Convergence Time of the Recombination Markov Chain on Small Planar Graphs

Emma Kolesnik

Follow this and additional works at: https://scholarship.claremont.edu/scripps_theses

Recommended Citation
https://scholarship.claremont.edu/scripps_theses/1739

This Open Access Senior Thesis is brought to you for free and open access by the Scripps Student Scholarship at Scholarship @ Claremont. It has been accepted for inclusion in Scripps Senior Theses by an authorized administrator of Scholarship @ Claremont. For more information, please contact scholarship@cuc.claremont.edu.
Convergence Time of the Recombination Markov Chain on Small Planar Graphs

Emma Kolesnik

Sarah Cannon, Advisor

Christina Edholm, Reader

Submitted to Scripps College in Partial Fulfillment of the Degree of Bachelor of Arts

December 4, 2020

Department of Mathematics
Abstract

Over the past decade, there has been an increased focus on how political districts are drawn and partitioned. How districts are drawn, and what population groups are included in different districts, can have a large impact on who is elected. Politicians have used this fact to manipulate election outcomes to give an advantage to a particular group in a process known as gerrymandering. Gerrymandering is not always unconstitutional, but generally goes against the fundamental principle of one person one vote. It is important to be able to detect and quantify, particularly in cases where gerrymandering has been used to suppress minority votes. However, this can be a difficult task. Various mathematical tools have been used to help aid in this process, primarily trying to create a baseline of possible plans through random sampling. The main problem is how little we know about how well these tools work. We don’t know the distribution the random sampling draws from and we also don’t know how large the sample needs to be for it to be representative.

This research studies mixing time, or how long the Markov chain needs to be run for it to generate a reasonable sample. We look specifically at the mixing time of Recombination, a type of Markov Chain used for sampling districting plans. We want to see how long the mixing time is if we start at a particular plan to reach the stationary distribution. We will do this by calculating total variation distance between a distribution. We transition from one plan to another using the transition matrix. To analyze this we will plot the total variation distances over time. We want to see the impact of starting at different plans, the size of graphs, and other factors on convergence to the stationary distribution. The main correlation we have mathematical evidence for is the inverse relationship between the total number of plans possible and mixing time. It also appears that as the number of districts increases by one, the mixing time at least doubles.
Contents

Abstract iii

Acknowledgments ix

1 Introduction 1

2 Prior Research 5
   2.1 Tools without sampling 5
   2.2 Markov Chains as a Tool 6

3 Methods and Definitions 9
   3.1 Markov Chains 9
   3.2 Recombination 9
   3.3 Convergence of a Markov Chain 10
   3.4 Methods 10
   3.5 Data Structures 11
   3.6 Code 11

4 Total Variation Distance in Random Planar Graphs 15
   4.1 State Graphs Represented as Random Planar Graphs 15
   4.2 Finding Random Planar Graphs and Analyzing 16
   4.3 Inverse Relationship Between Number of Plans and Mixing Time 16
   4.4 Testing for Factors of Convergence Time at Different Starting Points 28

5 Discussion and Conclusion 33

Bibliography 35
4.1 These graphs are segments of the graphs created to represent Texas (a) and Georgia (b). The graph from Texas includes 20 nodes and the graph from Georgia includes 24. Both only have one loop. After analyzing many similar segments from these states and others, we felt these were representative of an average segment of a state’s dual graph. From these segments we concluded triangulation would be a reasonable way to create similar graphs ourselves.

4.2 Graph (a) is a 9-node graph extracted from the Texas dual graph. Graph (b) is a random 9-node graph created using Delaunay triangulation. As we can see they are very similar. This is just one example, but after examining many small graphs extracted from states these similarities almost always remained. This provided a reasonable justification for using random planar graphs in our research and analysis.

4.3 We plotted the number of plans for each 9-node graph against the mixing time for each graph. The plot contains 100 data points. We are able to see the inverse relationship between the two variables.

4.4 Results from a standard regression looking at the influence of the number of plans on mixing time for 100 9-node graphs.

4.5 Results from a robust regression looking at the influence of the number of plans on mixing time for 100 9-node graphs.

4.6 This plot shows the inverse relationship between the number of plans and mixing time for 50 12-node graphs.

4.7 Results from a standard regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.

4.8 Results from a robust regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.

4.9 We plotted the number of plans for each 15-node graph against the mixing time for each graph. The plot contains only 10 data points, yet we are still able to see the inverse relationship between the two variables.

4.10 Results from a standard regression looking at the influence of the number of plans on mixing time for 10 15-node graphs. They show us a statistically significant and strong relationship.

4.11 Results from a robust regression looking at the influence of the number of plans on mixing time for 10 15-node graphs. Our t-statistic increased in the robust regression from the standard regression. This is likely because robust regressions are often more ideal when dealing with a small sample size.
4.12 For each number of nodes in graphs with three districts, we plotted the number of plans against the mixing times. For each number of vertices held fixed, there is an inverse relationship. However, overall mixing time increases for more numbers of nodes.

4.13 Sample Statistics including mean, median, min and max of mixing times for three district graphs. The min is the fastest mixing time of all the graphs. We included the statistics for 9-node, 12-node, and 15-node graphs. There were 100 9-node graphs, 50 12-node graphs, and 10 15-node graphs we got our sample statistics from.

4.14 We plotted the number of plans for each 12-node graph against the mixing time for each graph. The plot contains 50 data points.

4.15 Results from a standard regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.

4.16 Results from a robust regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.

4.17 We plotted the number of plans for each 16-node graph against the mixing time for each graph. The plot contains only 4 data points due to the long computational runtime of calculation.

4.18 For 12-node and 16-node graphs with four districts, we plotted the number of plans against the mixing times.

4.19 Sample statistics of mixing times of 12-node graphs into three versus four districts.

4.20 Mixing Time of 12-Node Graphs with districts increasing from three to four.

4.21 Here are our two 12-node graphs selected to showcase lack of correlation of many factors with convergence time. Figure (a) had the fewest number of plans and figure (b) had the most. We can see variation in the total number of edges in each graph.

4.22 Product of the spanning trees in each district for our first (a) and second (b) 12-node graph. In both cases there is no correlation between the factors.

4.23 Degree of a plan in the metagraph for our first (a) and second (b) 12-node graph. Overall, there was a lot of variation among these plots, however these two represent the vast majority.

4.24 Regression testing correlation of many factors. Here “time” refers to convergence time, “prob” to probability in the stationary distribution, “st” to product of spanning trees, “edges” to cut edges, and “metagraph” to degree in the metagraph. Our $R^2$ tells us only seven percent of our model is explained by our variables.

4.25 Results from a regression looking at the correlation between degree in the metagraph and convergence time. Our $R^2$ tells us that only two percent of our model is explained by the degree in the metagraph, which is very low.
I would like to thank my advisor Professor Sarah Cannon for her guidance and mentorship and for introducing me to the mathematics of gerrymandering. She has been a wonderful advisor and it has been a pleasure working with her. I would also like to thank my second reader Professor Christina Edholm for her support and time. Lastly, thank you to Camryn Hollarsmith, Ray Song, Yanting Hua, and Ethan Ong for the work they did starting and continuing this project.
Chapter 1

Introduction

Because of the structure of democracy in the United States, citizens elect politicians to represent their interests in their local, state, and federal government. People are divided into districts within their states and cities to help in this process. Generally, people want this process to be as fair as possible, in order to help elect representatives that actually represent their districts. Yet, with the increasing polarization in the United States, many politicians view it as critical for their party to remain in control.

Gerrymandering occurs when legislative districts are partitioned with the intent to favor one group over another. It is most commonly seen as congressional districts are redrawn to give one party an advantage in future elections. However, gerrymandering also occurs at smaller levels of government including the city and county level. For example, in Chicago, many people agree that the City Council suffers from gerrymandering due to segregation, machine politics, and inefficiency (2). Gerrymandering is an ongoing and complex problem because population demographics in the United States are constantly changing. Every decade when the U.S. Census is taken, congressional seats are reallocated among the states based on shifts in population size. As a result, state officials (often the state’s legislators from the majority party) must re-partition the state into representative districts. Some states have independent or bipartisan redistricting committees in charge of this, but many do not. When the majority party is able to control the redistricting process, state legislatures will sometimes strategically design these districts to give their party an advantage in elections. However, the threshold for classifying a plan as gerrymandered is ambiguous (14). The complexity of population demographics and other factors (i.e. physical geography) make it difficult to accurately assess if a plan is reasonable (15). Further complicating the task, after the death of Thomas Hofeller, thousands of his files were made public, and the large amounts of data that often go into creating a districting plan became clear (10). This helped to further solidify an understanding that computer science and math are necessary in quantifying and detecting gerrymandering.

Partisan and racial gerrymandering both aim to limit the power of minority votes and generate wasted votes. They are often achieved using techniques known as “packing” and “cracking”. In the extreme cases, “cracking” disperses a certain population into many different districts and makes it impossible for that population to win a majority of the seats in any district; “packing” puts all of a certain population into one non-competitive district, so that their votes do not impact the outcomes of other competitive districts. Both methods (often used in conjunction) allow one party to win more seats than what would be reasonably expected based on votes, geography, and population makeup.

In detecting gerrymandering, disproportionate outcomes are an initial red flag. For instance,
in the 2012 North Carolina congressional election, Democratic candidates received over half the votes but only won four of the thirteen congressional seats. The disproportional results continued in 2016. As part of a case taken to a U.S. circuit court in 2018, mathematicians identified that the disproportionate outcomes in 2012 and 2016 were a result of gerrymandering. As a result, a Federal Appeals court ruled the plans were unconstitutional for suppressing minority votes. However, when the Rucho v Common Cause case went to the Supreme Court in 2019, they ruled that federal judges do not have the jurisdiction to say gerrymandering is unconstitutional. They suggested that states needed to create laws and amend their constitutions so that there were clear limits to gerrymandering. Later in 2019, a State Court Panel did rule that the plans were unconstitutionally gerrymandered, and forced them to be redrawn. (1)

Though it was a good indicator in North Carolina, sometimes disproportionality of votes versus seats won does not indicate that a plan is gerrymandered. Due to the nature of voter preferences, demographics, and distribution, it is unlikely for some states to have a proportional and fully representative election outcome. Massachusetts is a clear demonstration of this. Even though Massachusetts' Republicans regularly receive around 30 to 35 percent of the votes statewide in elections, they have not won a seat in the U.S. House of Representatives since 1994 (14). The Metric Geometry and Gerrymandering Group at Tufts University proved that their under-performance is attributed to the physical distribution of votes throughout the state, not gerrymandering. Individuals that vote for the Republican party are uniformly distributed across all towns and precincts and so it is currently impossible for Republicans to win a majority of votes in any congressional district. Even winning a simple majority of votes in a precinct or county is rare. We can see from this example that identifying gerrymandering is not an easy task.

The approach many mathematicians have decided upon involves creating a baseline for what a plan could reasonably be for a given state. While many different techniques exist for creating the baseline, it is achieved through some form of random sampling from all possible plans. Once the baseline is representative of reasonable plans, if a proposed or current redistricting plan is an outlier, then there is evidence of gerrymandering. The more of an outlier a plan is, the better evidence it may be gerrymandered. The problem is that while many techniques are wide-spread and considered reasonable, very little is known about how reliable of a tool they are and there are few guarantees. When there are fewer total plans, it is not difficult to calculate how likely a given plan is to be included in the sample, but as the number of plans gets larger, this is no longer known. Without this information it is difficult to determine whether a particular baseline is actually representative.

Quantifying Gerrymandering is a research group at Duke University that uses math to study districting plans and find unbiased evidence to help prove or disprove gerrymandering in a particular plan. Their work was featured in the North Carolina court cases mentioned above. They use flip walk Markov chains to create a baseline for all possible plans. While flip walks can be a beneficial tool in analyzing gerrymandered plans, they are often slow and their mixing time is unknown. As more techniques are tested, it is clear that different states require different analytical techniques. Recombination is another widespread method that also involves finding a baseline of all possible plans through random sampling and comparing potentially gerrymandered plans to the baseline (5, 2, 4, 19). It is faster and involves fewer parameters than the flip walk Markov chain and so is the method we have chosen to analyze. Because the distribution the sample is drawn from is unknown, there is still much work to be done to generally say that Recombination is a reliable and generalizable technique for detecting and quantifying gerrymandering. This research looks at how long Recombination takes to converge to its stationary distribution. We also look at commonly
used heuristics to test for Markov chain convergence and to see how good they are at predicting convergence for Recombination. It is important to know the convergence time of a Markov chain to ensure that the sampling process runs for long enough to be representative. In this paper we looked at many different aspects of districting plans and their possible effects on convergence time. We found that, at least on a small scale, the more districting plans that could be made, the faster they converged.
Chapter 2

Prior Research

There are many mathematical techniques that have been designed to aid in detecting and quantifying gerrymandering. Most of them have some way of detecting whether a plan is an outlier. Many do this by attempting to create a representative sample of plans. When creating this sample and analyzing the mathematical methods used, there are a numerous factors that are especially important to consider. First, they look at population balance, and ensure that each district is within a certain percentage of an ideal district size. Next, they need to follow state guidelines for contiguity. Almost all states require district contiguity by law, and it is the standard practice even when not formally required. Contiguity is defined as topological connectedness, where a district should not be made up of multiple disconnected components. Contiguity has some subtleties in practice, because of water, corner adjacency, and other geographic difficulties. Most states also have rules regarding compactness, where certain shapes for districts are preferred. However, because of loose definitions by states, compactness is generally not considered as crucial as contiguity when mathematically creating plans. When trying to account for compactness, some techniques will utilize the length of the boundary between districts and similar metrics. This can help ensure possible plans meet a basic threshold for a "visual check". Population balance is also often checked, as districts must be roughly balanced in terms of number of people in each district. Different states and cities have different rules and percentages in regards to how balanced the districts must be. The last two considerations regard preferences for how communities are split up when necessary. Almost all states have a preference to preserve communities defined by cities, counties, municipalities, or other communities of interest. There are no well defined best practices when it comes to splitting rules. One method is counting the number of some defined community that are split and allowing this to impact the likelihood of including a plan in the sample. Lastly, the Voting Rights Act (VRA) has some conditions on how minorities are divided into districts to ensure fair representation. In this section I introduce some of the common techniques that have been used to try to detect and quantify gerrymandering and analyze their different strengths and weaknesses.

2.1 Tools without sampling

Though creating a baseline is currently the most widespread and accepted technique in testing for gerrymandering, it is often not the most simple. Here I review some techniques that are often good to use when first trying to get a sense of gerrymandering in a community or state. They are also much simpler to understand and as such are occasionally favored by non-technical communities.
2.1.1 Proportionality

A first approach when looking for potential gerrymandering is to examine proportionality of votes received versus seats won. However, proportionality can often be extremely misleading when detecting gerrymandering. For example, in Massachusetts, election results statewide do not reflect the proportion of democrats and republicans in the state. Researchers were able to prove that this was not indicative of gerrymandering. They conducted a rigorous proof to show it would be nearly impossible for republicans to win a single seat in Massachusetts due to numerical and geometric uniformity of votes in the state (14). The research in Massachusetts helped to show that an extreme representational election outcome does not always equate to gerrymandering and that the tool best suited to detect gerrymandering can vary greatly by state and region.

2.1.2 Efficiency gap

Mathematics researchers Mira Bernstein and Moon Duchin created a simple formula to test for gerrymandering called the efficiency gap (3). The formula can be computed using data from a single election. If the result surpasses a certain threshold, then the districting plan is found to have a discriminatory partisan effect and is therefore considered gerrymandered. The efficiency gap fundamentally measures the proportionality of seat share to vote share. The efficiency gap, equation 2.1, where $T$ is the vote share and $S$ is the seat share, can arguably identify legally actionable gerrymandering when its magnitude is greater than 8%. The efficiency gap formula counts wasted votes for the winning or losing side. When the efficiency gap is nearly zero, then the plan is fair, meaning that both parties waste about an equal number of votes. The efficiency gap generally captures packing and cracking, though it does not directly penalize plans for it in their score. It also does not penalize peculiarly-shaped districts. The efficiency gap was used to help provide evidence to courts for gerrymandering in Wisconsin. While a reasonable starting point in analyzing gerrymandering, the efficiency gap has many limitations and is often not reliable. It is a simple method which makes it appealing to use in court cases, but it’s simplicity is also a big flaw. Gerrymandering is almost always too complex to be captured and described by a single score. As it does not capture spatial distribution of voters, it also is not able to account for state-specific nuances to gerrymandering. The efficiency gap also often penalizes proportionality and is volatile in competitive political races.

$$EG = 2T - S - \frac{1}{2}$$  \hspace{1cm} (2.1)

2.2 Markov Chains as a Tool

Markov Chains are a widely utilized tool in identifying gerrymandering. Markov chains have many different applications not relating to gerrymandering, and as such are widely studied. Different research groups have different opinions on which Markov chain is best to create a sample and baseline of plans. Some important factors to consider for each Markov chain are how long it takes to create a truly representative sample (mixing time), computational cost, and ease of adding constraints to adapt with state laws regarding districts.
2.2.1 Flip Walk

The flip walk Markov chain has been a widely used tool in quantifying and detecting gerrymandering. The flip walk chain has had many different applications in many different fields. It’s application to gerrymandering was studied and tailored by the group Quantifying Gerrymandering at Duke University and it is the main tool they use. Flip walks build a baseline by randomly changing the district of a single node one at a time. The flip walk Markov chain found evidence of a firewall to protect Republican majorities in North Carolina \(20\). The group also informally conducted research into potential gerrymandering in Maryland using flip walks, but were inconclusive in their results \(19\). While flip walks offer many benefits as a tool, the method is much slower than other Markov chains used to create baselines, such as Recombination. Its mixing time is also unknown. The stationary distribution is known, but the ensemble of plans is too great to exactly know when it has been reached. Another main critique is that a large concentration of the distribution flip walks draw from are plans that are non-compact. This is hard to fix and even when compactness constraints are introduced, it further slows down the sampling process.

2.2.2 Recombination

Researchers with the Metric Geometry and Gerrymandering Group (MGGG) created Recombination as a sampling method to aid in detecting gerrymandering. They have used Recombination to help provide evidence for gerrymandering in Virginia as well as in other states \(12\)(13)(20)(4)(2). Recombination is done by merging two districts together and then splitting them in a new way utilizing spanning trees. As a tool it has many benefits over other Markov chains, in particular flip Walks. Primarily, it involves fewer parameters and is much faster. The main negative of Recombination in comparison to other Markov chains used to detect gerrymandering is that it is computationally more costly. However, even though each step is more costly, it appears to mix more quickly than other Markov chains and so the computational cost is not as problematic. The stationary distribution of Recombination is unknown, however some work has been done analyzing the stationary distribution of small planar graphs. \(18\)

2.2.3 Significance with no mixing time

One major problem with Markov chains such as flip walks and Recombination is that very little is known about their mixing time and thus whether the samples they create are actually representative, though this thesis begins to understand mixing times for small planar graphs. Researchers Chikina, Frieze, and Pegden came up with a test that uses a Markov chain but does not need any information about mixing time to provide statistically rigorous evidence of gerrymandering for a plan. Their test was used to show extreme evidence of gerrymandering in Pennsylvania based on the 2012 districting plan \(9\). Their statistical test is able to show that a given districting plan was not chosen from a stationary distribution. Their method is to take a random walk starting at a given plan for any number of steps. They have proven that if the state they end up at is an \(e\)-outlier at the \(p = \sqrt{2e}\) under the null hypothesis then the state was chosen from a stationary distribution. The only assumptions they make are that the Markov chain is reversible and that significance at \(p \approx \sqrt{e}\) is the best possible \(8\). The main flaw with their method is that it can only show that a plan is not generated from the stationary distribution, \(\pi\), and is not actually able to sample from \(\pi\).
Chapter 3

Methods and Definitions

When using mathematical methods to quantify redistricting plans, states are represented as dual graphs where the nodes represent some “building block” of a district (counties, cities, towns, census blocks, precincts, etc.) and each edge expresses that the two “building blocks” are geographically contiguous. A redistricting plan is a way of partitioning the nodes in a dual graph into some given number of connected sections with equal population. These plans represent potential political redistricting maps in the United States. For the purpose of consistency and simplicity, in this paper, all nodes are considered to have equal population and all districts are exactly equally sized. Due to these constraints, we only look at cases where the number of nodes is divisible by the number of districts.

3.1 Markov Chains

There are many algorithms that can be used to quantify gerrymandering. Markov chains are algorithms that are commonly used to generate random samples. A Markov chain is a mathematical process that moves between positions in a state space according to a transition rule. More specifically, it represents a random walk where the probability of arriving at a particular next state only depends on the present state. The sequence of random states being visited can be denoted by a sequence of random variables $X_i$, where $X_i$ is the state being visited at step $i$. This is otherwise known as a memoryless random walk. When taking a large number of steps, the initial state is nearly unknown. The walking process can converge to a steady state and for any ergodic Markov chain there exists a unique stationary distribution. Once the probability of transitioning from each state to every other state is known, this stationary distribution is the solution to the linear equation $\pi P = \pi$, and when the Markov chain is ergodic there is a unique solution $\pi$.

3.2 Recombination

Recombination is one of the Markov chains used to generate random samples of redistricting plans. The states are the districting plans and each transition probability corresponds to transforming one districting plan into another according to the rules below. Given the random walk procedure, we are able to calculate probabilities of different recombinations for the transition matrix. At each position of the transition matrix, entry $(i,j)$ is equal to the probability of transitioning from state $i$ to state $j$. In the context of gerrymandering, this is equivalent to the probability of transitioning from plan $i$ to plan $j$. The size of the matrix depends on the number of redistricting plans that a dual
graph has.

**Steps for Recombination**

1. From the current redistricting plan, randomly choose 2 districts (each pair of districts is equally likely to be chosen). If the districts don’t touch, we start over and select 2 different districts until they do.
2. Take the union of the nodes and edges in and between the 2 districts.
3. Pick a uniformly random spanning tree of the union.
4. Pick a uniformly random edge of the spanning tree.
   - If removing the edge creates 2 disjoint districts of equal size, it is a balanced cut edge of the spanning tree and we remove it to form the 2 districts. If this new plan is different from the previous plan, recombination was successful.
   - Else, stay at the current redistricting plan.

### 3.3 Convergence of a Markov Chain

*Total variation distance* is the measure of how close a Markov chain distribution is to convergence. The total variation distance between two probability distributions is defined as the maximum difference between the probabilities assigned to a single event by the two distributions:

\[
TVD = \max_{A \in \Omega} |\pi(A) - x_i(A)|
\]

Our set is countable and so we are able to use formula 3.1. In our equation, \(\Omega\) is the set of all plans, \(\pi\) is the stationary distribution, and \(x_i\) is the probabilities at each step \(i\).

\[
TVD = \frac{1}{2} \sum_{\sigma \in \Omega} |\pi(\sigma) - x_i(\sigma)|
\] (3.1)

_Mixing Time_ refers to the time it takes for a Markov chain to be “close” to its stationary distribution. “Closeness” is defined as when the Total Variation Distance at the worst case starting point is less than or equal to \(\frac{1}{4}\).

### 3.4 Methods

We want to analyze dual graphs to learn more about Recombination. It is important to be able to find the stationary distribution for any size graph. Once we know the stationary distribution, we are able to calculate the time to get there from each starting state.

We explicitly calculate the stationary distribution \((\pi)\) of Recombination by making the transition matrix \((P)\) and solving \(\pi \cdot P = \pi\).

To aid in our research and analysis, we needed to create a number of functions to help generate and analyze data using Python. These functions to do a variety of tasks. Most importantly: generate a list of all possible redistricting plans, create a transition matrix \((P)\) from the probabilities of transitioning from one plan to another, calculate the stationary distribution \((\pi)\) from the transition
matrix, and determine how close to convergence a probability distribution is at every step.

3.5 Data Structures

I. Dual Graphs
Dual graphs are represented as NetworkX Graphs. Nodes are labelled with integer values starting at 0. For grid graphs we do not use the .grid_graph() generator, as our code requires nodes represented as a single integer value and not as a coordinate.

II. Redistricting Plans
Redistricting plans are represented as dictionaries where the keys are the nodes of the graph and the values are which district the node is in. Both keys and values are represented as integers. The keys go from 0 to (total number of nodes - 1). The values range from 0 to (total number of districts - 1).

III. Transition Matrices
A transition matrix is represented as a NumPy array.

3.6 Code
All of the code for this research, along with all of the dual graphs we created and mixing times for each graph, can be found on GitHub: https://github.com/emmakolesnik/Senior-Thesis

Our code will find the stationary distribution for any given graph, where the nodes are divided evenly into a given number of districts. In this research our goal is to look at mixing time and convergence, and so the only graphs we input are ones where the Markov Chain is ergodic. In order to find the stationary probability distribution for a dual graph we first need to find the transition matrix. We then are able to calculate the total variation distance for our graph.

The function to find the transition matrix uses a number of helper functions, which we have listed and described below, beginning with the foundational functions.

Note: Throughout our code, we find the number of spanning trees for a particular graph, G, using Kirchoff’s Matrix-Tree Theorem. We find the Laplacian of G, delete the last row and column, and then calculate the determinant.

a. Redistricting Plans
This function returns a list of all districting plans for a particular graph. Each plan is represented by a dictionary. It takes in a graph, the number of districts you want, and then initializes an empty dictionary, list, and sets a counter to 0 (all in the function definition).

We utilize recursion so that this function will work for any size graph and any number of districts, as long as the number of nodes is divisible by the number of districts. The base case is if the number of districts is equal to 1, in which case we check if all the nodes are connected,
and if they are, use a for loop to add to the dictionary where the keys are the nodes and the values are what district they are in. We then append this dictionary to the list of plans.

If the number of districts is greater than 1, we begin by creating a list of all possible district 0s (checking for connectivity). In order to avoid any repeat plans, node 0 is always in district 0. We then loop through all possible district 0s.

For each option, we assign all of the nodes in district 0 value 0 in a dictionary. We then find the first available node not in district 0 and put it in district 1. We create a subgraph that includes all the nodes not in district 0. We then recurse on this inputting the subgraph, one fewer district, the dictionary we’ve started, the list of plans (at the first step this will be empty), and the counter incremented by 1.

Once all the levels of recursion have finished we return the list of dictionaries (this is outside the if/else).

b. **Find districts in common**

In order to calculate the probability of transitioning from one plan to another, it is necessary to be able to know how many districts two plans have in common.

This function takes in 2 redistricting plans and will return the districts the 2 plans share. The plan can have any number of districts.

We begin by creating 2 empty lists (one for plan 1 and one for plan 2). Then, using nested for loops we go through every node in each plan and add it to its respective list as a sorted tuple by district. We end up with each list consisting of a tuple for each district in the plan. We convert these lists to sets, and take the intersection. The function returns the intersection as a list of tuples.

c. **Number of sequences**

As part of the final probability calculation for probability of transition from one plan to another, we need to know how many ways exist to end up at one plan when we start at another.

This function calculates how many different sequences of random choices exist that will lead to recombination from one particular redistricting plan to another. It takes in a graph that has whatever districts were in common removed as well as the dictionary for the plan after recombination.

We begin by defining a graph that is all the nodes from the inputted graph. Using nested for loops, we add in edges from the initial graph, but only between nodes that are in the same district.

We check if adding a particular edge in between the 2 districts will make the graph connected and keep track of the number of edges that will do this. We can call this number k.

Next, for each district, we calculate the number of spanning trees the district has, we can call the answers x and y respectively.

The final number of sequences the function returns is equal to $x \cdot y \cdot k$.

d. **Probability of a transition**

This function takes in a graph and 2 redistricting plans and returns the probability of transitioning from the first plan to the second. The probability will only be non-zero if the plans have exactly 2 different districts or if the plans are the same.
If there are exactly 2 districts that are different (meaning recombination is possible), we find
the shared districts and delete them from a copy of the graph. We then calculate the number
of sequences (m), inputting this new graph and the second plan inputted into the function
(the plan we want to end up at).

We calculate the determinant (det) of the new graph using numpy arrays and linear algebra
packages.

We also calculate the number of edges in a spanning tree for the 2 combined districts (n).

Lastly, we calculate the number of ways to pick 2 districts from the initial number of districts,
or number of districts choose 2 (p).

The probability is equal to: $\frac{m}{\text{det} \cdot n \cdot p}$

If the two redistricting plans are the same, we begin by creating a list of all the plans not
including this specific plan. We then sum the probabilities of transitioning from this plan to
each of these (p). The probability of staying at the same plan is $1 - p$.

e. Finding the transition matrix
This function takes in a graph and number of districts and returns a transition matrix where
each (i,j) entry is the probability of transitioning from the ith redistricting plan to the jth
redistricting plan. We utilize the list of all redistricting plans we found in an earlier function.

We initialize a matrix where the number of rows and number of columns is equal to the
number of redistricting plans for the graph.

Using nested for loops we go through every (i,j) entry and add in the probability of transitioning
from plan i to plan j (found using the above helper function). The function then returns the
matrix of transitions.

f. Finding our distribution at each transition step
This function takes in a graph, the number of districts, and the number of “steps” you want
to take. Each step is defined as the number of times the transition matrix is multiplied. For
example, if we let P be our transition matrix, three steps would be calculated as: $P \cdot P \cdot P$.

Once we have the transition matrix, we are ready to find the stationary distribution. To begin,
we find the transition matrix for the particular graph (P). We want to use linear algebra to solve
$\pi \cdot P = \pi$ where $\pi$ is a non-zero row vector. Equivalently, we want to solve $\pi(P - I) = 0$. We begin
by subtracting the identity matrix, I, from the transition matrix. Then, we convert the matrix to a
numpy array. The transition matrix is currently not full rank. To make the matrix full rank, we
remove a column. $\pi = 0$ is always a solution but we want a non-zero answer. To ensure 0 is not
a solution, we append a column of all 1s so that each row sums to 1. We also add the condition
$\pi \cdot x = 1$ where $x$ is a column vector where the last entry is 1 and all others are 0. The numpy
package will only solve a matrix dotted with a column vector and $x$ is a row vector, so before doing
these calculations we had to transpose $x$. After solving for $\pi$ we transpose it again to get the answer
in the form of a row vector.

If we repeatedly square $P$ for an ergodic graph, each row of $P$ will eventually converge to $\pi$.
To determine how close $P$ is to convergence at each step (x), we take the Total Variation distance
between each row of $P^x$ and $\pi$. We need to individually look at each row, as until $P$ has become
stationary, its rows will differ. We find the Total Variation distance of each row from the stationary
distribution by summing the absolute value of the difference between each corresponding entry and halving our sum. We keep track of how many steps it takes until every TVD at a step is less than $\frac{1}{4}$. The mixing time is the number of steps it takes the slowest starting plan to reach a TVD of $< \frac{1}{4}$. 
Chapter 4

Total Variation Distance in Random Planar Graphs

4.1 State Graphs Represented as Random Planar Graphs

When looking for the best way to ensure our results could be applicable to real world redistricting problems, we first thought it might be a good idea to use actual graphs based on states. Many states have been transformed into graphs already, available at https://github.com/vrdi/vrdi-graphs.git. However, it proved difficult to create large samples of connected subgraphs with ergodic metagraphs small enough to computationally analyze. When looking at sections of the state graphs, such as in Figure 4.1, we realized that planar graphs created using Delauney triangulation would be a good substitution. Unless there is a lake or some other geographical reason, there were rarely loops in state graphs, and so we decided triangulation is a reasonable method for approximating portions of the state graphs. In Figure 4.2, we can see an example of an extracted 9-node graph from Texas, versus a random 9-node graph we created. They share many similar properties.

![Figure 4.1](image)

Figure 4.1  These graphs are segments of the graphs created to represent Texas (a) and Georgia (b). The graph from Texas includes 20 nodes and the graph from Georgia includes 24. Both only have one loop. After analyzing many similar segments from these states and others, we felt these were representative of an average segment of a state’s dual graph. From these segments we concluded triangulation would be a reasonable way to create similar graphs ourselves.
Figure 4.2  Graph (a) is a 9-node graph extracted from the Texas dual graph. Graph (b) is a random 9-node graph created using Delaunay triangulation. As we can see they are very similar. This is just one example, but after examining many small graphs extracted from states these similarities almost always remained. This provided a reasonable justification for using random planar graphs in our research and analysis.

4.2 Finding Random Planar Graphs and Analyzing

In order to create random planar graphs to analyze, we used Delaunay triangulation. We began by deciding the number of nodes we wanted in the graph, then randomly put them in a grid. We then use the Delaunay function from the scipy.spatial package to create a tessellation of our points. We add in an edges for every Delaunay triangle and then create a nx graph from our points and edges.

For each number of nodes, we created 100 or so random planar graphs. Computationally, I was only able to analyze all of the graphs with 9 vertices. In order to study factors on convergence, I kept track of every total variation distance at each step until mixed.

4.3 Inverse Relationship Between Number of Plans and Mixing Time

From studying many different graphs with three and four districts, we found that generally the more plans a graph had, the faster it converged. This was surprising as we expected to potentially see the convergence time increase. Overall, we looked at 100 different graphs with 9 vertices divided into three districts, 50 different graphs with 12 vertices divided into three districts, 50 different graphs with 12 vertices divided into four districts, 10 different graphs with 15 vertices divided into three districts, and five different graphs with 16 vertices divided into four districts.

4.3.1 Mixing Time of Plans with Three Districts

We began by looking at graphs with 9 vertices and plotting the number of plans each graph had against the mixing time for the graph, as seen in Figure 4.3.

We ran a regression to test the significance of this correlation. The exact results are shown in Figure 4.4. Our regression provided further evidence as we has a high $R^2$ value and a p-value of 0, implying strong statistical significance. $R^2$ measures the degree to which the data is explained by the model (the maximum value is one or 100 percent). A p-value of zero suggests that the results are statistically significant. The absolute value of the t-statistic tells us how strong the relationship is. A value of approximately eight is relatively high. We can note that the p-value and high $R^2$ are
not due to only having one independent variable in this model. We also ran a robust regression, which accounts for more outliers, in Figure 4.5 with similar findings.

When analyzing graphs with 12 vertices we found similar results to what we saw with the graphs with 9 vertices. Our plot in Figure 4.6 showed a similar downward trend to Figure 4.3. When we ran our regressions in Figure 4.7 our $R^2$ was a bit lower than for the 9-node regression, but that is somewhat expected given that our sample size was much smaller. We still have a p-value equal to 0 and a high t-statistic which indicate a strong relationship. This was true also in our robust regression in Figure 4.8.

Computationally we could only get 10 data points for 15-node graphs, however from the graphs we did look at the inverse relationship still stood out. In our plot in Figure ?? ?? for figure there was still a downward slope.

When we ran the regression we got a very high $R^2$ and still had a p-value of 0, so though we had few observations the correlation was still statistically significant. We still have a high t-statistic, though not as extreme as with the 9-node graphs.

In Figure 4.12 we plotted all of the data points on one graph, and as shown the relationship appears to get more extreme with each node increase. It seems as though graphs with similar numbers of plans have similar mixing times regardless of the number of nodes. However, there is an overall upward total trend in mixing times as the number of vertices increases.

**Figure 4.3** We plotted the number of plans for each 9-node graph against the mixing time for each graph. The plot contains 100 data points. We are able to see the inverse relationship between the two variables.
Figure 4.4  Results from a standard regression looking at the influence of the number of plans on mixing time for 100 9-node graphs.

```
. regress time numplans

Source | SS      | df | MS | Number of obs = 100
       | F(1, 98) = 164.80
Model  | 11152.9468 | 1  | 11152.9468 | Prob > F = 0.0000
Residual | 6632.21315 | 98 | 67.6756444 | R-squared = 0.6271
Total  | 17785.16  | 99 | 179.648081 | Adj R-squared = 0.6233

| time          | Coef.  | Std. Err. | t     | P>|t|     | [95% Conf. Interval] |
|---------------|--------|-----------|-------|--------|----------------------|
| numplans      | -1.237726 | .0964153  | -12.84| 0.0000| -1.429058 -1.046393  |
| _cons         | 95.29203 | 2.704375  | 35.24 | 0.0000| 89.92528 100.6588   |
```

Figure 4.5  Results from a robust regression looking at the influence of the number of plans on mixing time for 100 9-node graphs.

```
. regress time numplans, robust

Linear regression
Number of obs = 100
F(1, 98) = 112.67
Prob > F = 0.0000
R-squared = 0.6271
Root MSE = 8.2265

| time          | Robust Coef. | Std. Err. | t    | P>|t|     | [95% Conf. Interval] |
|---------------|--------------|-----------|------|--------|----------------------|
| numplans      | -1.237726    | .1166041  | -10.61| 0.0000| -1.469123 -1.006329  |
| _cons         | 95.29203     | 3.730638  | 25.54| 0.0000| 87.8887 102.6954    |
```
**Figure 4.6**  This plot shows the inverse relationship between the number of plans and mixing time for 50 12-node graphs.

**Figure 4.7**  Results from a standard regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.
. regress time numplans, robust

Linear regression
Number of obs = 50
F(1, 48) = 69.48
Prob > F = 0.0000
R-squared = 0.6031
Root MSE = 11.687

|       | Coef. | Std. Err. | t    | P>|t|  | [95% Conf. Interval] |
|-------|-------|-----------|------|------|----------------------|
| time  |       |           |      |      |                      |
| numplans | -.2531606 | .0303721  | -8.34 | 0.000 | -.3142278 to -.1920935 |
| _cons  | 143.1795  | 4.904111  | 29.20 | 0.000 | 133.3191 to 153.0399  |

Figure 4.8 Results from a robust regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.

15 Vertices

Figure 4.9 We plotted the number of plans for each 15-node graph against the mixing time for each graph. The plot contains only 10 data points, yet we are still able to see the inverse relationship between the two variables.
Inverse Relationship Between Number of Plans and Mixing Time

### Table 1

#### Figure 4.10
Results from a standard regression looking at the influence of the number of plans on mixing time for 10 15-node graphs. They show us a statistically significant and strong relationship.

**. regress time numplans**

<table>
<thead>
<tr>
<th>Source</th>
<th>SS</th>
<th>df</th>
<th>MS</th>
<th>Number of obs = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model</td>
<td>2515.63423</td>
<td>1</td>
<td>2515.63423</td>
<td>F(1, 8) = 37.60</td>
</tr>
<tr>
<td>Residual</td>
<td>535.265768</td>
<td>8</td>
<td>66.908221</td>
<td>Prob &gt; F = 0.0003</td>
</tr>
<tr>
<td>Total</td>
<td>3050.9</td>
<td>9</td>
<td>338.988889</td>
<td>R-squared = 0.8246</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Adj R-squared = 0.8026</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Root MSE = 8.1797</td>
</tr>
</tbody>
</table>

| time   | Coef.  | Std. Err. | t     | P>|t| | [95% Conf. Interval] |
|--------|--------|-----------|-------|-----|----------------------|
| numplans | -0.0473577 | 0.0077234 | -6.13 | 0.000 | -.0651678 to -.0295476 |
| _cons  | 188.9858 | 6.578077  | 28.73 | 0.000 | 173.8167 to 204.1549 |

#### Figure 4.11
Results from a robust regression looking at the influence of the number of plans on mixing time for 10 15-node graphs. Our t-statistic increased in the robust regression from the standard regression. This is likely because robust regressions are often more ideal when dealing with a small sample size.

**. regress time numplans, robust**

<table>
<thead>
<tr>
<th>Linear regression</th>
<th>Number of obs = 10</th>
</tr>
</thead>
<tbody>
<tr>
<td>F(1, 8)</td>
<td>57.15</td>
</tr>
<tr>
<td>Prob &gt; F</td>
<td>0.0001</td>
</tr>
<tr>
<td>R-squared</td>
<td>0.8246</td>
</tr>
<tr>
<td>Root MSE</td>
<td>8.1797</td>
</tr>
</tbody>
</table>

| time   | Coef.  | Std. Err. | t     | P>|t| | [95% Conf. Interval] |
|--------|--------|-----------|-------|-----|----------------------|
| numplans | -0.0473577 | 0.0062644 | -7.56 | 0.000 | -.0618033 to -.032912 |
| _cons  | 188.9858 | 6.8027   | 27.78 | 0.000 | 173.2987 to 204.6728 |
For each number of nodes in graphs with three districts, we plotted the number of plans against the mixing times. For each number of vertices held fixed, there is an inverse relationship. However, overall mixing time increases for more numbers of nodes.

4.3.2 Convergence of Three District Graphs with Increasing Numbers of Nodes

We were able to plot the mean, median, minimum, and maximum of the convergence times for 9-node, 12-node, and 15-node graphs divided into three districts. Without more data points from different sized graphs it is difficult to predict whether the increase is linear or exponential. However, it appears linear when looking at means and medians. It was unsurprising that the min-max range is tighter for 15-vertex graphs as we have significantly fewer data points.
4.3.3 Convergence of Plans with Four Districts

Similarly to with three district graphs we found a relationship between number of plans and convergence time with four district graphs. Overall we were able to analyze 50 12-node graphs and 4 16-node graphs.

In Figure 4.14 we can see that there is a downward slope to the curve when we plotted mixing time against the total number of plans for each graph. When we ran a standard regression in Figure 4.15 we see that there is a p-value of 0 and a relatively high absolute value of the t-statistic. We can see that our $R^2$ is much lower than for any of the three district regressions. While the $R^2$ is lower, for the expected amount of variance in our sample it is still high enough for our correlation to be significant.

Due to computational limits we weren’t able to collect a large enough sample of data points for 16-node graphs for results of a regression to be meaningful. In Figure ?? we can see that plotting the mixing time for our 16-node graphs still gave us a downward slope. We can also see in Figure ?? that the trends we saw when plotting different sized graphs with three districts in Figure 4.12 are similar to trends with 12-node and 16-node graphs into four districts.
Figure 4.14 We plotted the number of plans for each 12-node graph against the mixing time for each graph. The plot contains 50 data points.

Figure 4.15 Results from a standard regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.
Inverse Relationship Between Number of Plans and Mixing Time

. regress time numplans, robust

Linear regression

|                  | Coef.  | Std. Err. | t     | P>|t| | [95% Conf. Interval] |
|------------------|--------|-----------|-------|-----|----------------------|
| time             |        |           |       |     |                      |
| numplans         | -1.7782754 | 0.0956672  | -8.14 | 0.000 | (-2.9706272, -0.0859235) |
| _cons            | 327.2665  | 14.99098  | 21.83 | 0.000 | (297.1251, 357.4079)  |

**Figure 4.16** Results from a robust regression looking at the influence of the number of plans on mixing time for 50 12-node graphs.

**Figure 4.17** We plotted the number of plans for each 16-node graph against the mixing time for each graph. The plot contains only 4 data points due to the long computational runtime of calculation.
4.3.4 Convergence Time When District Size Increases

We found that when increasing the number of districts from three to four with 12-node graphs that the convergence time increases substantially, even when controlling for number of plans. We think this may be explained by the number of nodes included in each Recombination step. For a 12-node graph divided into three districts, each Recombination includes 8 nodes, or $\frac{2}{3}$ of the total nodes. For a 12-node graph divided into four districts, each step only includes 6 nodes, which is only $\frac{1}{2}$ of the total nodes in the graph.
**Figure 4.19** Sample statistics of mixing times of 12-node graphs into three versus four districts.

**Figure 4.20** Mixing Time of 12-Node Graphs with districts increasing from three to four.
4.4 Testing for Factors of Convergence Time at Different Starting Points

It would be potentially helpful to know what characteristics make a plan a “good” starting point (or converge quickly) for Recombination as it would allow Recombination to only be run to create a baseline starting at plans that converge quickly. To explore this, we looked at factors that could determine the convergence of a specific starting point. Throughout this section, we use convergence time to mean the number of steps it takes a specific plan to get a total variation distance within $\frac{1}{4}$ of the stationary distribution. Previous research found correlations between the product of spanning trees, cut edges, and shapes with the stationary distribution. (18) However, when testing these factors in addition to degree of a plan in the metagraph we were not able to find any correlation.

Overall we looked at correlations for 50 9-node, 12-node, 15-node, and 16-node graphs. All of the plots showed a lack of correlation for all the factors. In Figure 4.21 we have picked two extreme examples of 12-node graphs that showcase the lack of correlation with different factors.

First, I looked at the product of the number of spanning trees in each district and found nothing conclusive. In Figure 4.22 we can see there is no line of best fit. Next, I looked at the degree of a plan in the metagraph, or the number of other plans you can transition to starting at a particular plan, and again found no relationship. The results can be seen in Figure 4.23.

![Figure 4.21](image)

**Figure 4.21** Here are our two 12-node graphs selected to showcase lack of correlation of many factors with convergence time. Figure (a) had the fewest number of plans and figure (b) had the most. We can see variation in the total number of edges in each graph. Note that the drawing of the two graphs was done with the networkX `.draw_planar()` function which does not uniformly place the nodes.
Testing for Factors of Convergence Time at Different Starting Points

![Graph a](image1.png)  ![Graph b](image2.png)

**Figure 4.22** Product of the spanning trees in each district for our first (a) and second (b) 12-node graph. In both cases there is no correlation between the factors.

![Graph a](image3.png)  ![Graph b](image4.png)

**Figure 4.23** Degree of a plan in the metagraph for our first (a) and second (b) 12-node graph. Overall, there was a lot of variation among these plots, however these two represent the vast majority.

### 4.4.1 Regression

In order to help better understand potential factors correlation with convergence time, I ran multiple regressions of the effect of factors on convergence of different starting plans. Factors I considered for convergence time of specific plans were:

- **Number of cut edges** ($\beta_1$) The number of edges in a plan where if they were deleted, the graph would no longer be connected.

- **Product of Spanning Trees** ($\beta_2$): The product of the number of spanning trees contained in each individual district.

- **Probability in the Stationary Distribution** ($\beta_3$): The probability of a particular plan in the graph’s stationary distribution.

- **Degree in the Metagraph** ($\beta_4$): The number of other plans you can transition to from the current plan.
I ran regressions for these factors on multiple graphs of every size. Figure 4.24 shows an example regression from a 12-node graph. In every case, the $R^2$ was extremely low, meaning almost none of our results are explained by our model. Degree in the metagraph was the only factor that had a low enough p-value to be statistically significant, however, when I did an individual regression, as shown in Figure 4.25, the $R^2$ was still too low for there to be a correlation. In the individual model, the p-value went up and the t-statistic went down further suggesting that degree in the metagraph is not strongly correlated to convergence time of a plan.

![Regression Table]

**Figure 4.24** Regression testing correlation of many factors. Here “time” refers to convergence time, “prob” to probability in the stationary distribution, “st” to product of spanning trees, “edges” to cut edges, and “metagraph” to degree in the metagraph. Our $R^2$ tells us only seven percent of our model is explained by our variables.
Testing for Factors of Convergence Time at Different Starting Points

Figure 4.25 Results from a regression looking at the correlation between degree in the metagraph and convergence time. Our $R^2$ tells us that only two percent of our model is explained by the degree in the metagraph, which is very low.
Chapter 5

Discussion and Conclusion

While overall the mixing time of Recombination is still unknown, this paper has found some patterns that can be useful in determining if mixing has happened. We found that generally graphs with more total plans have faster mixing times when number of nodes is held constant, though in previous research we have found that there are no good predictors for the total number of plans. We also found that as you increase the number of districts, mixing time becomes significantly slower. If you hold the number of districts constant and increase the number of nodes, convergence time appears to increase more linearly. Upon testing many other factors, particularly those that could determine differences in mixing time depending on the starting plan for a graph, we were not able to find anything conclusive.

The scope of this work is somewhat limited by computational capacity and computer memory. We didn’t compute any statistics for graphs larger than 16 vertices. For 9- and 12-node graphs, we were able to generate enough data to produce statistically significant results for some factors, yet for larger graphs at least many hundreds more graphs would need to be analyzed.

Another consideration is to explore other methods and variations of Recombination. The type of Recombination this paper analyzes requires the starting and ending plans to have exactly two districts not in common. In previous literature this has not always been the case. The conditions of Recombination can also be modified. Some example conditions that can be set include controlling for the number of edges between the chosen two districts in the starting plan, limiting the unions of two districts to a certain number or range of spanning trees, etc. Subtly changing the steps of Recombination could possibly change the stationary distribution and thus the ensemble of plans sampled from. It would also be of interest to do similar analysis on the convergence and mixing times of other Markov chains, particularly Flip walks.

Our results are not extensive enough to conclude that the Recombination Markov chain mixes quickly, however they help build the knowledge base surrounding the mathematics of gerrymandering and hopefully can assist in future research. If our results hold for larger graphs, this would be provide evidence that mixing is happening when creating a baseline for states, as many states have few districts with many nodes. If the mixing time increase is linear when the number of districts is held constant, then even with population growth within districts, Recombination could computationally be run for sufficiently long times for mixing to occur. We hope that this will continue to be studied and that this research will contribute to the detection and quantifying of gerrymandering in the United States.
Bibliography


[18] Emma Kolesnik, Yanting Hua, and Ethan Ong. The Stationary Distribution of Recombination as it Relates to Gerrymandering. Summer Research at Claremont McKenna College, 2020.
