcal{B}(50, 50)$</td>
<td>✓</td>
<td>Strongly centrist/indifferent</td>
</tr>
<tr>
<td>Beta(4, 1)</td>
<td>$\mathcal{B}(4, 1)$</td>
<td>✓</td>
<td>Skewed in favor</td>
</tr>
</tbody>
</table>

experimentation shows it to be a good balance between having few enough agents that an agent going inactive can change the results, while not so few that an agent going inactive is catastrophic. Since having no delegates or all delegates would not be interesting, we will experiment with 1 to 22 inactive agents and examine how error correlates with the number of delegators. We chose 1 inactive agent as the minimum because the case with 0 agents is not interesting due to there not being any proxies. Similarly, we chose 22 as the maximum because the case with 24 agents means there are no active agents to vote, and the case with 23 simply devolves into the coordination mechanism replacing the voting mechanism.

For each experiment, agents will randomly select some preference using the current distribution. Afterward, one agent will become inactive and choose the closest active agent to be their proxy. Following this, the proxies and their constituents will apply a coordination mechanism, and each voting mechanism will be applied. The next coordination mechanism will be used, then each voting mechanism will be applied again. Once all coordination mechanisms have been used, the number of inactive agents will increase until 22 agents are inactive. Finally, agents will select a new preference from the next distribution and the process will begin again. This will continue until all possible combinations have been used. Each combination is run 1024 times to ensure accuracy in the analysis.

We will show that proxy voting with the right combination of mechanisms generally
yields considerably lower error than active-only voting. We also show proxy voting is beneficial even when agents’ preferences change. However, proxy voting appears to be least effective on highly-polarized topics.
CHAPTER 2
ANALYSIS

2.1 Results

2.1.1 Mechanism Combinations

The absolute mean error as a percent of the total preference space by coordination mechanism per voting mechanism is graphed in Figure 2.1. The absolute mean error as a percent of the total preference space $|MSE\%|$ is defined as $|MSE\%| = \frac{1}{n(\chi_{\text{max}} - \chi_{\text{min}})} \sum_{i=1}^{n} |\epsilon_i|$, where $n$ is the number of runs, $\chi_{\text{max}}$ and $\chi_{\text{min}}$ are the maximum and minimum values in the preference space, respectively, and $\epsilon_i$ is the error of run $i$. Error is defined as $\top_{(\text{test})} - \top_{(\text{baseline})}$, meaning the output of the test run minus the output of the baseline (all agents voting).

From this plot, there are a few trends that are immediately identifiable. First, error unsurprisingly increases as the number of inactive agents increases.

We can also see the Mean and Median coordination mechanisms generally yield similar amounts of error. In particular, when either of these mechanisms is combined with the Mean voting mechanism the results are very similar to when all agents are active and able to vote.

Notably, the Expert coordination mechanism exhibits higher error than the others as the number of delegators reaches a significant portion of the population. In particular, the Expert mechanism tracks nearly identically to Active-Only when using the Midrange voting mechanism. Depending on the use case, this may or may not be beneficial. If we are attempting to find the best result for all agents, this mechanism may yield undesirable results. However, if we are attempting to exploit the expert’s experience, as described by [4], this error could actually be considered an improvement over the original result. This
is because the experts were able to influence and change the vote, presumably bringing it
closer to what their expertise dictates. Nevertheless, when the number of delegators is low,
the Expert mechanism works similarly to the other coordination mechanisms.

Importantly, it is clear that using proxy voting, regardless of the mechanisms used, is
better than losing information by not allowing inactive agents to vote in any way.

2.1.2 Inactive Penalty

What if we were to introduce some penalty for the inactive agents? Since they aren’t
participating as much, perhaps it would be fair that their voice is not as strong. We
repeated the same experiments as above with different weights between a proxy and their
constituents. Specifically, the proxies maintain their weight of 1, while their constituents
have their weight reduced to \( \frac{1}{5} \) (0.2) of their original weight. We choose \( \frac{1}{5} \) because it
still allows constituents to have some impact while still having their influence significantly
reduced. The difference between all agents voting and proxy voting of such a setup are
shown in Figure 2.2.
Fig. 2.1: The absolute mean error as a percent of the preference space by coordination mechanism per voting mechanism. ‘Error as a percent of the preference space’ means the error divided by the total size of the preference space. Note All-Agents represents when all agents are voting, meaning no proxy voting is used and all agents are present. Similarly, Active-Only represents when proxy voting is not used and not all agents are present, and when proxies lose their constituents. The results are averaged over 1024 total runs. There are 24 total agents.
Figure 2.2 shows there is an immediate difference from when all agents have the same weight. This is because, due to the weight, there is a shift towards the proxies’ preferences instead of the inactive voters’. Depending on the situation, this may be desirable. Since the inactive agents are not participating they may not have as much information and so their preference may not be as ‘up-to-date’ as those who are actively participating. In situations where new information is constantly being introduced, such penalties could be very valuable to ensure the most current information is being used.

Notably, the Median voting mechanism suddenly takes a sharp dip and produces a result closer to all agents voting when using different weights than when using the same weight. This is strange since it is completely different compared to how any other combination, or even the same combinations with less inactive voters, works. With some inspection, one can find this dip is due to active voters having a significantly larger influence compared to inactive voters. With this setup, an active voter is worth 5 times the amount of weight than an inactive voter. Therefore, when an active voter’s weight is added to the sum it is more able to push the sum over the half-way mark, thereby making that active voter’s preference the median. This is also why we see the most extreme dip when using the Median voting mechanism with the Median coordination mechanism: the proxy is able to reach the median both when coordinating and when voting, thereby making the proxy and its constituents’ vote be the proxy’s preference. This makes the system less volatile by making it less likely the output is an inactive voter’s preference, and so we avoid the large spike in error when a high percent of agents are delegating.

Additionally, from Figure 2.2, we see both the Median and Mean voting mechanisms quickly increase in difference before beginning to level out. This is again due to the difference in weight between the active and inactive voters. When both types of voters have the same weight, the coordination mechanisms help ensure the loss of preference from the agent going inactive is minimal. However, when they have different weights, in particular when the weight of the inactive voters is substantially less than that of the actives, the results are more akin to when the voter does not vote.
The Midrange voting mechanism, however, does not increase as quickly and does not achieve as much of a difference. Midrange, or $L_\infty$, takes the highest and lowest voting and averages the two. This means it ignores weight, which is likely why the change in weights does not affect it as much. Any difference in the outcome, then, must come from the coordination mechanisms used prior to the final vote being calculated with Midrange.

Notably, the Median mechanisms, both voting and coordination, yields the largest change. Mean is close behind, and Expert yields slightly less. Since the Median voting mechanism has already been shown to be difficult to manipulate [15], using both a Median voting and Median coordination mechanism would likely be a good choice when choosing which mechanisms to use.
Fig. 2.2: The difference between all agents having the same weight and inactive agents having only 1/5 the weight as their proxies. The difference is in terms of distance as a percent of the preference space from all agents voting. ‘Distance as a percent of the preference space’ means the absolute difference divided by the total size of the preference space. The results are averaged over 1024 total runs. There are 24 total agents.
2.1.3 Effects of Voter Distribution

Figure 2.3 shows the absolute error as a percent of the preference space by coordination mechanism for each voting mechanism and preference distribution. Figure A.1 shows the same graphs, but with a different scale for each plot to more easily see the shape. Additionally, each graph is zoomed in for readability in Appendix B.

Unsurprisingly, the tighter the probability density function (PDF) of a distribution, the less error it has overall due to many votes being close to each other. $B(50,50)$ and Normal are two such distributions. Nevertheless, these plots show the increase in error is consistent regardless of the mechanisms or distributions used. Mean and Median coordination mechanisms show a similar amount of error, while Expert shows slightly more except when a large portion of the population is delegating and when using the Midrange voting mechanism. Interestingly, this seems to indicate again that it does not matter how you do proxy voting so long as it is done. Additionally, it should be noted the mechanisms perform well on most preference distributions, including the asymmetrical distribution $B(4,1)$.

Unfortunately, while the error for other distributions is relatively similar, $B(0.3,0.3)$ yields considerably more error and starts accumulating this error earlier when using the median voting mechanism. $B(0.3,0.3)$ represents a highly polarized topic, with its probability density function having large spikes at either end of the preference space, making it extremely important to ensure results are fair and accurate. This would indicate that proxy voting, while it is still better than active-only voting, does not work quite as well when agents have stronger, opposing opinions.

For the median mechanism voting mechanism in $B(0.3,0.3)$, the increase in error makes sense. The median will most often be the preference of a specific agent, instead of some value in between. As such, the result will often be in one of the spikes in the PDF, since that is where the majority of the agents’ preferences will be, yielding higher error than when using the Mean or Midrange voting mechanisms. With the other voting mechanisms, the error is similar to other distributions using the same mechanisms until the vast majority of the agents are inactive. The increase in error is because as more agents go inactive there
Fig. 2.3: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism and preference distribution. The results are averaged over 1024 total runs. There are 24 total agents.
are fewer agents able to serve as proxies, and so there may not be as desirable of a proxy to represent each agent.

This finding is particularly important, as often the most polarizing topics are the most important to individuals. While it is still more beneficial to use proxy voting than not, in these polarized cases, agents should make an extra effort to participate in person to learn all they can about the topic and ensure their voice is heard. Those using proxy voting on split issues ought to be aware of this weakness and implement some limitations to prevent inaccurate results. These limitations could include requiring some percentage of the population, say 20–30%, to be active. Alternatively, if it becomes apparent that the discussion is changing the opinions of the active voters, which could be learned by occasionally polling the participants, it may be wise to require voters to attend in person. That way, those who have not been present to attend the deliberations can properly participate.

2.1.4 Preference Change

Figure 2.4 shows the difference in error between when proxies do have a preference change versus when they do not. In this case, the proxies were able to shift their preferences by up to 10% of the total preference space, while delegators kept their original preference. This represents the proxies changing their preference for an issue after deliberation. While this movement is random, if an agent is an extremist, meaning they are far to the left or right of the preference space, they are more likely to move towards the center.

Unsurprisingly, the amount of error increases as more agents become inactive. However, the coordination mechanisms are much more tightly grouped than in other scenarios. We also see all voting mechanisms actually yield worse error than active-only voting when using the Median and Mean coordination mechanisms. This might be because delegators are selecting proxies who then shift a large amount, leaving the delegator unsatisfied.

This potentially worse outcome highlights not only the importance of actively participating in discussions, but also the importance of selecting a proxy who would act as the delegator would want them to act.
Fig. 2.4: The absolute difference in the error produced between no preference change and with a preference change as a percent of space. The results are averaged over 1024 total runs. There are 24 total agents.
2.2 Conclusions

We have shown proxy voting is able to produce low-error results, even when the delegating portion of the population is large. We have additionally shown the Mean voting mechanism with the Mean and Median coordination mechanisms achieve the lowest error. These mechanism combinations are generally able to produce results with less than a 5% change in outcome when the delegating portion is more than half the total population and produce considerably less when the delegating portion is lower.

We have additionally shown proxy voting is at its weakest with highly polarized topics, such as those represented by $B(0.3, 0.3)$. In these cases, agents should make an extra effort to participate in deliberation and vote in person instead of by proxy.

However, while proxy voting certainly makes it easier for agents to be represented, agents should be strongly encouraged or required to participate in deliberations that may alter their preferences. When a proxy performs this work on behalf of an agent, the outcome of the vote may introduce more error than would otherwise be seen with only active voters participating.

Finally, we have shown that proxy voting is a powerful tool that consistently performs better than not allowing inactive agents to vote. By employing proxy voting, systems will be able to maintain their accuracy while increasing the total system utility.

2.3 Future Work

This study has investigated different methods to aggregate proxy votes, as well as determined techniques to allow proxies and their constituents to work together. Nevertheless, several additional avenues could be further investigated.

While we have shown it is better to use proxy voting than to not allow inactive agents to vote at all, proxy voting has clear weaknesses. Specifically, we have shown proxy voting is not as beneficial for polarized topics and when deliberations change agents’ preferences. We have suggested potential techniques that may help with these weaknesses, specifically introducing a penalty for not being active and limiting the number of agents that are allowed to be inactive. However, these techniques should be investigated to determine their
merit. Other error mitigation techniques should also be investigated to ensure proxy voting remains a beneficial tool.

Finally, more coordination mechanisms and voting mechanisms would be beneficial. We primarily used previously known methods to aggregate votes, but research may find a technique that would be able to negate the downsides of the aforementioned weaknesses of proxy voting.
REFERENCES


APPENDICES
APPENDIX A

Error by Distribution with a different scale
Fig. A.1: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism and preference distribution. Note the y-axis does not have the same scale for each plot. The results are averaged over 1024 total runs. There are 24 total agents.
APPENDIX B

Error by Distribution zoomed in
Fig. A.1: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $B(0.3, 0.3)$. The results are averaged over 1024 total runs. There are 24 total agents.
Fig. A.2: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $B(4, 1)$. The results are averaged over 1024 total runs. There are 24 total agents.
Fig. A.3: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $B(50, 50)$. The results are averaged over 1024 total runs. There are 24 total agents.
Fig. A.4: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for $\mathcal{N}(\mu, \sigma)$. The results are averaged over 1024 total runs. There are 24 total agents.
Fig. A.5: The absolute error as a percent of the preference space by coordination mechanism for each voting mechanism for \( \mathcal{U}_{[-1,1]} \). The results are averaged over 1024 total runs. There are 24 total agents.