The Stationary Distribution of Recombination as it Relates to Gerrymandering

Emma Kolesnik, Scripps College
Yanting Hua, Pitzer College
and Ethan Ong, Pomona College

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 unfairly benefit a particular group in a process known as gerrymandering. While gerrymandering is generally an unconstitutional practice (going against the fundamental principle of one person one vote), it can be incredibly difficult to detect and quantify. Various mathematical tools have been used to help aid in this process, primarily trying to create a baseline of possible plans through random sampling. Our research looks specifically at how Recombination, a type of Markov Chain, is used to find this baseline. The size and attributes of this particular baseline are currently unknown. In order to better understand the probability distribution the Recombination Markov Chain samples from when creating plans for states and much larger graphs we analyzed the probability distributions of small graphs.

Contents

1 Introduction

Gerrymandering occurs when legislative districts are partitioned to unfairly favor one group over another. It is frequently seen when congressional districts are redrawn to ensure one party receives an advantage in future elections. However, gerrymandering also occurs at smaller levels of government including the city and county level. For example, in Chicago, in the public eye many people agree that the City Council suffers from gerrymandering due to segregation, machine politics, and inefficiency [1]. Gerrymandering is an ongoing and complex problem because population demographics are constantly changing. Every decade when the U.S. Census is taken, congressional seats are reallocated among the states based on shifts in population. As a result, it is then up to the state officials (often the state's legislatures from the majority party) to repartition the state into representative districts. State legislatures can and sometimes do strategically design these districts to give their party an advantage in elections. However, the threshold for classifying a plan as gerrymandered is ambiguous [7]. The complexity of population demographics and other factors (i.e. physical geography) make it difficult to accurately assess the fairness of a plan [8]. Many mathematicians are working to help solve this problem and find ways to utilize mathematics in detecting and quantifying 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 they are entitled to based on votes and population makeup.

In detecting gerrymandering, disproportional 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. circut court, in 2018 [9] researchers 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.

Yet, 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 [7]. 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.

To prevent the practice of gerrymandering, we must know how to detect a gerrymandered plan. Currently, mathematicians use a number of different techniques to detect and quantify gerrymandering [5]. One of the most widespread methods involves finding a baseline of all possible plans by taking random samples of plans and comparing a proposed or current plan to this baseline [3, 2, 5, 9]. If the plan is an outlier, then it can be argued that the plan is gerrymandered. Indicating a plan is an outlier among all possible plans is often a method researchers have used to classify a redistricting plan as gerrymandered [6]. While this method of analysis can correctly identify disproportionate plans as outliers as demonstrated in North Carolina's 2012 and 2016 redistricting plans [9], there are cases where disproportionate plans flagged as outliers are not true outliers as demonstrated in Massachusetts [7]. Moreover, when generating random samples using a Recombination Markov Chain, the distribution the random samples are drawn from is unknown. Our research focuses on determining what the baseline is and what properties it has in order to help evaluate the effectiveness of this method.

1.1 Approach and Results

To generate the distribution of redistricting plans, we focus on Recombination, a type of Markov Chain and one method used to randomly sample redistricting plans. We explicitly calculate the stationary distribution of recombination for various small graphs in order to conduct our analysis of the shape and make-up of various redistricting plans and how that affects their probability in the stationary distribution. Through studying number of edges, the product of spanning trees, and other factors, we were able to differentiate probabilities based on defined indicators which will be mentioned below. Furthermore, we have differentiated the distributions of probability groups, and identified factors that have a more influential impact on probabilities.

Key Findings Include:

- The stationary distribution can be partitioned into groups based on number of edges within districts.
- Probabilities within probability groups can be differentiated based on product of spanning trees.
- Number of edges within districts performs better at partitioning probability groups, while product of spanning trees performs better at differentiating individual probabilities within a group or overall.
- The sum of probabilities within each probability group is approximately normally distributed.
- The sum of probabilities within each probability group for grid graphs are outliers and not normally distributed.

2 Background

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 equal and connected sections. These plans represent potential political redistricting maps in the United States. For the purpose of consistency, in this paper, all nodes are considered to have equal population and all districts are equally sized.

2.1 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 in this context. 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, that is, it is memoryless. 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. 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 transition matrix 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 π. The Markov chain we study, Recombination, is not always ergodic but we focus on cases in which it is.

2.2 Recombination

Recombination is one of the Markov Chains used to generate random samples of redistricting plans [5]. 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).
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 the edge is a balanced cut edge of the spanning tree, remove the edge to form 2 districts. If this new plan is different from the previous plan, recombination was successful.
   - Else, stay at the current redistricting plan.

2.3 Metagraphs

The *metagraph* is a graph where each node represents a redisticting plan of the dual graph and each edge represents that recombination from one plan to another is feasible. A dual graph is considered ergodic when its metagraph is connected and not bipartite. In the context of redistricting plans, no metagraph will be bipartite because recombination is always aperiodic. Understanding the metagraph can be useful in visualizing feasible plans for a particular graph and the the differentiation of plans.

3 Methodology

We explicitly calculate the stationary distribution (π) of Recombination by making the transition matrix (P), solving $\pi \cdot P = \pi$, and analyzing π for patterns.

To aid in our research and analysis, we needed to create a number of functions to help generate and analyze data using Python scripts and notebooks. 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, and calculate the stationary distribution (π) from the transition matrix.

All of our code can be found on GitHub:
`https://github.com/CMC-Summer-Research-2020/stationary-distribution-recom.git`

3.1 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 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.

3.2 Code

Our code will find the stationary distribution for any given graph, where the nodes are divided evenly into a given number of districts. There are two techniques to find this distribution. Which one of the two we utilize depends on whether the dual graph in question is ergodic. In order to find the stationary distribution for a dual graph using either method we first need to find the transition matrix.

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 will ultimately return a list of dictionaries, where each dictionary is a particular redistricting plan. 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**
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**
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*y*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: $\frac{m}{det*n*p}$

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. **Determining if the dual graph is ergodic**

Note: As discussed in Section 2, because Recombination is aperiodic the metagraph is connected if and only if the dual graph is ergodic.

This function begins by initializing a graph that will be the metagraph. We add a node for every redistricting plan. We then use nested for loops to see if it is possible through recombination to transition from one plan to another (meaning the probability of transition is non-zero and the 2 plans are not the same). If it is possible we add an edge to the metagraph between the 2 nodes representing these plans. Lastly we check to see if the metagraph is connected using the networkx function 'networkx.is_connected()'.

Now that we have the transition matrix, we are ready to find the stationary distribution. If the dual graph is ergodic we use Technique 1. If not, we use Technique 2.

Technique 1: 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 π 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, and the rows do not sum to 1. To make the matrix full rank, we remove a column. We then append a column of all 1s so that each row sums to 1. $\pi = 0$ is always a solution but we want a non-zero answer. Thus, we want to 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 π we transpose it again to get the answer in the form of a row vector.

Technique 2: We could always use this technique, however if we know a graph is ergodic it is more efficient to use Technique 1. For this method, we find the transition matrix and repeatedly square it until the entries become stationary. If the graph is ergodic, every row in this repeatedly squared matrix will be the same. We arbitrarily can return the first row. This row is the stationary distribution for the given graph. If the graph is not ergodic, each row will represent the stationary distribution from different starting points. Because of this it is necessary to look at the entire matrix.

4 Results

From investigating patterns of graphs, we are able to seperate redistricting plans of a dual graph into probability groups based on useful indicators, including the number of same shapes, the total number of edges within all districts and the presence and frequency of the longest path. The sums of dual graph probability groups are usually normally distributed, except for grid graphs which were found to be an outlier. Furthermore, there appears to be a numerical relationship between each pair of consecutive probability group sums by a factor of 2-3. Additionally, in larger dual graphs, subgroups within probability groups are differentiated by the product of spanning trees for individual districts in a plan. From the probability groups and subgroups, we have also examined small discrepancies within groups, identified potential factors that could predict these differences, and performed regression analysis to detect which factor(s) have the most influential impact on plan probabilities.

4.1 Exploring the Stationary Distribution

Based on prior work [10], we know the presence of shapes, like squares, in a plan increase its probability. In the process of expanding this finding, we found that the presence of shapes like triangles, pentagons, or other larger shapes in districts increases the probability of a particular plan. We know that the stationary distribution is related to the number of spanning trees because taking the union of where at least one district has a shape results in more spanning trees than taking the union of two districts without any shapes. Thus, plans with shapes have higher probabilities in the stationary distribution. In fact, the probability increases proportionally in different ratios depending on the number of shapes. For example, for plans with triangles, the ratio is approximately between 2.25 and 2.5. Below is one example to demonstrate the proportional relationship in our disconnected, ergodic graph (Figure 1). Plan 0 and Plan 2 both have one triangle and as a result have the same probability. Plan 1 has two triangles, therefore, the probability is almost 2.25 times larger than Plan 0 and Plan 2's. Therefore, from the proportional relationship, we could group plan probabilities based on the number of same shapes.

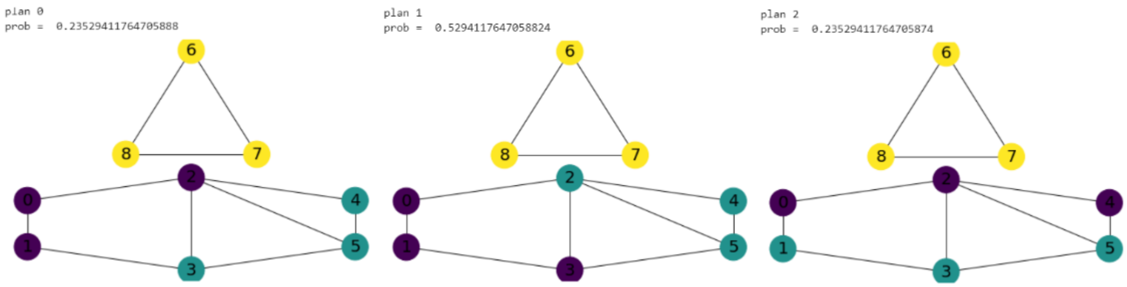

plan 0
prob = 0.23529411764705888
6
8
7
0
1
2
3
4
5
plan 1
prob = 0.5294117647058824
6
8
7
0
1
2
3
4
5
plan 2
prob = 0.23529411764705874
6
8
7
0
1
2
3
4
5

Figure 1: This is an example of a graph with a disconnected dual graph yet is ergodic (has a connected metagraph).

Instead of only investigating individual shapes within districts, we have also analyzed probabilities of graphs with combinations of these shapes as well. Though the probabilities are quite different under the same graph, the results are consistent with our previous findings: more shapes equates to more spanning trees and thereby higher probability. For instance, Figure 2 is a 3x4 grid with two diagonal edges. The probability of transition from the plan with no triangle (plan 0) compared to the plan with one square or two triangles (plan 13) is increased by 6 times; for the plan with no shapes (plan 0) and the plan with two squares and two triangles (plan 10), the probability grows substantially, nearly by 16 times.

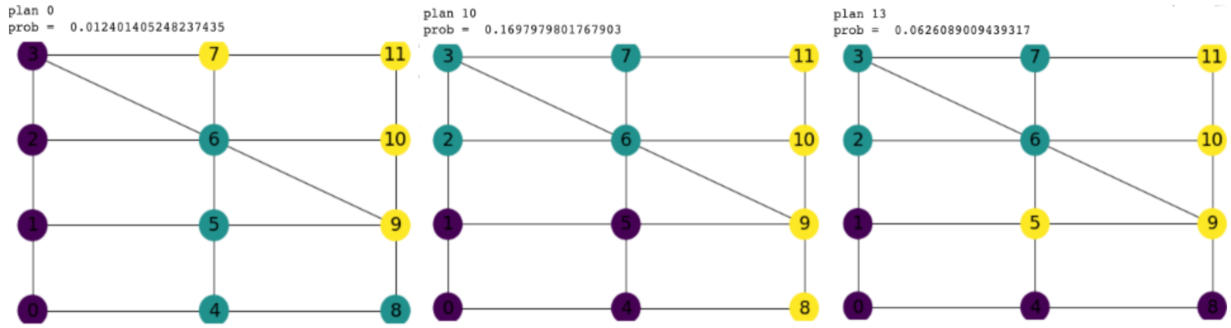

plan 0
prob = 0.012401405248237435
plan 10
prob = 0.1697979801767903
plan 13
prob = 0.0626089009439317
3
7
11
2
6
10
1
5
9
0
4
8

Figure 2: The probability increases 6-fold from plan 0 to plan 13 and increases 16-fold from plan 0 to plan 10.

However, when we examined larger graphs, we found out even if some plans have the same number or no number of shapes, there is a discrepancy in their probabilities. At first, we assumed that the disparity may be due to rounding issues. Yet, after we calculated the probabilities with a greater degree of accuracy, we excluded the precision assumption.

4.2 Probability Groupings

4.2.1 Number of Edges

The proportionality between probabilities and the number of shapes gives us a useful insight on grouping probabilities. However, the grouping is not applicable to plans with no shapes or the combination of different shapes. Shapes can quickly become too complex to quantify and measure for graphs larger than 12 nodes. As we continued to analyze small probability discrepancies in the same probability group, the total number of edges within all districts became a more apparent generalization for probability groupings. Below we include plots that express this relationship for G18, one of the random 12-vertex graphs we generated using a Delaunay triangulation (Figure 3). In Figure 4, the number of edges within districts is plotted against probability for G18. The presence of groupings is quite obvious at first glance and distribution of the data seems to resemble a square root or logarithmic function. In Figure 4, the probabilities within each group are not as easily differentiable, thus, Figure 5 was created by partitioning the graph by probability group and rescaling the x-axis such that the probabilities within each group can be visually differentiable. Among plans within the same group, the total number of edges within districts is the same. With every consecutive probability group, the total number of edges increases.

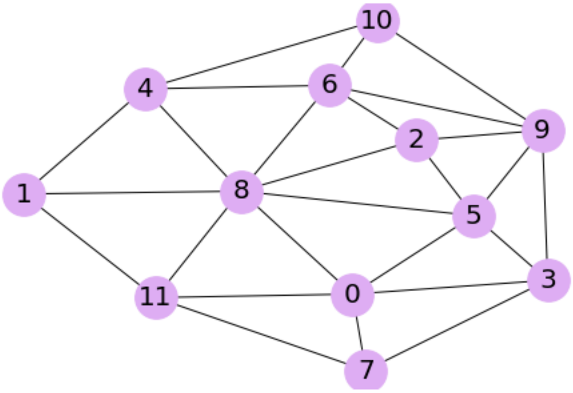


Figure 3: This graph (G18) pertains to Figure 4, Figure 5, and Figure 6.

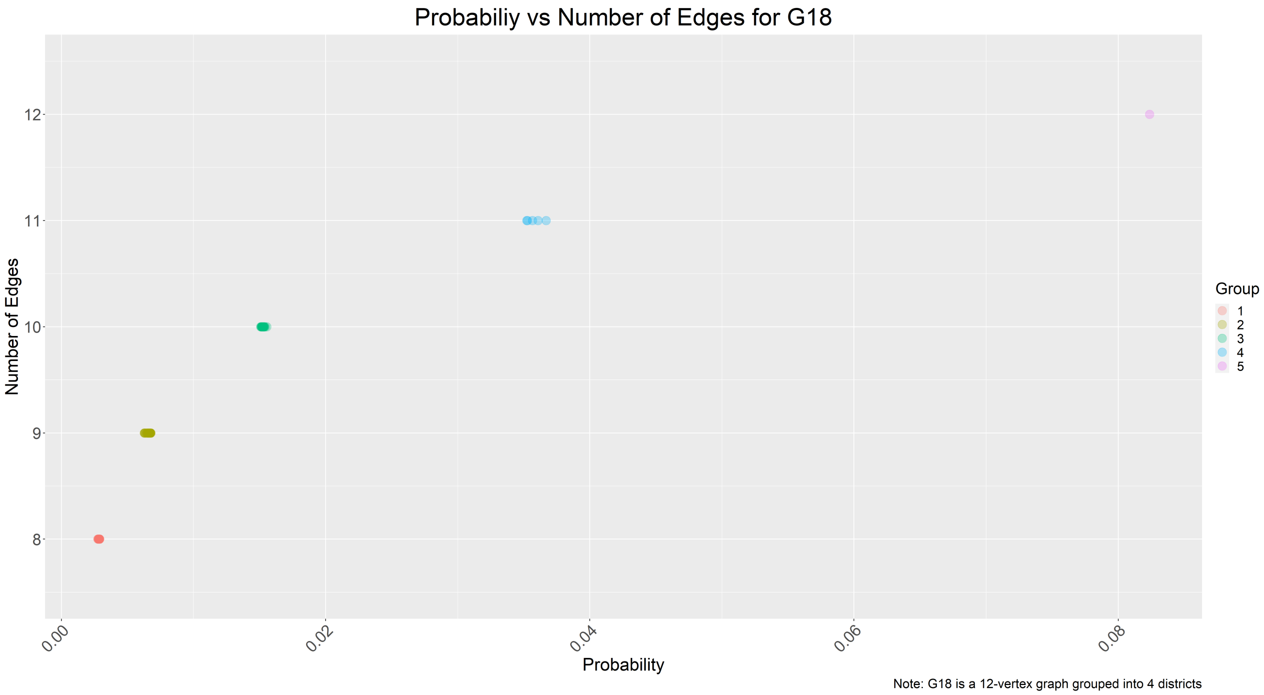


Figure 4: This graph plots all 96 probabilities in G18's stationary distribution (Figure 3). Probability groups are differentiated by color.

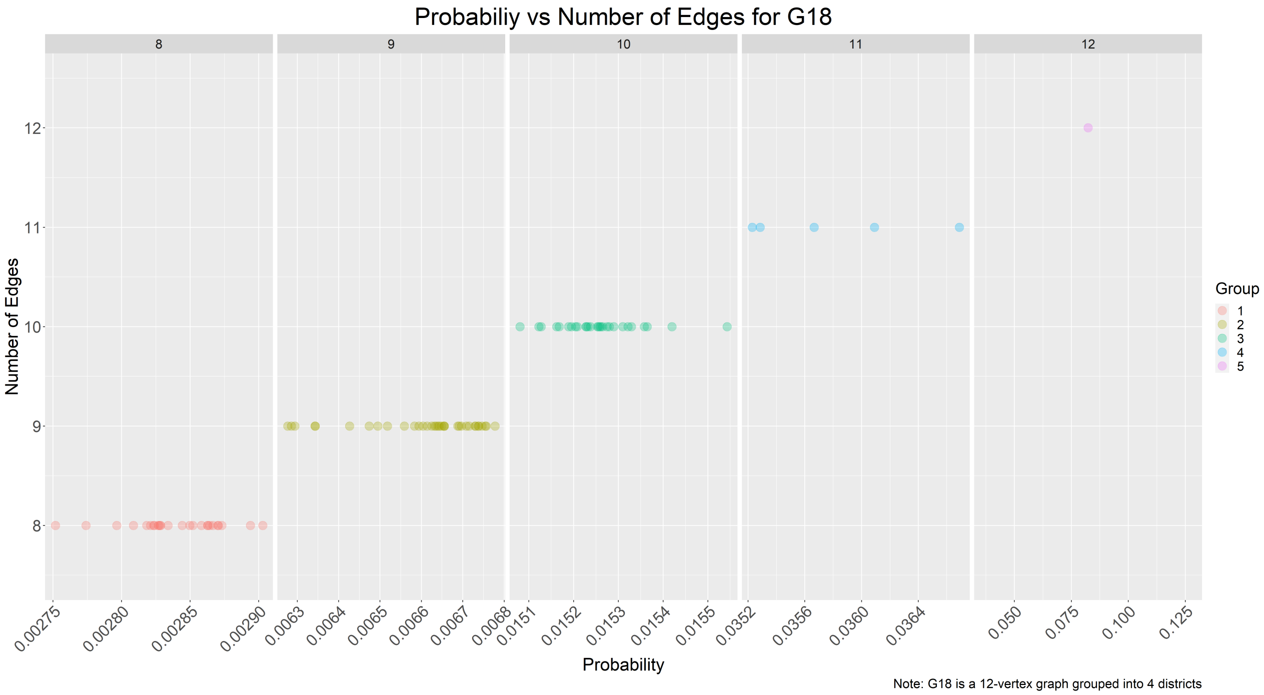
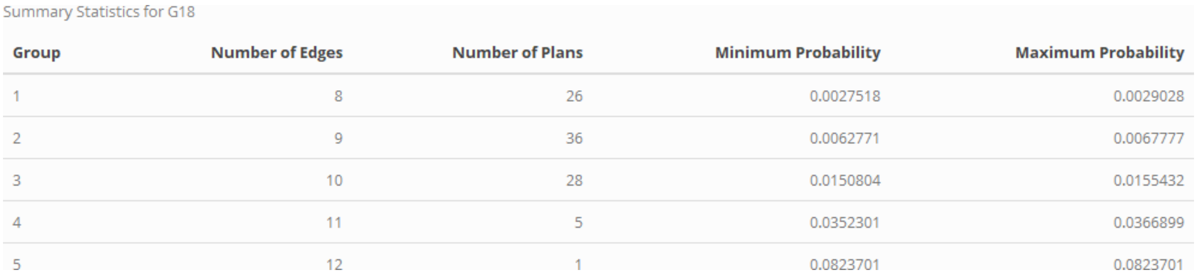


Figure 5: This graph partitions the graph in Figure 4 by probability group such that the probabilities within each grouping are more visibly differentiable. *Note: The x-axis scales between the different groups are different to aid visual clarity.*



Summary Statistics for G18

Group	Number of Edges	Number of Plans	Minimum Probability	Maximum Probability
1	8	26	0.0027518	0.0029028
2	9	36	0.0062771	0.0067777
3	10	28	0.0150804	0.0155432
4	11	5	0.0352301	0.0366899
5	12	1	0.0823701	0.0823701

Figure 6: This table displays the following summary statistics for each probability group in G18 (Figure 3): (1) number of edges within districts, (2) number of plans within probability group, and (3) minimum/maximum probability within probability group.

4.2.2 Numerical relationship between probability groups

We calculated the ratio between the average probabilities in groups as we increased the number of edges contained in groups by 1. As can be seen in Figure 7, this ratio, or factor, between probability groups for all size graphs that we have tested is around 2-2.5. For larger graphs, it appears the multiplier gets smaller as the probabilities get larger. The smaller graphs tended to have more uniform factors. While the smaller graphs led us to think that the relationship might be logarithmic, it appears to be a polynomial relationship with power less than one as graphs get larger We have not yet found this exact relationship.

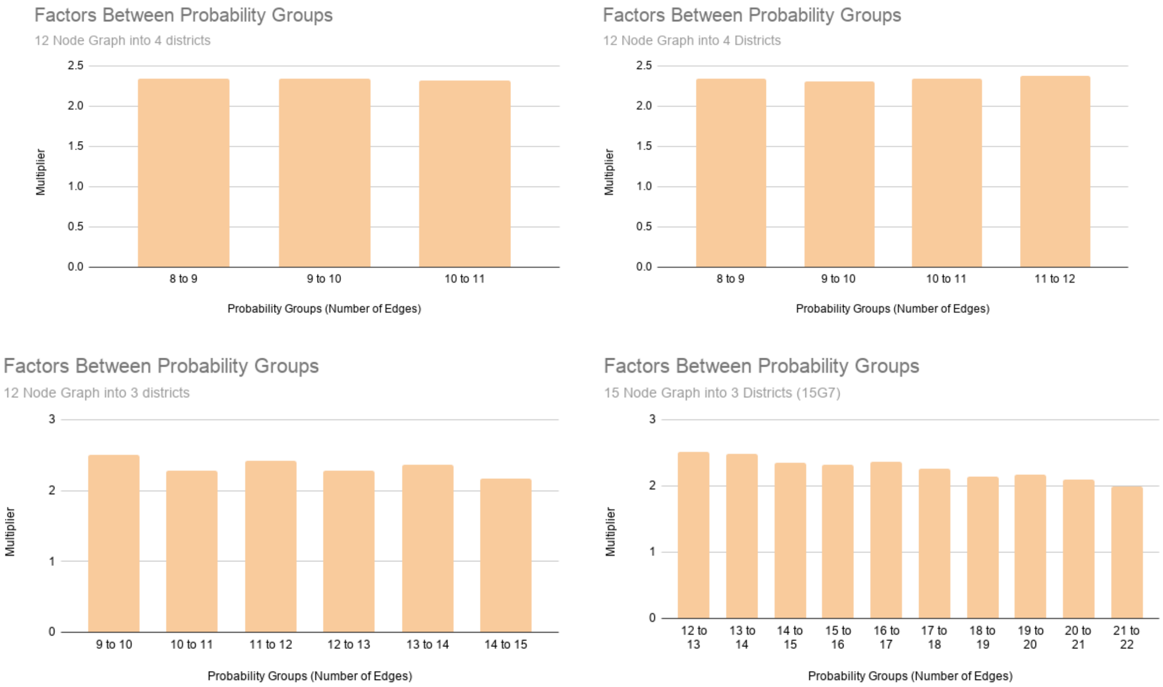
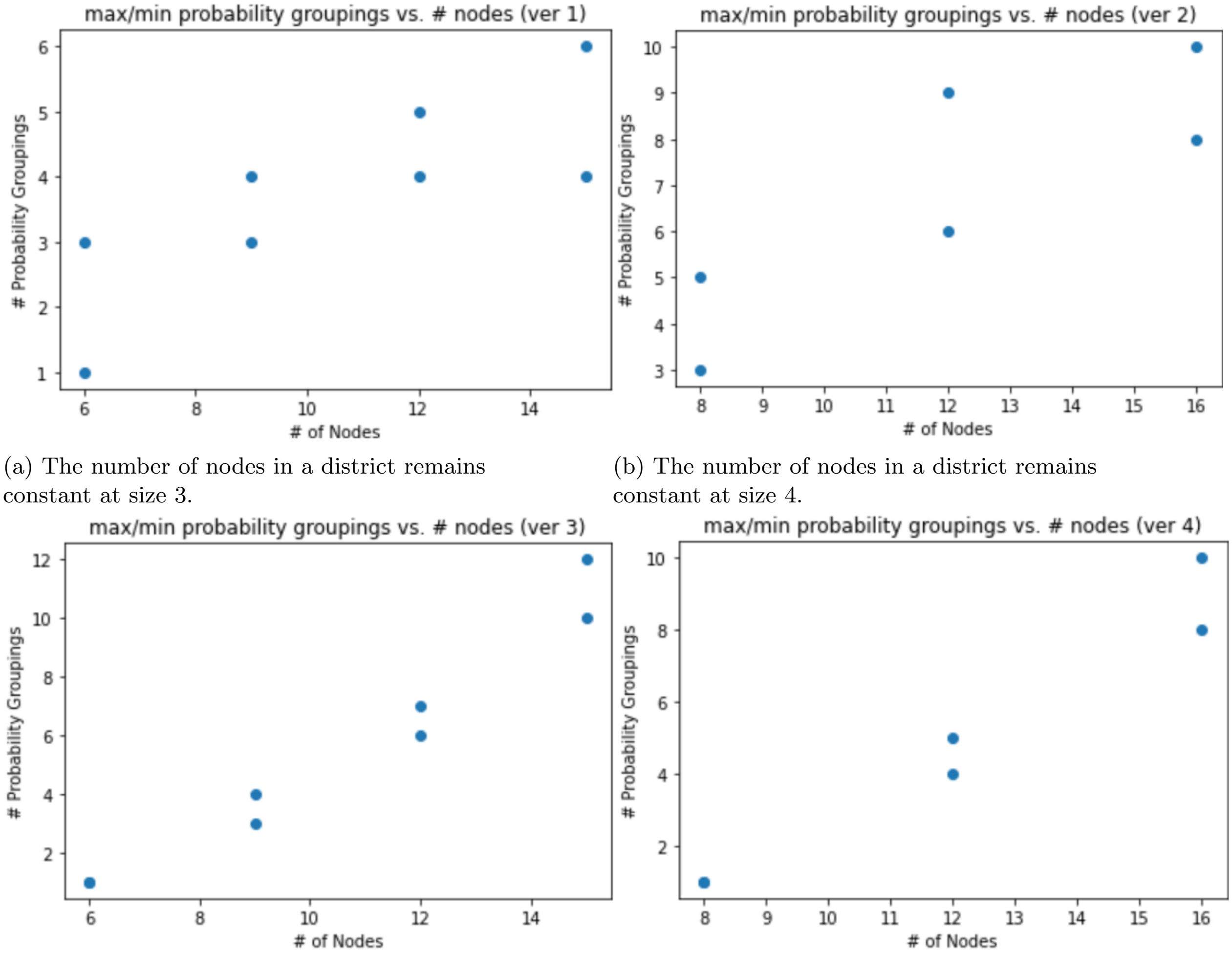


Figure 7: The average probability in a group increases by around a multiplier of 2-3 between each pair of probability groups.

4.2.3 Number of Probability Groups Distribution

Predictors for the number of probability groupings was a point of interest we researched in-depth. We were not able to find any factors that could predict the specific number of probability groups for a given graph, only that in general the number of groups was approximately normal among all possibilities. We tested factors such as total number of plans, total number of edges, and total number of spanning trees. We were able to find a correlation between the potential number of probability groupings and the size and number of districts of a graph as shown in Figure 8. We randomly sampled 200 12-vertex graphs and plotted the number of probability groupings on a histogram for 12-vertex graphs with 4 districts in each plan (Figure 9) and 12-vertex graphs with 3 districts in each plan (Figure 10). In this exploration, we discovered the most common number of probability groupings is dependent on the size of the graph as well as how many districts the graph is to be separated into for each redistricting plan. Furthermore, running the same random sampling simulation on larger graphs resulted in distributions more closely resembling Figure 10 than Figure 9.



(a) The number of nodes in a district remains constant at size 3.

(b) The number of nodes in a district remains constant at size 4.

(c) Each graph is divided into exactly 3 districts.

(d) Each graph is divided into exactly 4 districts.

Figure 8: Plots showing correlation between graph size and number of districts and the minimum and maximum number of probability groupings.

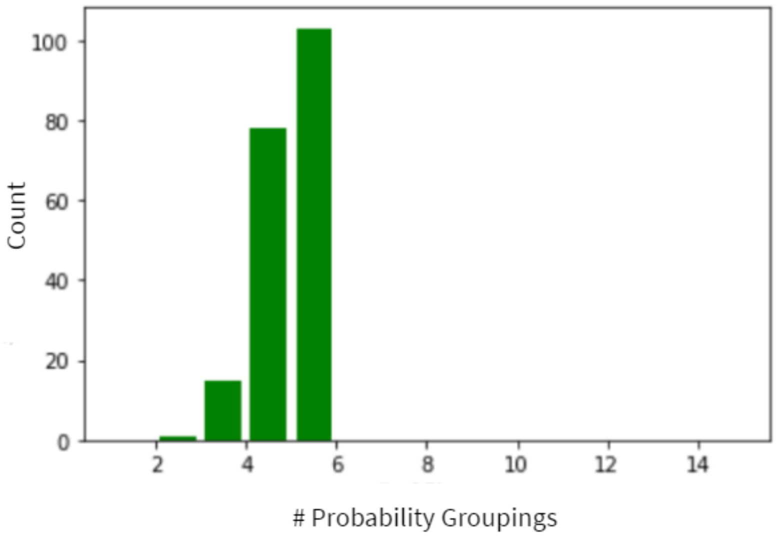


Figure 9: Histogram counting number of graphs for number of probability groupings for 200 12-vertex graphs with 4 districts in each plan.

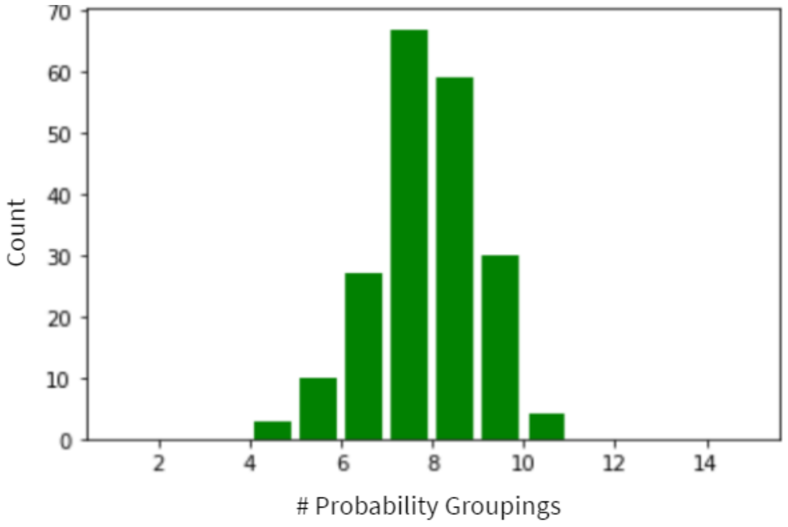


Figure 10: Histogram counting number of graphs for number of probability groupings for 200 12-vertex graphs with 3 districts in each plan.

4.3 Subgroupings of Probability Groups

However, there are still discrepancies in probabilities within the same group. Figure 11 is an example of probabilities from the same group that vary slightly yet the number of edges is not a sufficient explanatory variable. Both contain the same number of edges within all districts, however, there is no clear and generalizable way to reason why plan 73 has a higher probability than plan 74 in the stationary distribution of this graph.

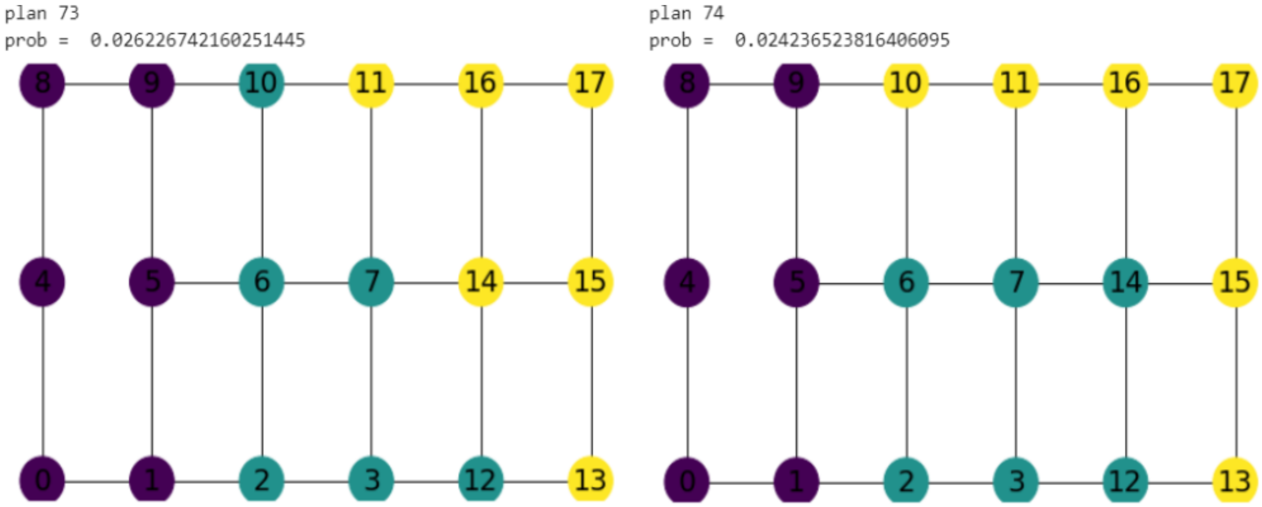


Figure 11: Plans 73 and 74 are in the same probability group yet have slightly different probabilities.

Furthermore, being able to differentiate redistricting plans of a dual graph into probability groupings based on the number of edges within districts is very useful, however, the number of plans exponentially increases as the dual graph gets larger increasing the number of plans in each probability grouping as well. Further investigation was conducted to explore a different generalization for probabilities within the same probability group.

This section elaborates on subgroupings within probability groups, discovered in several dual graphs of size 15. The subgroupings were indicated when plotting the probabilities of several 15-vertex graphs versus the product of spanning trees (product of the number of spanning trees in each district for each plan). Figure 12 depicts 12 such 15-vertex graphs indicating that increasing the product of spanning trees positively correlates with the probability.

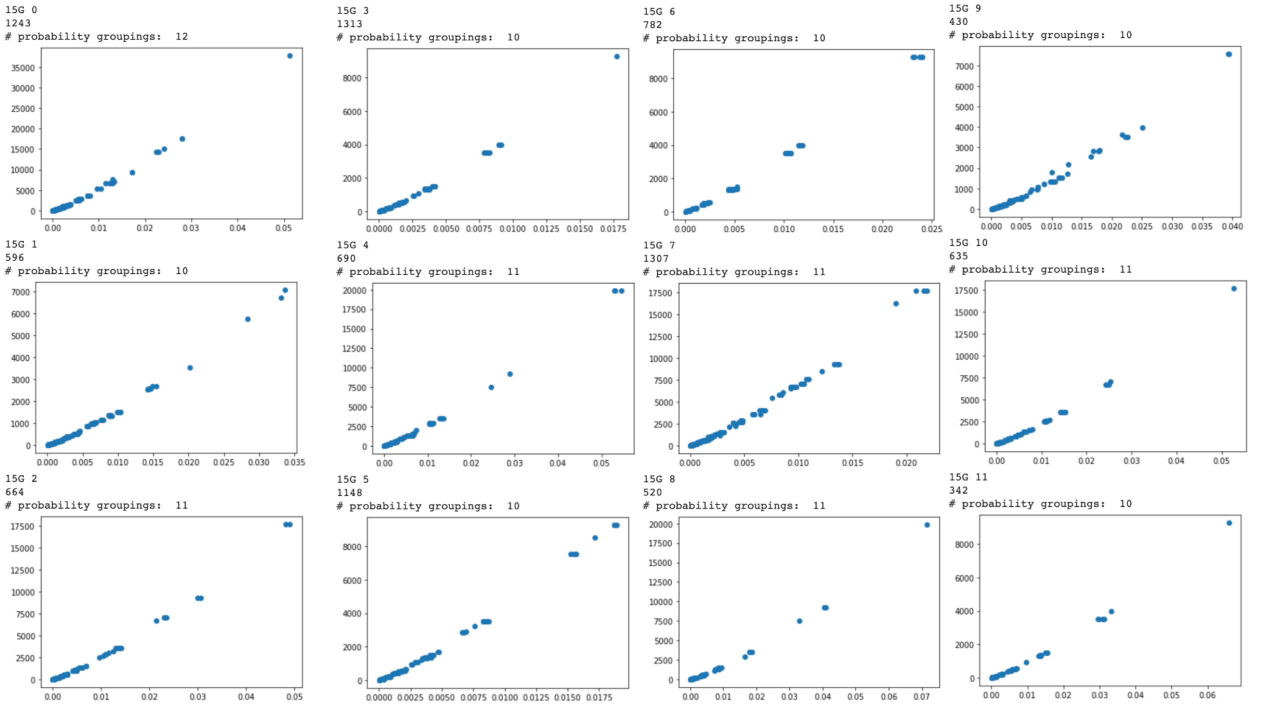


Figure 12: Depicted are the stationary distributions of 12 15-vertex graphs plotted against the product of spanning trees for each plan. At first glance, it appears that the product of spanning trees may be a good predictor for the probability of a redistricting plan and that the relationship is linear.

More specifically, when looking at the probabilities within a particular probability grouping, there are probability subgroups based on the product of spanning trees. This section will look at one particular 15-vertex graph and probability grouping subgroups based on the product of spanning trees. In Figure 15, four of the eleven probability groups from 15G7 (Figure 13) have been plotted against product of spanning trees, indicating varying levels of probability differentiation based on the product of spanning trees. From the analysis thus far, we have observed that not all probabilities within probability groupings can differ in the product of spanning trees, however, for the probability groups that can be differentiated in this way, the product of spanning trees appears to be an accurate distinction between higher and lower probabilities within probability groupings.

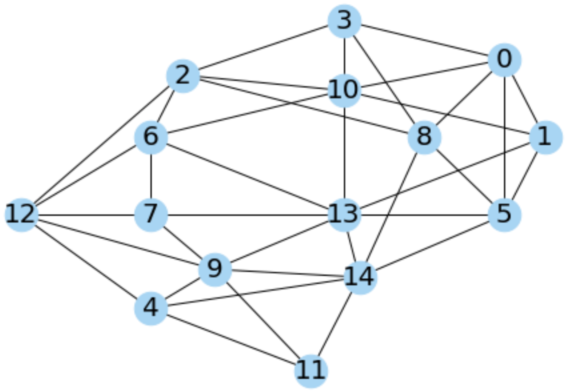
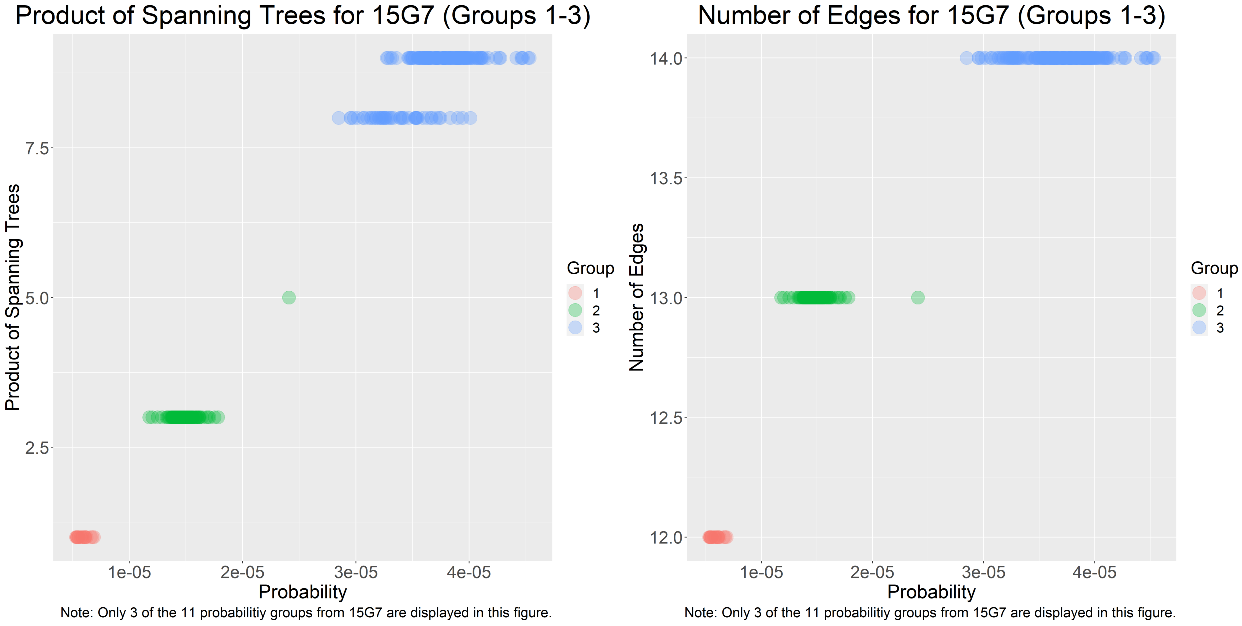


Figure 13: This graph (15G7) pertains to Figure 14 and Figure 15. *Note: This graph is planar but is drawn in this way to be more visually digestible.*

Furthermore, product of spanning trees appears to be a more accurate predictor of probability than number of edges. Figure 15 compares four of the eleven probability groupings in 15G7 (Figure 13) between product of spanning trees and number of edges. In Figure 15a, plotting product of spanning tree indicates varying levels of probability differentiation. Group 1 indicates no differentiation, Group 4 indicates slight differentiation with some overlap, and Groups 8 & 10 indicate a positive correlation for differentiating probabilities within the same probability grouping, though there is not a monotone relationship between probability and product of spanning trees. While the product of spanning trees does not maximally differentiate probabilities in each group, it differentiates probabilities in each group much more so than number of edges. In Figure 14 and Figure 15b, the number of edges is clearly a better predictor of probability grouping, however, is not a better predictor for probabilities within each probability grouping compared to product of spanning trees. While within each probability grouping, the number of edges of a redistricting plan is equivalent to every other redistricting plan in the same probability group, there are distinctive product of spanning trees between redistricting plans within each probability group.

Moreover, we briefly explored this conjecture for product of spanning trees with graphs of size greater than 15 and found similar variance in the ability for the product of spanning trees to differentiate higher from lower probabilities within a probability group. This makes sense as larger graphs have more distinct plans and consequently more distinct probabilities. As a result, the product of spanning trees is expected to have more variance among plans in the same probability groups in larger graphs. The rest of section 4 focuses on other outlets of exploration for determining the underlying factors for the minute differences in probabilities within each probability group.


Product of Spanning Trees for 15G7 (Groups 1-3)

Note: Only 3 of the 11 probabilitiy groups from 15G7 are displayed in this figure.

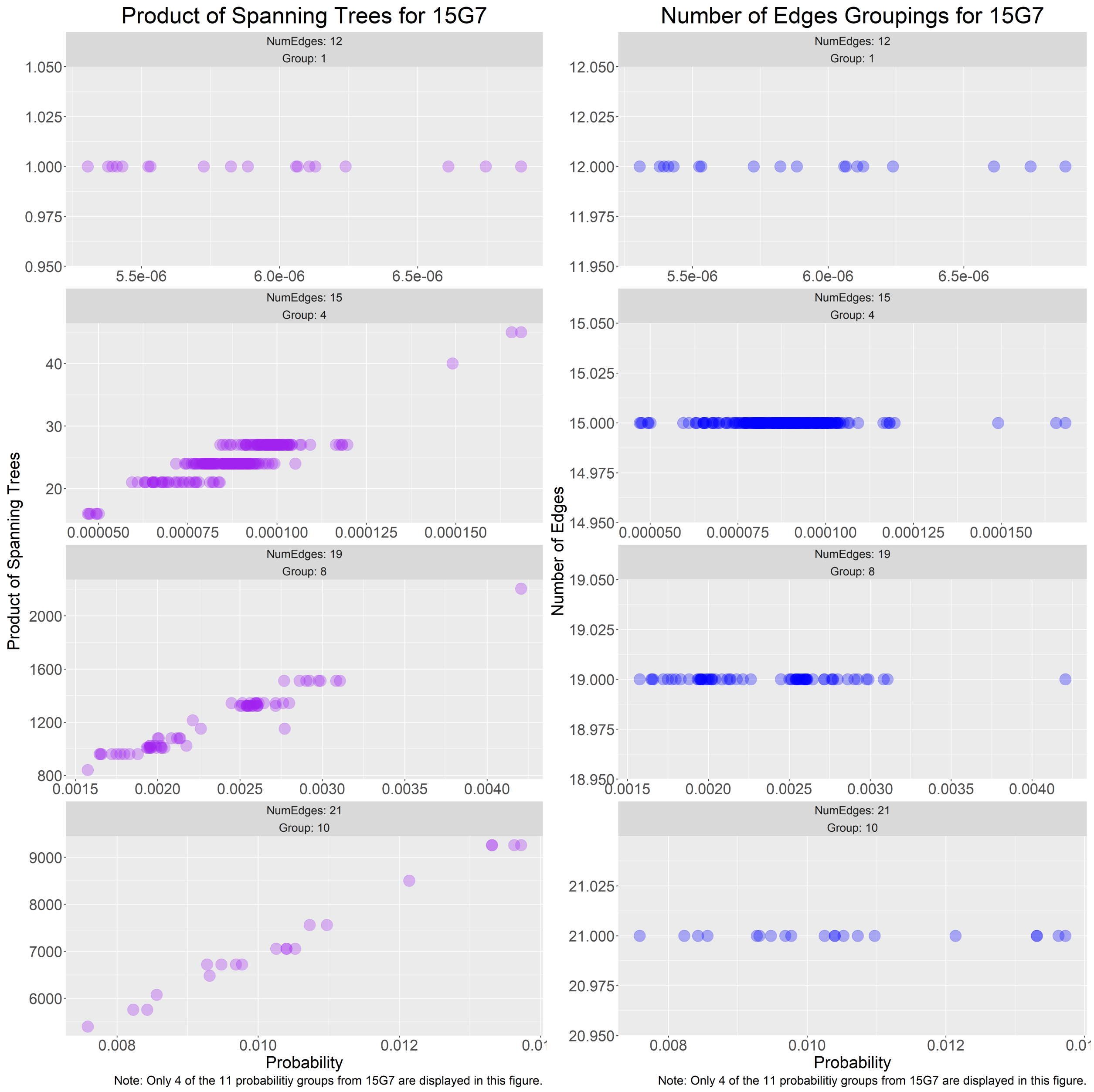
(a)

Number of Edges for 15G7 (Groups 1-3)

Note: Only 3 of the 11 probabilitiy groups from 15G7 are displayed in this figure.

(b)

Figure 14: The number of edges more clearly differentiates probability groupings than the product of spanning trees. On the other hand, product of spanning trees may be a more accurate predictor of probability within each grouping than number of edges.



Note: Only 4 of the 11 probabilitiy groups from 15G7 are displayed in this figure.

Note: Only 4 of the 11 probabilitiy groups from 15G7 are displayed in this figure.

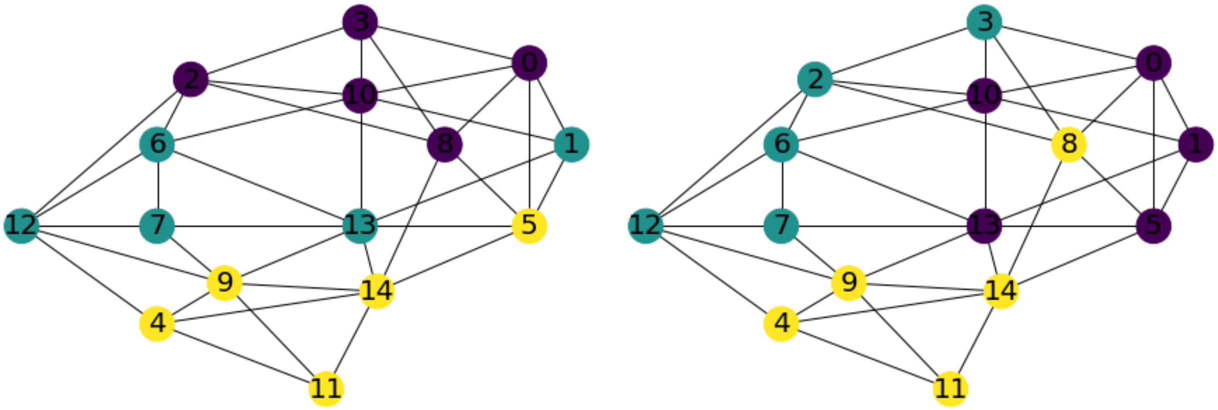
(a)

(b)

Figure 15: This figure plots four of the eleven probability groupings of 15G7 (Figure 13), comparing the ability of predicting probability between product of spanning trees and number of edges.

4.4 Small Differences in Probabilities

Knowing that the stationary distribution can be partitioned into probability groups and subgroups with sufficiently larger graphs is helpful in understanding larger probability differences, however, small differences in probability still exist within even the smallest subgroups. We revisit some exploration techniques from section 4.1 in comparing small probability differences within subgroups. Particularly we examine two plans, plans 750 and 475, both which are in the same subgroup from Group 10 of 15G7 (Figure 15). These plans both have a product of spanning trees of 5760 yet have probabilities of 0.0082293 & 0.0084270 respectively (Figure 16).



(a) Plan 750
Prob = 0.0082293

(b) Plan 475
Prob = 0.0084270

Figure 16: Plans 750 and 475 are in the same subgroup within probabiliy group 10 of 15G7, however, their probabilities still differ slightly.

From observing plans in the same subgroup, we did find there to be shape discrepancies matching up with the small probability differences. For example, in Figure 16, plan 475 has a slightly higher probability than plan 750. By observing the shapes present in both plans, we can see that both have a quadrilateral with two internal diagonals in the yellow district and quadrilateral with one internal diagonal in the teal district. What differentiates plan 475 from plan 750 is plan 475 has a five-side shape while plan 750 has a quadrilateral with one internal diagonal in the purple district. Thus, we believe that the frequency and size of shapes in a redistricting plan are not as measurable in the full ensemble of plans but are much more quantifiable within the subgroups of probability groupings.

4.5 Distributions of Probability Groups

After grouping probabilities based on the indicators mentioned above, we checked the distributions of probability groups for random graphs of sizes 9, 12, 15, 16, and 20. Under the same number of vertex graphs, plotting the average of each probability group for different graphs resulted in normally distributed and probability grouping sums with heavier right tails. Figure 17 depicts an example of the distribution of probability groups for multiple 16-vertex graphs. As the total number of edges within all districts are higher in the last three groups than these of the first three groups, there is a higher probability for a plan from the last three groups to be transitioned to in recombination than the first three groups.

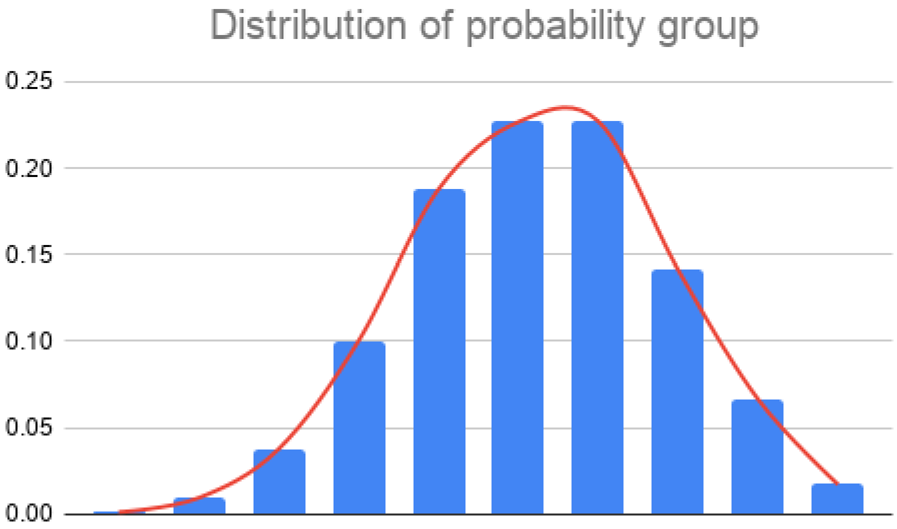


Figure 17: The average distribution of the sums of probabilities within each probability group for ten 16-vertex graphs.

Important to note that in analyzing many different graphs, grid graphs appear to be an outlier among the baseline of graphs to sample from. The probability grouping distributions for grid graphs seem to be randomly distributed with most of the observations clustering around the central peak (Figure 18). For the 4x4 grid, the distribution is almost uniformly distributed; the distribution for 3x4 grid is symmetric; the remaining two grids seem to have a random distribution. As there is no consistent distribution pattern for the grids, we assume that grid graphs are outliers in the graphs we found above whose probability groups are normally distributed.

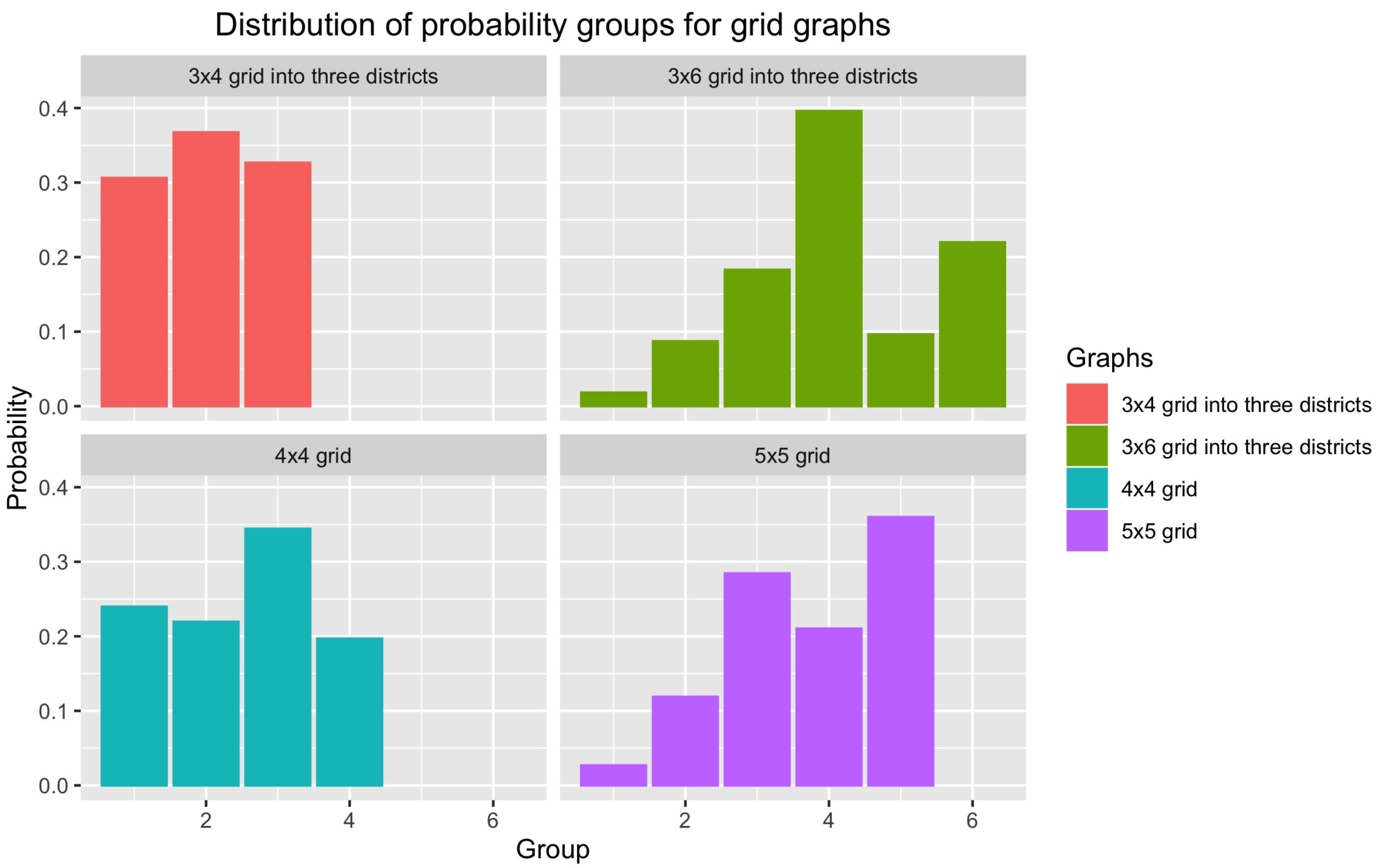


Figure 18: Distribution of the sums of probabilities within each probability group for grid graphs (3x4, 3x6, 4x4, 5x5).

4.6 Regression on Potential Factors

In order to identify variables that could have impacts on probabilities, we considered modeling various graphs using regression. We ran the regression on 4068 observations of 12-vertex graphs divided into three districts that have a similar number of redistricting plans. The factors we analyzed included:

- **Variance of Spanning Trees** (β_1): the variance of the individual district spanning trees from the average.
- **Product of Leaving Edges** (β_2): the product of the leaving edges for each district in a plan.
- **Variance of Edges** (β_3): the variance of the total edges in a district from the average edges in a district in a plan.
- **Most Edges** (β_4): the maximum number of edges between all pairs of districts in a plan.

At first, we ran linear regression on these independent variables and found the coefficient of variance of spanning trees to be positive. The positive coefficient surprised us as we expected less variance from individual spanning trees to result in a higher probability. Since signs of coefficient were the opposite of what we expected to see, we checked variance inflation factor (VIF) to see if there is high collinearity between predictors. However, the VIF value is 1.28 which indicates that there is a small amount of collinearity among the predictors which could allow us to assess accurately the contribution of predictor variables to the model. Also, after we applied the Breusch-Pagan Test to check whether the linear regression remains unbiased, we found heteroscedasticity does happen and as a result, regression prediction is currently not efficient.

One method we considered to fix the problem is Weighted Least Squares (WLS), and the model below demonstrates the correlation between the probabilities and related factors:

$$P_0 = 0.0053491 + 0.0001431\beta_1 - 3.13 * 10^{-6}\beta_2 - 0.0002652\beta_3 + 0.0000157\beta_4$$

Among all factors, variance of spanning trees, product of leaving edges and variance of edges are statistically significant due to computer p-values of 0. The number of edges in the union of the two largest districts has no direct impact on the probabilities. In explaining why variance of spanning trees has a positive coefficient, we assume that the gap between districts with fewer spanning trees and districts with more spanning trees is substantial, and a district with many spanning trees causes a higher probability. When the individual edges are less variable from the average, edges in the districts are balanced which leads to more spanning trees and a higher probability. Even though the product of leaving edges is statistically significant, its coefficient is very small and may not be particularly considered in the regression analysis.

4.7 Correlation with Degree of Metagraph

We considered a potential cause for the probability discrepancies to be the degree of the metagraph. This prompted us to check the degree of the metagraph for plans in the same probability group. But after we plotted the degree of the metagraph against the probabilities of redistricting plans, we found the degree of the metagraph to not be strongly correlated with the probability discrepancies. From Figure 20, though it is true that plans with higher probability tend to have a higher degrees in the metagraph, and plans with lower probabilities tend to have lower degrees, some plans with a high probability have a lower degree in the metagraph than plans with a lower probability. Seen in Figure 20, for plans with the degree of metagraph equivalent to 17, the plan probabilities range from approximately 0.01 to 0.0875. Thus, it is difficult to conclude that the degree of the metagraph has a direct impact on a redistricting plan's probabilities.

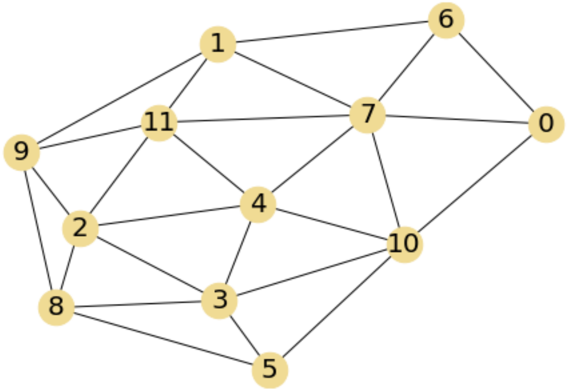


Figure 19: This 12-vertex graph (G25) pertains to Figure 20. *Note: This graph is planar but is drawn in this way to be more visually digestible.*

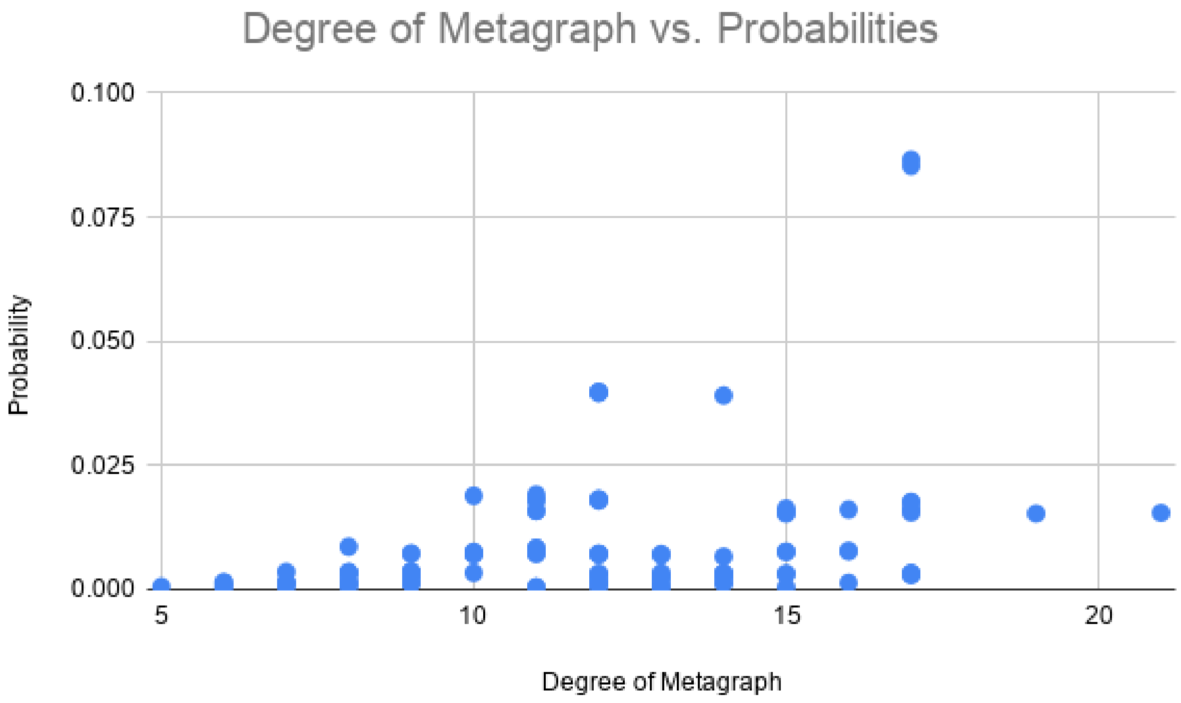


Figure 20: This figure plots the degree of the metagraph for all 118 plans in G25, divided into 3 districts (Figure 19), against their probabilities. The correlation appears to be a slight but positive relationship.

4.8 Special Considerations

In investigating trends and patterns in the stationary distribution, the following need to be taken into account due to the nature of this research and its applications to gerrymandering.

4.8.1 Negative Stationary Distribution

The first is the occurrence of the negative stationary distribution. The negative stationary distribution can be encountered when the dual graph is not ergodic. For dual graphs with disconnected metagraphs, the stationary distribution is calculated by repeatedly squaring the transition matrix to get a resultant stationary distribution. Figure 21 depicts a 9-vertex loop graph that isn't ergodic. With closer inspection, it is notable that through the process of recombination, every plan is not feasible to create from every other plan.

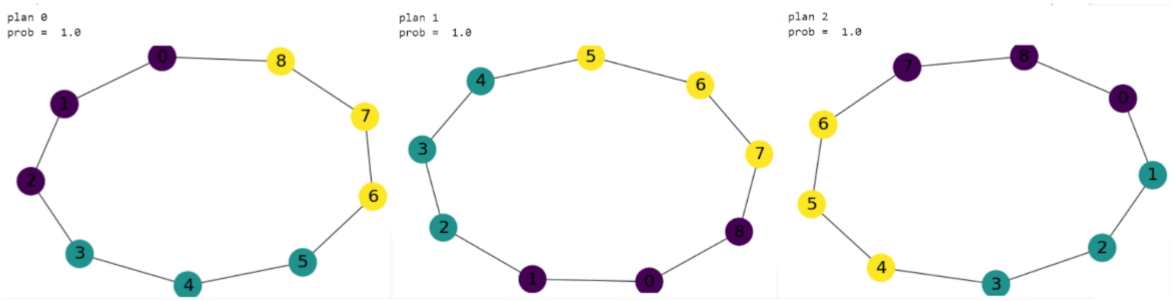


Figure 21: All three plans from the 9-vertex loop graph.

Notably important is that encountering a negative stationary distribution indicates that the metagraph is disconnected, however, not all non-ergodic dual graphs have a negative stationary distribution. Some non-ergodic graphs such as the Figure 21 are not particularly insightful since recombination, the mathematical sampling method we are choosing to create a baseline, does not allow us to analyze and draw connections between all possible redistricting plans of a particular dual graph. As a result, in analyzing dual graphs, we only consider ergodic dual graphs.

4.8.2 Recursive Graphs

In the process of analyzing the stationary distribution of various graphs, we found potential in graphs in which they could be expanded by recursively drawing a given shape inside each other. Figure 22 displays examples of variations of 'recursive' square graphs.

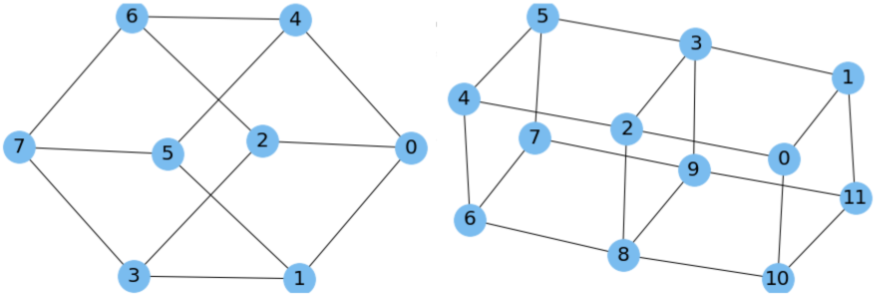


Figure 22: Recursive square graphs with 8 and 12 nodes respectively.

From analyzing recursive triangle, square, pentagon, and hexagon graphs, we found there to be around 3-5 exact probabilities that were scalable by a constant depending on what shape the graph "recursed" on. While these findings seem promising, we did not continue with recursive graphs since the application to redistricting plans resulted in infeasible plans. This is because redistricting plans of the "recursive graph" nature do not follow the conditions of simple connectivity in current redistricting regulations. As a result, in our research, "recursive graphs" cannot be considered among the distribution of feasible plans to be analyzed. With this in mind, we moved on in our exploration to analyze dual graphs that can be feasibly drawn within the plane where the individual districts can be drawn as entities not containing each other such that no edges cross each other. Also, even though recursive graphs cannot be feasible plans, it gives us insights of how probabilities have changed by scaling the dual graph and enables us to expand ideas to look at dual graphs which could be recursed side by side.

4.8.3 Metagraph Connectivity

In analyzing the stationary distribution, we only considered dual graphs which were ergodic. Thus, for each dual graph considered, we must ensure that the corresponding metagraph is connected. Some graphs in consideration had disconnected dual graphs yet had connected metagraphs, such as in Figure 1. In order to ensure that any considered disconnected dual graph with ergodic disconnected components has a connected metagraph, we conjectured the following proof to apply to all disconnected dual graphs.

Claim: Given an n-vertice disconnected dual graph, ω. If the k disconnected subgraphs have connected metagraphs, ω has a connected metagraph as well.

Proof.
Base Case: $k = 2$

Considering the context of node coloring, every node in the metagraph of ω is a unique coloring partition of the nodes in each of the k subgraphs (k_1, k_2).

Assume k_1 and k_2 are each ergodic such that every coloring partition of each subgraph can be attained through recombination. Note: Recombination can only occur when the there are exactly two adjacent components within a disconnected subgraph that differ from the starting to the ending coloring partitions.

The coloring partitions of ω are comprised of different combinations of the coloring partitions of k_1 and k_2. Since k_1 and k_2 are disconnected in the dual graph, all possible coloring partitions in ω can be attained through recombination of the coloring partitions of k_1 and k_2 by mapping every coloring in k_1 to each coloring in k_2. Thus, for $k = 2$, given an n-vertice disconnected dual graph, if the two subgraphs have connected metagraphs, the dual graph must also have a connected metagraph.

Inductive Hypothesis:

Fix $k = u$, we assume the proposition holds true for u.

Inductive Step:

Assume the above claim holds for $k = u$. Let's reference $\alpha = k_1, ..., k_u$ which are the subgraphs of ω when $k = u$. Since we know that each coloring of ω can be attained through recombination of any of the subgraphs in α, by adding an additional ergodic subgraph such that ω has $u+1$ subgraphs, we know each of the $k_1, ..., k_u, k_{u+1}$ subgraphs are ergodic such that every coloring partition of each subgraph can be attained through recombination. Since $k_1, ..., k_u, k_{u+1}$ are disconnected in the dual graph, all possible coloring partitions in ω can be attained through recombination of the coloring partitions of $k_1, ..., k_u, k_{u+1}$ by mapping every coloring in k_i to each coloring in k_j such that $1 \leq i, j \leq u + 1$. Thus, for $k = u + 1$, given an n-vertice disconnected dual graph, if the $u + 1$ subgraphs are ergodic, the dual graph must be ergodic as well.

□

5 Discussion, Limitations, and Future Work

While the shape and size of the stationary distribution is still unknown, this research has identified measurable ways in which to construct groups and subgroups of the probabilities in the stationary distribution. These factors are helpful and insightful, however, currently limited in their general applications.

First, the largest graph analyzed for this paper is a 25-vertex graph due to the computational intensity required to generate plans, calculate the stationary distribution, etc. Our findings are quite generalizable for graphs of size 25 or less, however, verifying if our findings apply to larger graphs requires much more computer memory than we currently have access to. For example, we assume that grid graphs are outliers from the ensemble of graphs we analyze in terms of their stationary distribution. We only consider grid graphs of size 25 or less. Further computation on large graphs should be conducted in order to verify our assessment of grid graphs of larger sizes. This is necessary because as the stationary distribution applies to the ensemble of possible plans for congressional districts as it pertains to gerrymandering, abstracting congressional districts results in extremely large graphs.

Secondly, we understand that the factors we have analyzed in finding probability groups and subgroups are not exhaustive of all the potential factors. This paper attempts to group probabilities in the stationary distribution into groups and then into subgroups. In addition, the this paper attempts to find the combination of factors that acts as an accurate predictor for a redistricting plan's probability. As a result, further work can be done in creating a more accurate model for predicting redistricting plan probabilities. Particularly in regression analysis, changing which variables are included, transforming variables, controlling for confounding factors, and tuning interactions could be another area of interest for predicting minute differences in probabilities. In addition, the current regression attempts to predict probabilities without the group distinction at first. Partitioned model construction should be tested under a similar regression model before and after including more potentially significant factors. Moreover, other classification models could be explored in modeling any combination of these variables against probabilities in the stationary distribution.

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 is not the case as recombination has been applied where a starting and ending plan have two or more districts not in common [5, 4, 1]. Moreover, the conditions set on recombination can be modified to explore smaller and potentially different ensembles of plans. 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. Additionally, subtly changing the steps of recombination could possibly change the stationary distribution and thus the ensemble of plans sampled from. As a result, modifying the procedure of the Recombination Markov Chain is another area of interest in investigating the processes that generate the ensemble of plans. Ultimately, more specific investigation can be conducted on recombination, the process at which the ensemble plans is generated, and the multitude of factors that may or may not have significance in predicting probability.

Acknowledgments

This report was written in accordance with the undergraduate summer research experience at Claremont McKenna College. We warmly thank Professor Sarah Cannon, from the Department of Mathematical Sciences, who has generously provided us with excellent research support, guidance, and mentorship.

References

[1] Angulu, Buck, DeFord, Fain Duchin, Hully, Khan, Schutzman, and York. Study of reform proposals for chicago city council. Technical report, MGGG Technical Report, 2019.

[2] Maria Chikina, Alan Frieze, Jonathan Mattingly, and Wesley Pegden. Separating effect from significance in markov chain tests. Preprint. Available at https://arxiv.org/abs/1904.04052, 2020.

[3] Maria Chikina, Alan Frieze, and Wesley Pegden. Assessing significance in a markov chain without mixing. *Proceedings of the National Academy of Sciences*, 114(11):2860–2864, 2017.

[4] Daryl DeFord, Moon Duchin, and Justin Solomon. Comparison of districting plans for the virginia house of delegates. Technical report, MGGG Technical Report, 2018.

[5] Daryl DeFord, Moon Duchin, and Justin Solomon. Recombination: A family of Markov chains for redistricting. Preprint. Available at https://mggg.org/uploads/ReCom.pdf, 2020.

[6] Moon Duchin. Outlier analysis for pennsylvania congressional redistricting. *LWV vs. Commonwealth of Pennsylvania Docket No. 159 MM 2017*, 2018.

[7] Moon Duchin, Taissa Gladkova, Eugene Henninger-Voss, Ben Klingensmith, Heather Newman, and Hannah Wheelen. Locating the representational baseline: Republicans in massachusetts. *Election Law Journal: Rules, Politics, and Policy*, 18(4):388–401, 2019.

[8] Moon Duchin and Bridget Eileen Tenner. Discrete geometry for electoral geography. *arXiv preprint arXiv:1808.05860*, 2018.

[9] Gregory Herschlag, Han Sung Kang, Justin Luo, Christy Vaughn Graves, Sachet Bangia, Robert Ravier, and Jonathan C. Mattingly. Quantifying gerrymandering in North Carolina. *Statistics and Public Policy*, 7(la):1–17, 2020.

[10] Camryn Hollarsmith. Stationary distribution of recombination on 4x4 grid graph as it relates to gerrymandering. Senior Thesis at Scripps College in Partial Fulfillment of the Degree of Bachelor of Arts, 2020.