Analysis of South Carolina State House Districting

Laurence Lo and Taiyo Williamson and Jerry Chen
Code link: `https://github.com/laurencelo1/gerrymandering-analysis.git`

May 13, 2025

1 Introduction

Redistricting is the process of redrawing electoral district boundaries, a procedure that directly affects the balance of political representation in a state. In the USA, redistricting typically occurs every 10 years during the census. However, this process is not merely a technical exercise; it is a highly political game which will determine which party can gain more seat advantage by drawing unfair districts which separate strong opposition political alignment population and group pro-partisan yet loose areas to create a form of majority. This method is called **Gerrymandering**.

In this report, we are conducting analysis on **South Carolina**'s districting map. South Carolina, like many other states, has a history of contentious redistricting battles. The state's redistricting process is controlled by a state legislator which is heavily prone to partisan influence. Moreover, South Carolina designed Voting Right Acts(VRA) districts after 1990 census to comply the Voting Right Act in 1965 to ensure that minority groups can have equal opportunity to elect representatives of their choice. However, after 2020 census, this redistricting efforts led to legal challenges as more recent 2020 SC 1st Congressional District was being ruled by a federal court as unconstitutional racial gerrymander for favoring Black votings. In 2024, US Supreme Court reversed this decision as it is more partisan than racial. Continuing the debate on whether the gerrymandering is by partisan or racial as South Carolina is heavily accused of favoring Republicans since 1990s.

The study will focus on the South Carolina State House to see if gerrymandering do exist in SC. The procedure is to first take South Carolina's districting data, conduct **MAUP** to get **shape file**, and then do **Markov Chain** on both 2018 and 2020 election data. Finally, a short burst will also be conducted to verify correctness of MAUP produced shp file to really tell what kind of gerrymandering does South Carolina have.

1.1 Background: Map Political History

1.1.1 South Carolina State Conference of the NAACP v. Alexander

The case of South Carolina State Conference of the NAACP v. Alexander centered on challenges to the South Carolina State House redistricting map following the 2020 Census. The redistricting process was significantly delayed by multiple factors, creating a compressed timeline that raised concerns about proper review and implementation.

The 2020 Census results were released on August 12, 2021, following delays caused by the COVID-19 pandemic. Despite these results becoming available, the South Carolina legislature adjourned in fall 2021 without addressing its redistricting responsibilities, prompting the American Civil Liberties Union (ACLU) to file a malapportionment lawsuit on October 12, 2021.

The House and Senate maps were finally drawn on December 10, 2021. However, plaintiffs argued that the legislature's delayed timeline did not allow sufficient opportunity for review and implementation of legally compliant maps before the 2022 election cycle. The case eventually settled on May 5, 2022, with findings that the initial maps improperly packed and cracked Black communities—a process plaintiffs' lawyers referred to as "bleaching" districts. As a result of the settlement, revised maps were drawn on June 17, 2022, which were then implemented for the South Carolina general election held on November 8, 2022.

2 Methods

The method used are **MAUP** for data cleaning and create shape file, **Markov** chain to get **Ensemble** and **Marginal Box Plots**, and **Geopandas** to read shp. files produced by MAUP to do short burst analysis.

2.1 Data Collection and Preparation

2.1.1 Imports for each method

There are multiple datas got from Redistricting Hub.

- 2021 State Senate District Data: 2021 State Senate Approved Plan[1]
- Population data: From 2020 Census Redistricting Data (P.L. 94-171) Shapefiles [2]
- 2020 County Data: From 2020 Census Redistricting Data (P.L. 94-171) Shapefiles[3]
- 2018 Election Data: VEST 2018 SC precinct and election results [4]
- 2020 Election Data: VEST 2020 SC precinct and election results [5]

Import necessary libraries in order to use MAUP:

`pandas`, `geopandas`, `maup`, `from maup import smart_repair`

For Markov Chain:

`pandas`, `geopandas`, `maup`, `matplotlib.pyplot`
`gerrychain` - `Graph`, `Partition`, `proposals`, `updaters`, `constraints`, `accecpt`
`MarkovChain`, `Election`
`gerrychain.tree` - `bipartition_tree`
`gerrychain.updaters` - `cut_edges`, `Tally`
`gerrychain.proposals` - `recom`
`gerrychain.accept` - `always_accept`
`functools` - `partial`
`gerrychain.metrics` - `efficiency_gap`

For Short Burst (Same as Markov Chain Imports, but more imports at below):

`gerrychain.optimization` - `SingleMetricOptimizer`, `Gingleator`
`numpy`

*The Election Data used on Markov Chain for 2018 is Attourney General(G18ATG) and 2020 is Senate(G20SEN)

2.2 MAUP

Modifiable Areal Unit Problem(MAUP) is the process of combining data of different geographic scales (block, precincts, and counties) and aggregating them in an unbiased manner. The end result is a consistent map that we will then use to see how past election results change when we change the district boundaries.

We used the MAUP Python library and smart_repair method to fix topological inconsistencies in our geographic data. The maup.doctor method helped us identify and resolve issues where precincts crossed county lines or where there were gaps and overlaps between adjacent polygons. This preprocessing step was crucial to ensure accurate aggregation and analysis of demographic and electoral data across different geographic scales.

We then made sure that the voting precincts are properly nested within county boundaries. We used a minimum rook length of 30 meters, meaning two districts are adjacent if they share a boundary of at least 30m.

Lastly, we took all the group codes that included Black as a partial race and combined them into one column that will be used in our Short Burst Analysis. There were about 30 codes total for population groups that were partially Black.

2.3 Ensemble Analysis

Markov Chain Monte Carlo(MCMC) is a computational method for sampling from a complex probability distribution which works by building a Markov Chain - a sequence of states(districting map) where each state is derived from previous one using a specific proposal method. **Recom** is the proposal used which will talk later on. Over many iterations, the chain converges to a **stationary distribution** as a representative sample of all possible states(districting plan).

The reason to use MCMC is due to redistricting is a complex problem with enormous number of possible districting plans. MCMC allows examiner to explore this space efficiently, creating a representative **ensemble of valid districting plans**. By analyzing this ensemble, examiner will be able to measure how a given map compares to a large set of alternative maps.

2.3.1 About GerryChain

Gerry Chain is a Python library for redistricting analysis using MCMC. It has following useful tools:

- Representing the districting map as a **Graph** which is a network of nodes represnenting geographic areas.
- Defining a **Partition** as the initial districting plan.
- Implementing various **MCMC proposal methods** (ways to move from one map to another)
- Applying **constraints** to ensure maps meet legal requirements(e.g population balance)
- Calculating metrics such as **Efficiency Gap**

2.3.2 Parameter Choices

Parameters in the python notebook are being carefully considered in order to run chain accurately. Parameters listed below:

- **num_steps**: Number of steps to run a Markov Chain
- **proposal**: "recom" for randomly selecting 2 adjacent districts and recombining them, then re-dividing them into 2 new districts while maintaining compactness.
- **constraints = [compactness_bound, pop_constraint]**: **compactness_bound** as to ensure district are reasonably compact(avoiding bizzare shapes), and **pop_constraint** ensures all districts have similar populations

- **my_updaters** = Self difined updaters for stuff user wish to have. E.g, cut edges, population, etc.
- **initial_partition**: Dictionary object output after running Partition(sc_graph, assigments="You choose", updaters = your_updaters
- **ideal_population**: Sum of initial population list divided by initial partition's dictionary length.
- **epsilon**: Population tolerance parameter. Controls the allowable **deviation from the ideal population size** for each district in the generated districting plans. In this analysis, 0.1 as each population vary by $\pm 10\%$ from the ideal population.
- **cut_edge_list**: List of cut edges produced after doing GerryChain
- **dem_win_list**: List of districtings of seat counts that Democrats wins
- **EG_list**: List of efficiency gap produced after doing GerryChain for wasted vote record

2.3.3 Procedure of MCMC

Below are the steps doing MCMC

1. Load graph and gdf from Maup'ed shp. file using geo pandas
2. Create Election list of election object of different elections such as a party's attourney general, senate, governer, or presidential election.
3. Create updaters you wish to have. E.g Cut edges, population tally, democratic_wins, etc.
4. Do initial partitions to get partition by inputting graph got from geopandas, and then set desired assignments("SEND" for Democrat senate, etc.) along with self defined updaters.
5. Get ideal population from produced initial partition
6. Set proposal as partial method **recom** proposal with epsilon 0.1 using ideal population got earlier.
7. Set compactness_round for constraints
8. Finally, do Markov Chain with all result got from before and append it to 3 lists: Cut edge, Democrat wins, and Efficiency Gap list.
9. Set steps and call **enumerate(chain)**

10. At last plot marginal box plots and histogram from 3 lists appended from Markov Chain

*Note: From step 8 - 10, you need a for loop to make those do exactly 2 times which will be 2 ensembles mentioning below

2.3.4 Ensembles

There are 2 ensembles being run by the Markov Chain. Ensemble 1 will be run with 20000 steps and Ensmble 2 is double the previous as 40000 steps. The purpose of 2 ensembles is to see if increasing the number of steps changes the distribution(testing convergence) and ensure the analysis is through by sampling more districting plans. Normally, 2 ensembles got from Efficiency Gap and Democrat/Republican wins should be consistent across both of the results which suggest stability. Running a second ensemble allows more comprehensive exploration of the districting space.

2018 MCMC Histograms

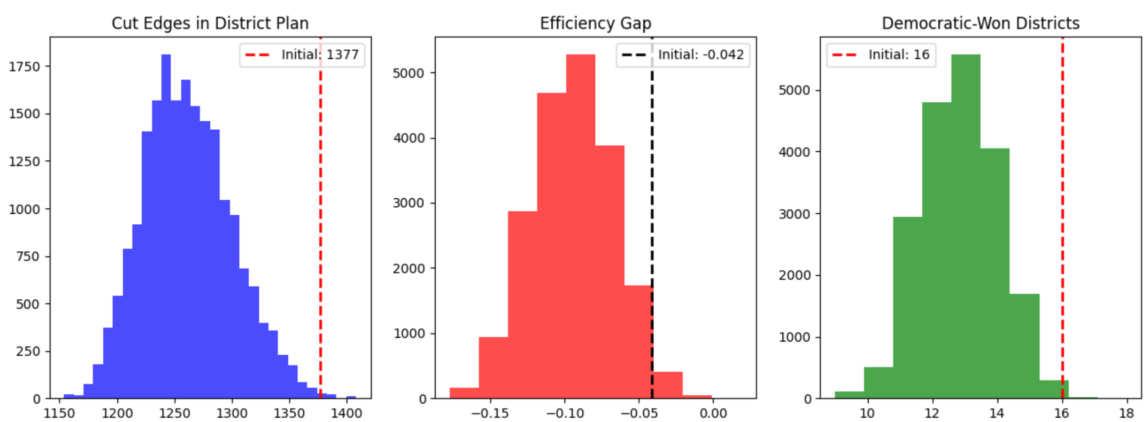

Figure 1: 20000 Step MCMC for 2018 Ensemble

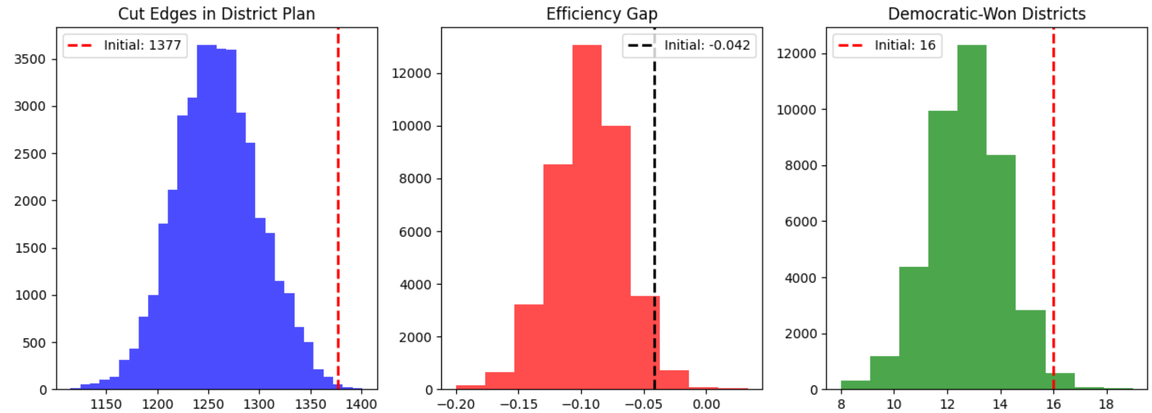

Figure 2: 40000 Step MCMC for 2018 Ensemble

We can see in 2 histograms of 2018, the difference are not that varied which can be evidence of convergence even with increasing number of steps.

2020 MCMC Histograms

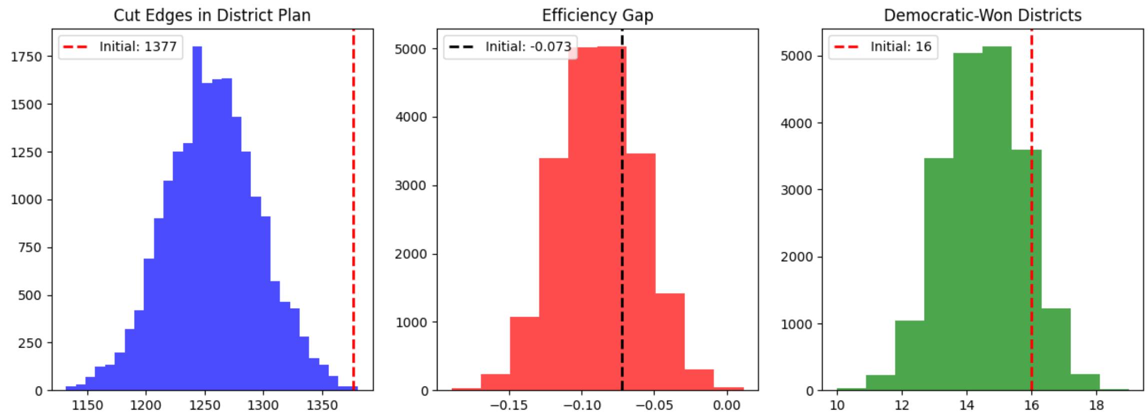

Figure 3: 20000 Step MCMC for 2020 Ensemble

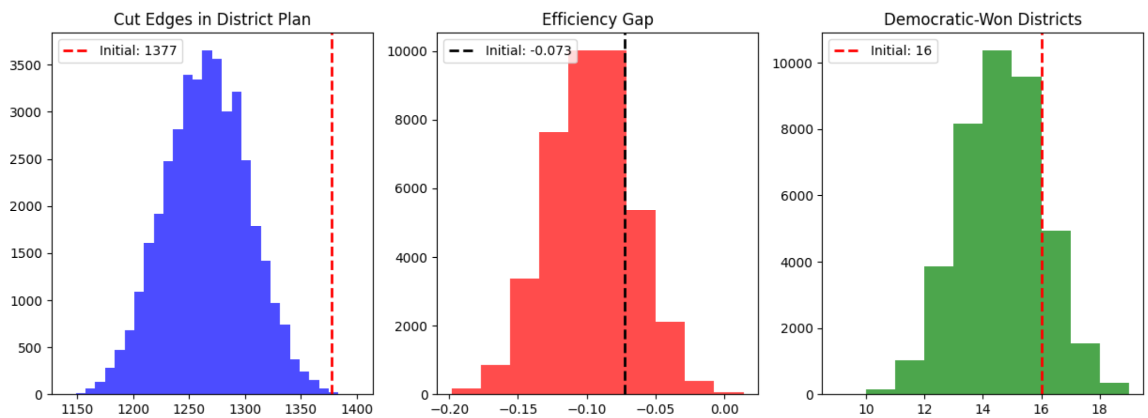

Figure 4: 40000 Step MCMC for 2020 Ensemble

We can also see in 2 histograms of 2020, the difference are not that varied which can be evidence of convergence even with increasing number of steps.

Given strong inderect evidence of convergence as 2 ensembles of 2018 and 2020 are almost identical, the MCMC chains are exploring the same region of possible districting plans. More steps do not significantly change the distribution.

However, from what we got, we can see the **Efficiency Gap** is mostly negative for Democrats is overall negative, suggesting that a **systematic bias** towards Republicans exist in the districting map. But here is a twist, in **Democrat wins** part, Democrats can still win around **14-15** districts suggest that **geographic clustering** for Democratic voters may be a factor. This implies that Democrats may be heavily concentrated in a few districts, ideally urban, while Republicans are more efficiently spread out.

2.3.5 Outlier Graph

The initial plan has **1377 Cut Edges**. The value shown is **significantly to the right of the main distribution** of Cut Edges which deviates away from normal range of **1260 to 1300** Cut Edges. This proves the initial plan is an outlier. This can be a sign of **gerrymandering** where high Cut Edges can indicate **cracking**(spreading a voter group thinly across multiple districs) as well as may suggest **non-compact districts**.

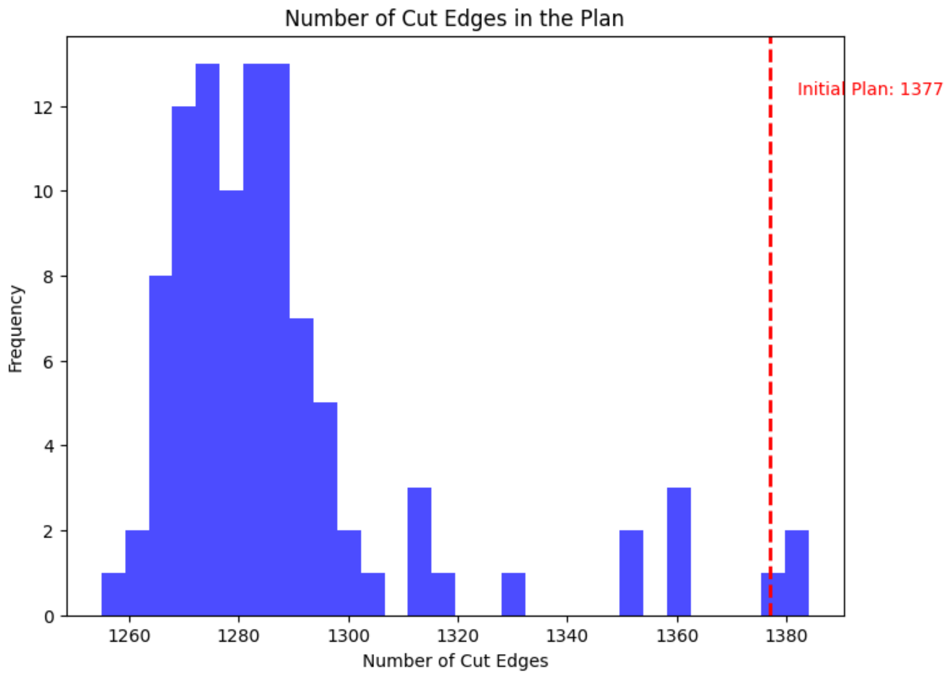

Figure 5: Outlier Graph

2.4 Marginal Box Plots

Marginal Box Plot is a specialzed visualization used in redistricting analysis to reveal **how voes are distributed across districts**. This is also done after ensembles got from doing Markov Chain. Marginal Box Plots generated from each districting plan's calculated **percentages of votes** for a target party in each district. Which will then sort these percentages in ascending order for each plan.

These sorted percentages are then plotted as a box plot. Each box representes **spread of vote percentages** in one of the sorted districts actross all plans in the ensemble. The **median(middle line)** of each box shows the typical vote percentage in that district. The **interquartile range** shows the middle 50% of values. **Red dots** are vote percentages of initial plan for the selected party, if it is out of box plot, then it is an outlier.

2018 Box Plots for 2 Ensembles

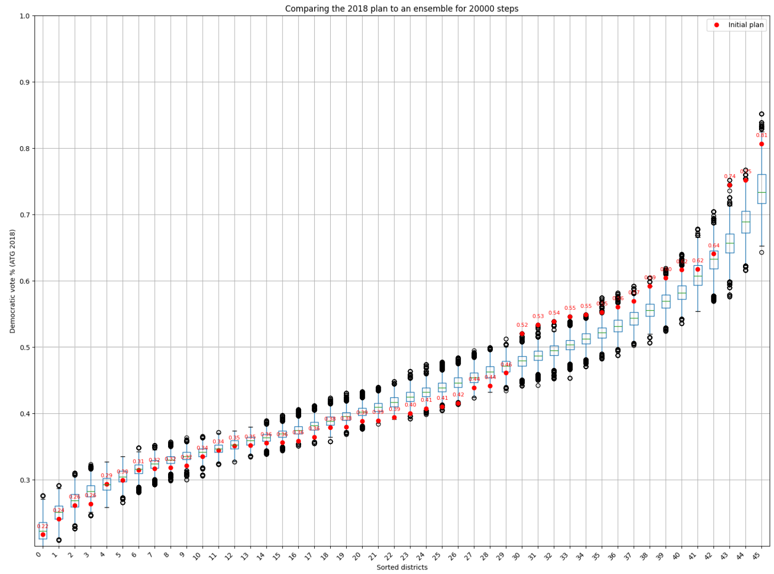

Figure 6: 2018 Box Plot for 20000 step

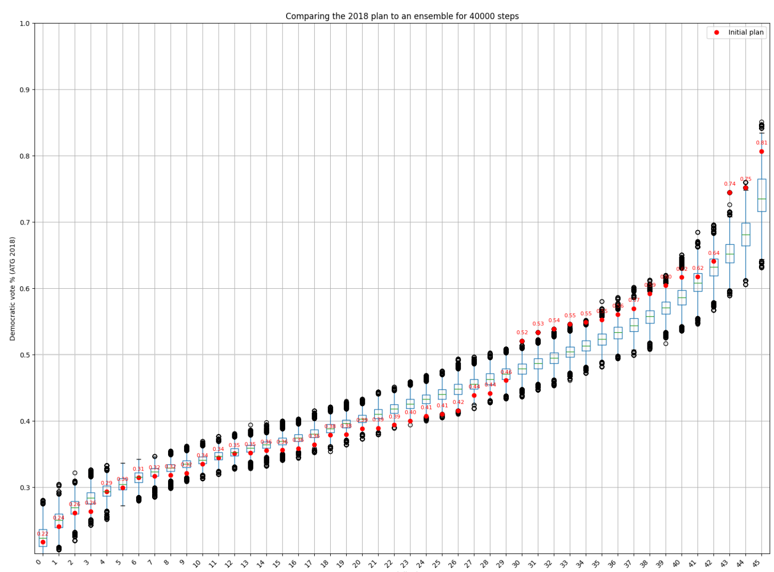

Figure 7: 2018 Box Plot for 40000 step

2020 Box Plots for 2 Ensembles

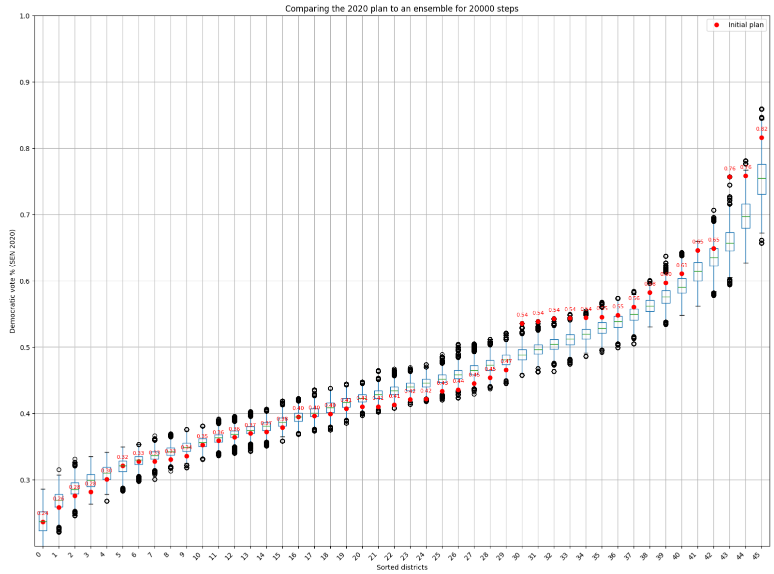

Figure 8: 2020 Box Plot for 20000 step

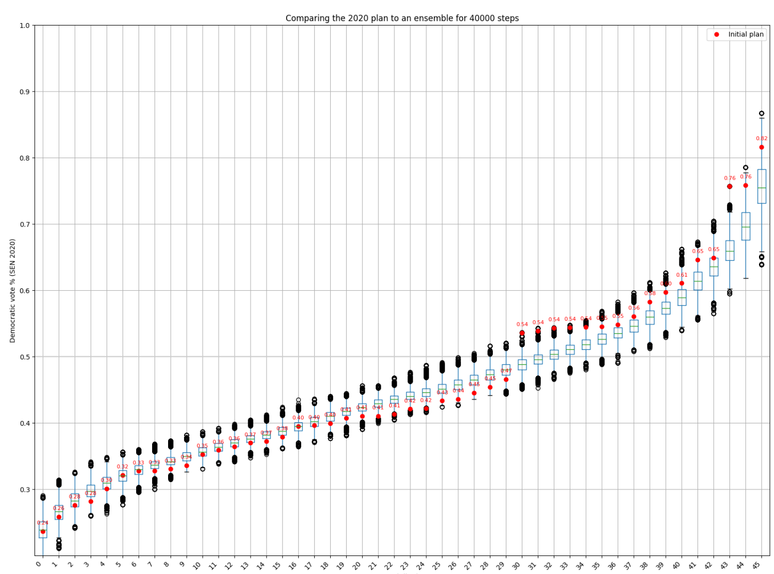

Figure 9: 2020 Box Plot for 40000 step

For 2018, 2 ensembles are being plotted. As shown, red dots(initial plan) are generally lowere middle of the box plots in the range of district 13 to 30 indicating more Democrat supports(box) than expected(red dot), meaning **cracking** Democratic voters to make them minority in each district in order to reduce Democrats winning the seat. However, in 31 to 45 district shows Democrats has way lower percentage than initial plan meaning these districts are being **packing** Democratic voters.

For 2020, the results are similar to 2018 ones, but the High-Democrat district is being "improved" as red-dot increased than 2018 ones meaning more **packing** may occur. Since district 13 to 30 remain similar for both years, yet High-Democrat district is bing "improved", that slightly stronger **packing** and more consistent **cracking** occurs.

From the box plot, it can be strongly implied that gerrymandering happens by both **cracking** and **packing** Democratic votes given consistent red dots below the center box from district 5 to 30(cracking) and abnormally high center box than red does from district 30 - 45(packing).

2.5 Short Burst Analysis

Short Burst analysis is a diagnostic technique used to examine how a Markov Chain behaves during the early steps of its execution. In the context of redistricting, it helps assess the quality of the proposal mechanism—i.e., how effectively and efficiently the chain explores the space of valid districting plans. When performing an ensemble analysis with the Short Burst optimization method, it hopes to maximize the score function, and select the next graphs/nodes in the chain to best optimize said score.

The Short Burst method, as implemented in the GerryChain library, involves generating multiple short chains (bursts) starting from a common initial plan. Each chain runs for only a small number of steps (e.g., 100 to 1000), and key properties—such as number of cut edges, partisan outcomes, and other summary statistics—are recorded at each step. By aggregating results across many bursts, the method allows for quick testing of proposal and constraint settings before committing to longer chains. It is especially useful during development to identify tuning issues and to ensure that the chain is capable of producing a diverse set of districting plans that meet the necessary legal and geographic constraints.

For the Min # Cut Edges Observed, as the chain iterates for longer, the less cut edges are observed, until it levels out at roughly 900 cut edges. Over time, the Short Burst method is discovering plans with more compact boundaries and contiguous districts, which in turn lowers the amount of cut edges.

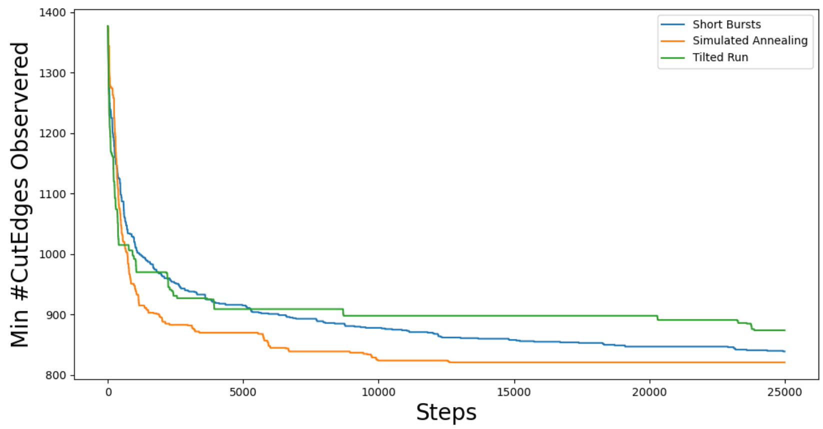

Figure 10: SC Min Number Cut Edge

For the Max Score Observed, we see an upward trend in the vertical axis, until the methods/lines trail off after about 22000 steps. The score measures the number of majority-minority districts in a plan, and the maximum for that plan. For our state of South Carolina, Short Bursts maximizes black voting age population (BVAP).

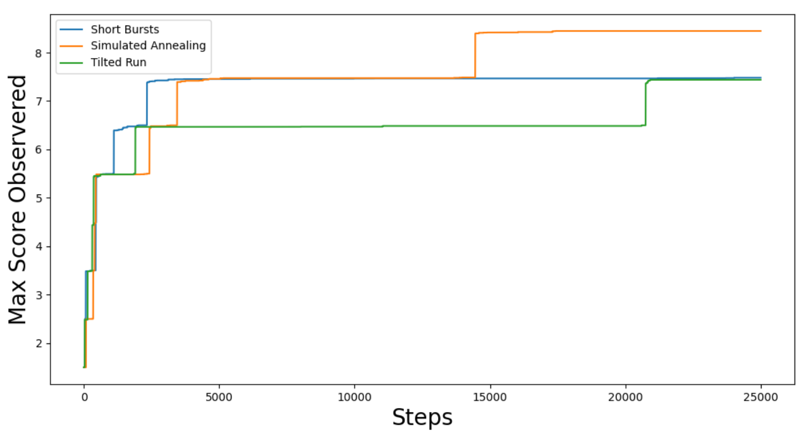

Figure 11: SC Max Score

3 Discussion/Conclusions

3.0.1 Data Cleaning and MAUP

To start, a foundational step in our redistricting analysis was the construction of a clean and complete shapefile for South Carolina's voting tabulation districts (VTDs). We utilized the MAUP Python library to manage spatial aggregation and alignment between demographic and electoral data layers.

Our final shapefile was topologically clean, contiguous, and matched population and voting data accurately across all precincts. This ensured the integrity of the downstream analysis. Given the difficulty of obtaining harmonized and clean shapefiles for many states, our processed dataset can serve as a high-quality, reusable resource. By producing a clean and accurate shapefile, we reduced noise in our simulation results and enabled reproducibility.

3.0.2 ReCom-Based Markov Chain Sampling

To explore the space of legally valid districting plans, we implemented a Markov Chain Monte Carlo (MCMC) simulation using the ReCom (Recombination) proposal method from GerryChain. ReCom merges adjacent districts and repartitions them while preserving population balance and contiguity, making it well-suited for navigating high-dimensional plan spaces more effectively than edge-flipping proposals.

With the constructed dual graph (metagraph) of South Carolina, we ran a chain of several thousand steps, collecting summary statistics at each step. Specifically, we tracked:

- Efficiency Gap (EG), a measure of partisan fairness;
- Democratic-Won Districts, capturing partisan outcomes;
- Number of Cut Edges, used as a proxy for boundary compactness.

Histograms of these metrics across the ensemble revealed clear patterns. The efficiency gap distribution had a median of roughly -0.1 with an initial EG of -0.042, suggesting that many sampled plans were supposedly less balanced. However, the EG metric is also not as trustworthy in this context. The number of Democratic-won districts varied across the ensemble, generally ranging between 12 and 14 out of 46 seats, demonstrating the sensitivity of partisan outcomes to district boundaries. The number of cut edges averaged roughly 100 less than the enacted plan's, indicating that the districted plan could have been more compact.

3.0.3 Short Bursts

We observed that the minimum number of cut edges across bursts decreased quickly and stabilized near 900, indicating that the chain was rapidly finding more compact alternatives. The short bursts method was ran with 5000 bursts 5 times, to run it long enough for the 25000 steps deemed acceptably long enough for the score function to level off.

We observed data we are generally satisfied with, which implied that there was a maximum majority-minority district count (districts with majority BVAP) of 8 in some of the most extreme sampled plans later in the short bursts chain run.

3.0.4 Final Word

From the analysis, there is a clear discovering that South Carolina has **both partisan racial gerrymandering** gerrymandering. There are districts being **packed** for Democrats for minorities which causes lots of **wasted votes** despite still having 13-15 seats. There are also districts being **cracked** for potential Democratic lead districts making them harder to win despite the party still made it through during election which were close calls. Therefore in 2022, a redrawing of the district was conducted in response of these concerns.

4 Acknowledgements

We would want to thank Professor Ellen Veomett for guiding us through doing analysis and constructing code bases. We would not have good and accurate plots/graphs without the help.

References

[1] Redistricting Hub. 2021 south carolina state senate approved plan, 2021. Last accessed 4/16/2025, https://redistrictingdatahub.org/dataset/2021-south-carolina-state-senate-approved-plan/.

[2] Redistricting Hub. South carolina block pl 94-171 2020 (by table), 2021. Last accessed 4/16/2025, https://redistrictingdatahub.org/dataset/south-carolina-block-pl-94171-2020-by-table.

[3] Redistricting Hub. South carolina county pl 94-171 2020, 2021. Last accessed 4/16/2025, https://redistrictingdatahub.org/dataset/south-carolina-county-pl-94171-2020/.

[4] Redistricting Hub. Vest 2018 south carolina precinct and election results, 2021. Last accessed 4/16/2025, https://redistrictingdatahub.org/dataset/vest-2018-south-carolina-precinct-and-election-results/.

[5] Redistricting Hub. Vest 2020 south carolina precinct and election results, 2021. Last accessed 4/16/2025, https://redistrictingdatahub.org/dataset/vest-2020-south-carolina-precinct-and-election-results/.