Incorporating Risk into Algorithmic Redistricting

Assessing and Optimizing Redistricting Algorithms with a Bias-Variance Trade-Off and Real-World Voting Data

Ivan Sinyavin and Owen Xu Li

December 12, 2024

Contents

1 Introduction

Our paper aims to refit existing algorithmic redistricting methods to explicitly optimize for a specific party's objectives. Redistricting refers to the process of redrawing electoral district boundaries, which, in the United States, it typically conducted after each decennial census to reflect population changes. While intended to ensure equal representation, redistricting can be manipulated in a practice known as gerrymandering, where boundaries are drawn to benefit a particular party. Generally, gerrymandering can lead to unequal representation by either "packing" opposition voters into a few districts or "cracking" them across many districts to dilute their influence.

While current research primarily focuses on models that assess whether a certain district configuration is gerrymandered or algorithms that broadly simulate gerrymandering, these approaches often lack explicit mechanisms and practical implementation toward the goal of optimizing toward one party or another. Our goal is to develop an algorithm that explicitly optimizes towards a single party's objectives, incorporating past election data to redistrict maps to create districts with equal population, a measure of compactness, and, most importantly, strategically distributed voters. Furthermore, we will use real-world voting data to construct measures of expectation and variance to assess to what extent a certain district configuration can benefit a party — and to that extent, use these metrics to optimize the redistricting towards a specific party's interests. We also fill a gap in optimizing long-term favor-ability of redistricting plans. Since the census and subsequent redistricting (on the congressional level) occurs every ten years, as time passes, the variance of voting result from the initial district plan often deviates. We capture the extent of this deviation from past historical data and propose ways to incorporate the variance of these temporal shifts into the redistricting algorithm. Therefore, our algorithm will be better equipped to ensure party success in both the short term (1–2 years) and the long term (7–8 years).

We will build off existing literature on this subject matter. In "Automated Redistricting Simulation Using Markov Chain Monte Carlo," (Benjamin Fifield and Tarr, 2020), the authors develop develop a Markov Chain Monte Carlo method to generating redistricting maps under a set of constraints. The algorithm models redistricting as a graph-cut problem, where smaller precincts are swapped between districts and assessed whether or not they minimize an objective function. In the algorithm, small connected components of precincts are identified and iteratively swapped between districts, which is controlled by a probability distribution that maintains a balance between exploration and acceptance of valid configurations. Swaps are evaluated against an objective function that assesses equal population requirements, compactness, and party distribution. A Metropolis-Hastings acceptance probability determines whether to accept a proposed swap. Furthermore, the "Sequential Monte Carlo" algorithm proposed by McCartan and Imai, 2023 builds on the above algorithm by drawing multiple plans in parallel rather than sequentially, which facilitates a more efficient exploration of the space of redistricting plans, which allows the algorithm to converge faster and requiring fewer samples to approximate the target distribution.

While the Fifield algorithm incorporates partisan considerations indirectly by assessing party distribution in its objective function, it does not explicitly optimize for partisan objectives. We will extend this algorithm by explicitly incorporating past election data to evaluate and optimize the distribution of voters in districts for a targeted partisan advantage.

2 Data Sourcing

2.1 Past Voting Behavior in Congressional Elections

Our main modeling objective by using real-word data is to understand relationship between the percentage of voters supporting a particular party in one district and that party's electoral performance in subsequent elections. This relationship informs how a slight advantage, for example, translates into varying probabilities of victory and margins of success — which becomes in integral piece in optimizing performance toward one party in our algorithm. We analyze district-level election data following the 2010 redistricting cycle, using 2012 as our baseline for exploring how initial percentages in 2010 influence shifts in voter outcomes over subsequent election cycles.

2.1.1 Data Sourcing

We sourced election data from the 2012, 2016, and 2020 presidential elections. While the focus is on modeling congressional elections, these elections are more unpredictable data quality — i.e. in some congressional districts, candidates from one party may run unopposed, which may make it more difficult to draw conclusions for a more generalized congressional race that we are modeling. Thus, we use presidential election data as a proxy for congressional votes.

We collected the data using a web scraping Python script to extract voting information from each state's election results Wikipedia page. For each election year, we retrieved vote percentages for both major parties in all 435 Congressional districts, and merged this dataset with the Cook Partisan Voting Index (PVI) scores to create a comprehensive dataset for analysis. The Cook Partisan Voting Index (PVI) is a measure used to classify congressional districts based on their relative political leanings compared to the national average. For the 2010 redistricting cycle, the PVI was calculated using voting data from the 2008 presidential election — which we used as a predictive baseline for voter shares in newly drawn districts.

2.1.2 Fitting Logistic Regressions

To estimate the relationship between voting share within a district and the probability of a certain party winning, we fit logistic curves for the election years 2012, 2016, and 2020. The logistic regression model is given by:

$$P(Y = 1 \mid x) = \frac{1}{1 + \exp\left(-\left(w_0 + w_1 x\right)\right)},$$

where x represents the Cook PVI, w_0 is the intercept, and w_1 is the coefficient learned from the data.

To account for yearly swings in voting data we normalized the data. We do this because from 2012, to 2016, 2020, each district systemically moved in one direction or the other based on the presidential candidate, so to simply isolate the variance over time, we remove general systemic changes from the data. Specifically, we calculated the movement from the expected baseline:

$$\text{Movement}_t = \text{Vote Share}_t - \left(0.5 + \frac{\text{Cook PVI}}{100}\right),$$

where $t \in \{2012, 2016, 2020\}$. The mean movement for each year was subtracted from the raw vote shares to adjust for nationwide shifts:

$$\text{Adjusted Vote Share}_t = \text{Vote Share}_t - \text{Mean Movement}_t.$$

This adjustment ensures that the logistic regression models reflect local dynamics rather than national trends.

The logistic regression curves for 2012, 2016, and 2020 in **Figure 1** show the probability of a party's win as a function of voting share within that district, adjusted for the movement calculations/normalization. Over time, as evidenced by the progression of logistic regression curves, the predictive power of initial vote share on electoral outcomes diminishes. The sharpness of the 2012 curve reflects the immediate impact of redistricting decisions made in 2010. At that time, a small shift in voter share near the 50% threshold resulted in a significant change in the likelihood of a party's victory. As time progresses (2016 and 2020), the logistic curves flatten slightly, which suggests that initial vote share loses its predictive accuracy. As such, while redistricting efforts may optimize voter shares for the immediate future, the declining predictive power of voting shares may bias optimal redistricting efforts toward favoring solidly red/blue districts over narrower districts.

Figure 2 shows derived distributions of party vote share for the 2012, 2016, and 2020 elections across selected Cook PVI values (-15, -10, -5, 0, 5, 10). Note the increased variance of outcome over time (through the 2012, 2016, 2020 elections).

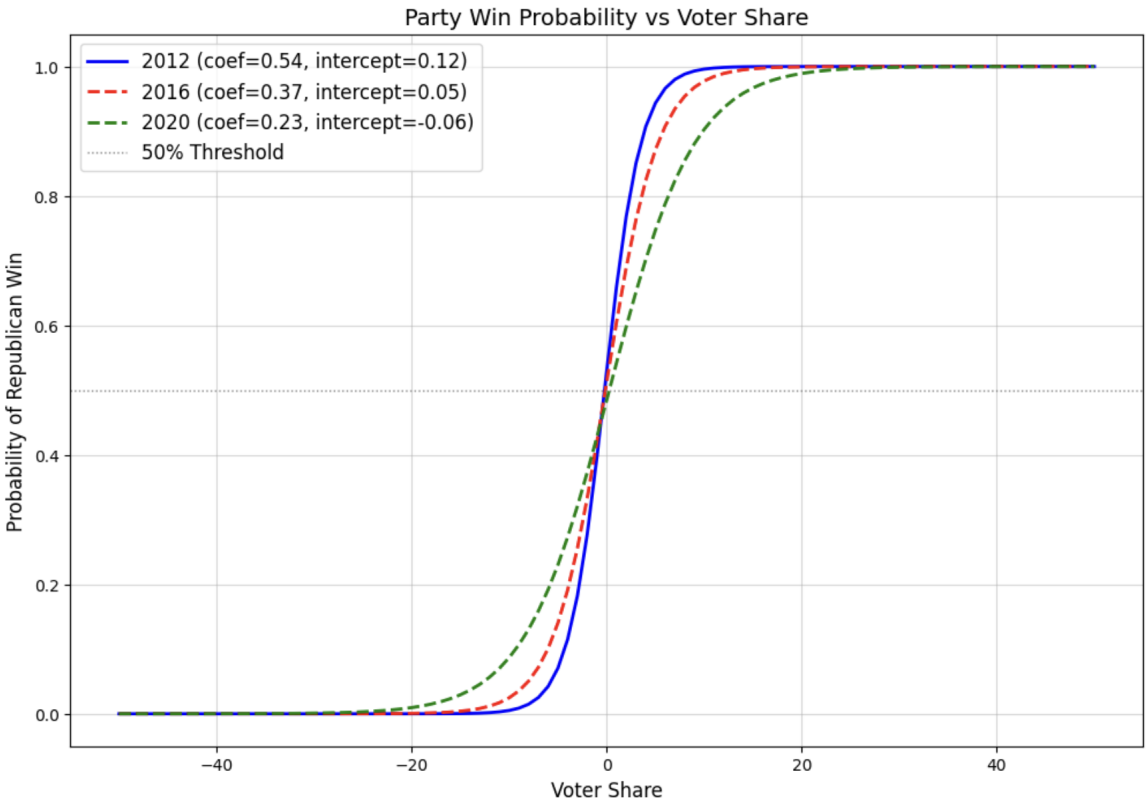

Figure 1: Logistic Regression Curves for 2012, 2016, and 2020, Adjusted for Movement.

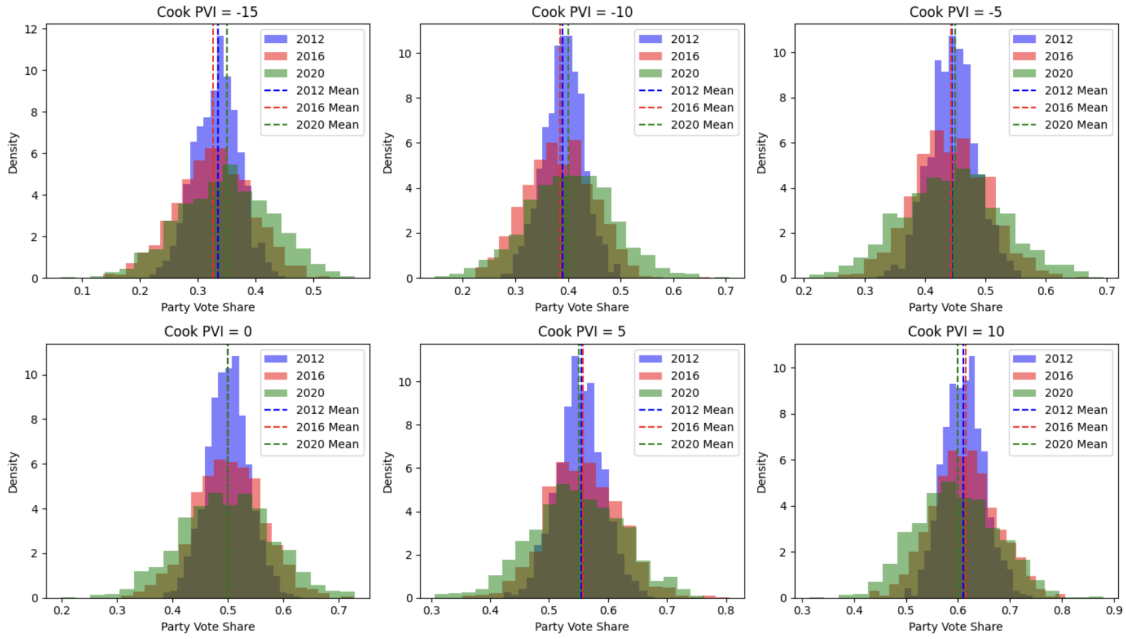

Figure 2: Density of Outcome Based on Party Vote Share (based on Cook PVI)

2.2 Election Maps

The dataset utilized in this study was sourced from the Harvard Dataverse (Voting and Team, 2018), a publicly accessible data repository. Specifically, we extracted precinct-level voting results from the 2016 U.S. general election, which included information on votes cast per precinct (voting district). For the purpose of this analysis, we focused on data from Florida and Wisconsin, two key swing states in the election, to evaluate the effectiveness of our algorithm. These states were selected due to their significance in shaping electoral outcomes and their variability in voting patterns.

3 Proposed Algorithm

Broad Overview of Algorithm

1. Start with an initial partition $\mathcal{D} = \{D_1, D_2, \ldots, D_n\}$, where each district D_k is a subset of the graph G. We employ a Voronoi Tessellation with a bias toward population in the initial partition.
2. Randomly select a boundary node v that lies on the border between two districts D_i and D_j.
3. Reassign the boundary node v from its current district D_i to an adjacent district D_j, with a strict condition that the reassignment maintains the contiguity of D_i.
4. Compute the impact of the reassignment on the overall objective function, and determine whether to accept the reassignment. If it improves the objective function, accept the district flip, but if it doesn't, accept it with a probability based on the Metropolis-Hastings acceptance criterion.
5. Repeat the process until convergence or desired expected number of districts achieved.

3.1 Initializing Districts through Voronoi Tessellation

The algorithm begins by initializing a set number of districts within the study area using a population-biased Voronoi tessellation approach. This process ensures that districts are distributed in a manner that reflects the underlying population density, allowing for a fairer representation.

The input data is first rasterized to create a population density map at the pixel level. Using geospatial data from a state's precincts, we normalized vote counts (Democratic and Republican) by precinct area to calculate vote densities. These densities were rasterized into a uniform grid, with adjustments made to reflect population per pixel by multiplying the densities by pixel area. Pixels outside the state's boundaries were masked to ensure the population map exclusively lied within state boundaries.

To generate the districts, we used Voronoi tessellation. A Voronoi tessellation partitions a space into regions based on a set of seed points. Mathematically, given k seed points $S = \{s_1, s_2, \ldots, s_k\}$

in a two-dimensional space, the Voronoi region V_i for a seed s_i is defined as:

$$V_i = \{x \in \mathbb{R}^2 \mid \|x - s_i\| \leq \|x - s_j\|, \forall j \neq i\}$$

where $\|x - s_i\|$ represents the Euclidean distance between a point x in the space and the seed point s_i. Each region V_i contains all points closer to s_i than to any other seed s_j.

Pixels with valid population values were identified, and their population densities were scaled by a bias factor to increase the likelihood of selecting highly populated areas as district seed points. Using a probabilistic sampling method, district seed points were chosen, and Voronoi tessellation was applied to assign each pixel to its nearest seed point, resulting in initial district boundaries.

The algorithm also calculated the target population per district as:

$$P_{\text{target}} = \frac{P_{\text{total}}}{N}$$

where P_{total} is the total population within the state of interest and N is the number of desired districts. This target informed subsequent refinement steps to balance populations across districts. The resulting map and accompanying data provided a foundation for further analysis of district characteristics and algorithm performance.

Figure 3 shows a generated map of Florida with 27 districts created through the population-biased Voronoi Tessellation. Black pixels represent pixels either outside of state boundaries, or with no population.

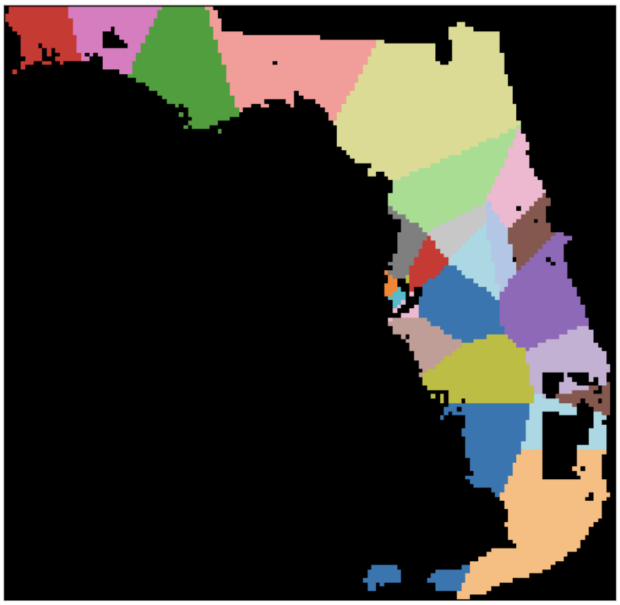

Figure 3: Population-biased Voronoi Tessellation applied on Florida

3.2 Objective Function

Objective Function

We aim to minimize the objective function, incorporating three components:

$$f(\xi) = \alpha P(\xi) + \beta C(\xi) + \gamma F(\xi)$$

where:

- Population Equalization:

$$P(\xi) = \sum_{i=1}^{n} \max\left(0, \left|\frac{\text{pop}(V_i) - \text{Total Population}/n}{\text{Total Population}/n}\right| - 0.05\right)^2$$

- Compactness Constraint:

$$C(\xi) = \sum_{i=1}^{n} \frac{\text{Perimeter}(V_i)}{\sqrt{\text{Area}(V_i)}}$$

- Expected Number of Districts Won:

$$F(\xi) = N - \left(\sum_{i=1}^{n} \frac{1}{1 + e^{-(\beta_0 + \beta_1 V_i(X))}} - \text{Variance Penalty}\right)$$

Population Balance:

$$P(\xi) = \sum_{i=1}^{n} \max\left(0, \left|\frac{\text{pop}(V_i) - \text{Total Population}/n}{\text{Total Population}/n}\right| - 0.05\right)^2$$

The term enforces population equality across districts in the map, which is a required condition since a district should have an equal population.

The inclusion of the 0.05 threshold in the formula serves to allow for a small, acceptable deviation (5%) from the target population before penalization begins. Without the 0.05 relaxation, even minor, insignificant changes population caused algorithm to overly fixate on population balance at the expense of biasing a district plan toward one party. The 0.05 threshold ensures that small deviations (e.g., caused by boundary node flips) do not dominate the optimization process if the populations are roughly equal.

Compactness:

$$C(\xi) = \sum_{i=1}^{n} \frac{\text{Perimeter}(V_i)}{\sqrt{\text{Area}(V_i)}}$$

This discourages districts with high perimeter relative to their area, which are typically less compact and often indicative of gerrymandering. Although our goal is to gerrymander, we do want to impose some compactness constraint to give some structure to the districts.

Political Favorability.

As explained above, most algorithms in the literature do not incorporate a measure of risk/expected reward into their models, but rather use a more broad stroke approach that either uses the algorithm to validate whether a map is gerrymandered or not, or leaves the objective distribution of district control uncertain. However, we endevour to construct this parameter with the goal of optimizing for a specific political party.

From our regressions above, we fit three logistic regression models to historical election data for the years 2012, 2016, and 2020. The logistic function models the probability of a party's win in a district as a function of the initial voter share in that district. The logistic function is expressed as:

$$P(\text{win}_i) = \frac{1}{1 + e^{-\beta_0 - \beta_1 V_i(0)}},$$

where $P(\text{win}_i)$ represents the probability of winning district i, β_0 and β_1 are the fitted logistic regression parameters, $V_i(0)$ represents the initial voter share in district i.

The fitted coefficients for each election year are summarized in the table below:

Election Year	β_0	β_1
2012 ($t = 2$)	$\beta_{0,2} = 0.118$	$\beta_{1,2} = 0.54105964$
2016 ($t = 6$)	$\beta_{0,6} = 0.051$	$\beta_{1,6} = 0.37283259$
2020 ($t = 10$)	$\beta_{0,10} = -0.0619$	$\beta_{1,10} = 0.229$

Table 1: Fitted Logistic Regression Coefficients

Using the logistic regression models, we calculate the expected number of districts won by summing the probabilities of winning each district across the entire map. The expected number of districts won is defined as:

$$\mathbb{E}[D_{\text{won}}] = \sum_{i=1}^{n} P(\text{win}_i),$$

The variance of the win probability for a single district i at time t is given by:

$$\text{Var}[\text{win}_i(t)] = P(\text{win}_i(t)) \cdot (1 - P(\text{win}_i(t))).$$

The total variance of district outcomes is computed as:

$$\text{Var}[D_{\text{won}}] = \sum_{i=1}^{n} \text{Var}[\text{win}_i].$$

The political favorability term incorporates the expected number of districts won and the variance across districts. For a map ξ, the political favorability is defined as:

$$F(\xi) = N - (\mathbb{E}[D_{\text{won}}] - \lambda \cdot \text{Var}[D_{\text{won}}]),$$

where λ is a weight controlling the trade-off between expected outcomes and variance.

3.2.1 Variance Penalty

In our objective function, the term controlling for political favorability includes an embedded variance penalty.

There are multiple ways to implement a variance penalty. One comprehensive approach combines short-term and long-term probabilities of winning a distric by taking an average of the distributions at $t = 2$, $t = 6$, and $t = 10$, and this can be written as:

$$P_{\text{average}}(\text{win}_i) = \frac{1}{3}\left(P_{\text{short}}(\text{win}_i) + P(\text{win}_i(t = 6)) + P(\text{win}_i(t = 10))\right),$$

where $P(\text{win}_i(t = 6)), P(\text{win}_i(t = 10))$ are probabilities derived with logistic functions found above. We found that in practice, this can be approximated by relaxing the logistic curve's steepness by dividing the exponent of e by a parameter T, we obtain:

$$P(\text{win}_i) = \frac{1}{1 + e^{-\frac{\beta_0 + \beta_1 V_i(0)}{T}}},$$

where T acts as a "time scaling factor." Larger values of T produce flatter logistic curves, penalizing variance over a longer time period.

Accepting or Rejecting the New Map

Recall the objective function:

$$f(\xi) = \alpha P(\xi) + \beta C(\xi) + \gamma F(\xi)$$

For each iteration, we randomly select a node v on the boundary of a district. We then consider moving v from district i to an adjacent district j. We then calculate the change in objective function:

$$\Delta f = f(\xi') - f(\xi)$$

We calculate the acceptance probability of the proposal using the Metropolis-Hastings criterion:

$$P(\text{accept}) = \begin{cases} 1 & \text{if } \Delta f \leq 0, \\ e^{-\Delta f / T} & \text{if } \Delta f > 0. \\ 0 & \text{if the flip disconnects the district} \end{cases}$$

With probability $P(\text{accept})$, set $\xi(v) = j$. We change the temperature using the following approach:

$$T_k = T_0 \times \alpha^k$$

where T_0 is the initial temperature, $\alpha \in (0, 1)$ is the cooling rate, and k is the current iteration.

Refined Cooling Schedule

Initially, we experimented with a high initial temperature ($T_0 = 1000$) and a decay rate of $\alpha = 0.995$, which reduced the temperature exponentially with each iteration. However, we observed that this schedule decayed the temperature too quickly, causing the algorithm to converge prematurely to suboptimal solutions. Later attempts to reduce α (slower cooling) did allow the initial iterations to be more variable but did not consistently find better solutions in later iterations.

To address this, we devised a dynamic cooling schedule that leverages periodic resets of the temperature to inject variability into the search space:

1. Start with $T_0 = 1000$ and apply a cooling factor of $\alpha = 0.995$ at each iteration:

$$T_k = T_0 \times \alpha^k.$$

2. Once the temperature reaches a threshold of $T_k < 10^{-8}$, reset $T_k = 400$. The cooling process then resumes with the same factor $\alpha = 0.995$.

This periodic temperature reset allows for the algorithm to escape local minima by reintroducing the possibility of accepting "worse" moves that may lead to better solutions in subsequent iterations.

4 Results and Discussion

4.1 Florida, favoring Republicans

Figure 4 displays the redistricted map of Florida after 130,000 iterations. Districts which are colored in darker red are more likely to be won by Republicans, while darker blues are more likely to be won by Democrats.

First, note that the bar chart illustrating the final population per district indicates a relatively even distribution, with most districts containing approximately 300,000 voters. In our algorithm, we specified a target population per district of approximately 350,000, but allowed the districts to deviate by around 5–7.5% (approximately 17,000 to 26,000 voters) in either direction. This tolerance incentivized the districts to optimize for political favorability rather than bottlenecking the optimization process in achieving strict population equalization. Although some districts exhibited notably larger populations compared to others (with differences of approximately 50,000

voters), this range aligns with real-world population imbalances seen in congressional districts, where population differences often exceed 100,000 voters in some cases.

Moreover, the favorability chart reveals a polarized distribution, with most districts exhibiting probabilities near 0 (Democratic-majority districts) or 1 (Republican-majority districts). This binary favorability outcome reflects the strong partisan alignment within most districts. Specifically, 19 districts exhibit a Republican vote share exceeding 50%, securing their status as Republican-majority districts (with win probabilities close to 100%). The remaining districts fall below this threshold, corresponding to Democratic-majority districts. Notably, very few districts are near the 50% threshold, as the algorithm naturally penalized swing districts and favored more polarized outcomes, as evidenced by the favorability distribution.

In terms of optimization dynamics, the total loss is initially dominated by population loss, as districts vary greatly in size in early iterations. As shown in the loss plot, the total loss (purple line) rapidly declines, driven primarily by a steep reduction in population loss as the algorithm equalizes population across districts. The total loss plateaus in later iterations, indicating convergence to a solution that balances the three objectives within the constraints of the simulated annealing framework. Sudden increases in total loss are observed periodically due to temperature resets (e.g., to $T = 400$), which temporarily allow the algorithm to escape local minima and explore alternative configurations. Additionally, the loss plot only tracks accepted iterations (i.e., iterations accepted by Metropolis-Hastings criterion), which explains the discrepancy between the number of iterations on the loss plot and the map's reported iteration number.

Finally, the chart depicting the expected number of districts won by Republicans highlights the algorithm's success in optimizing toward the target of 18 Republican-majority districts. Under the 2012 and 2016 voting patterns, the expected number of Republican-majority districts converges near the target, stabilizing at approximately 18.5 and 17.8 districts, respectively. However, under the 2020 voting pattern, the number stabilizes slightly lower, at approximately 17 districts. This divergence reflects increased Democratic competitiveness in 2020, likely influenced by demographic changes, urbanization, and higher Democratic turnout in key areas. Furthermore, the widening confidence interval for 2012 suggests greater variability in Republican dominance under the 2012 voting pattern, potentially driven by more volatile voter behavior or stronger Republican turnout in certain regions. The algorithm's robustness is demonstrated by its ability to approach the target of 18 Republican-majority districts across varying electoral conditions. However, the inability to fully meet the target for 2020 underscores the influence of shifting voter behavior and demographic distributions, which cannot be entirely mitigated through redistricting. These results emphasize the importance of incorporating evolving electoral trends and uncertainty into districting strategies.

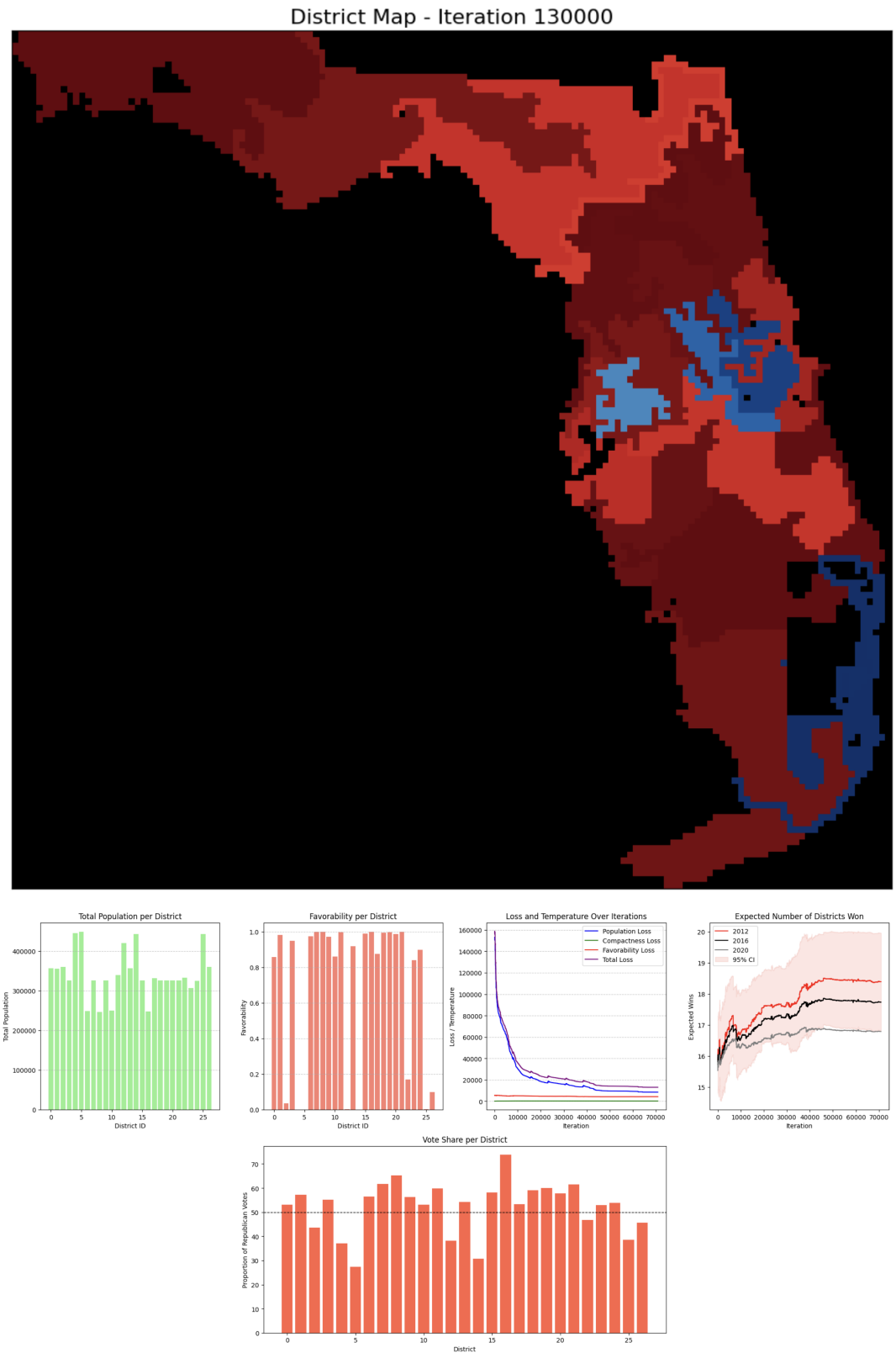

Figure 4: Redistricting Results for Florida, favoring Republicans

4.2 Wisconsin, favoring Democrats

The districting map in **Figure 5** demonstrates the results of an optimization process targeting eight districts (since Wisconsin had 8 congressional districts drawn in 2010). The optimization objective has also been adjusted to favor Democrats, as indicated by the negative favorability loss parameter ($\gamma < 0$). Consequently, the algorithm aims to minimize the expected number of Republican-majority districts, as reflected in the "Expected Number of Districts Won" plot, where the expected Republican wins decrease over iterations.

The map shows that the Democratic-majority districts (blue) are concentrated in the southern region, likely corresponding to urban areas with historically strong Democratic support. Republican-majority districts (red) dominate the northern and rural areas, consistent with broader voting trends. The total population per district is evenly distributed around the 350,000 target, with only minor deviations, demonstrating the algorithm's ability to maintain population parity while optimizing for partisan favorability.

The favorability chart reveals a strong polarization, with four districts exhibiting a Democratic majority (probabilities near 0) and four districts showing a Republican majority (probabilities near 1). This outcome reflects the algorithm's success in aligning with the objective of reducing Republican dominance. The vote share chart supports this, with the proportion of Republican votes exceeding 50% in only four districts (instead of the 6 districts won in 2016).

The favorability loss, being negative due to $\gamma < 0$, drives the total loss lower as the algorithm minimizes Republican advantages. The plateauing of the loss values indicates convergence toward a stable solution that balances all objectives.

The "Expected Number of Districts Won" plot highlights the algorithm's success in reducing Republican dominance, with expected Republican wins stabilizing at 4.1 districts. The sharp early decline reflects significant progress in flipping marginal districts to Democratic majorities. This adjustment to favor Democrats demonstrates the flexibility of the optimization framework to achieve specific partisan objectives by modifying the loss parameters.

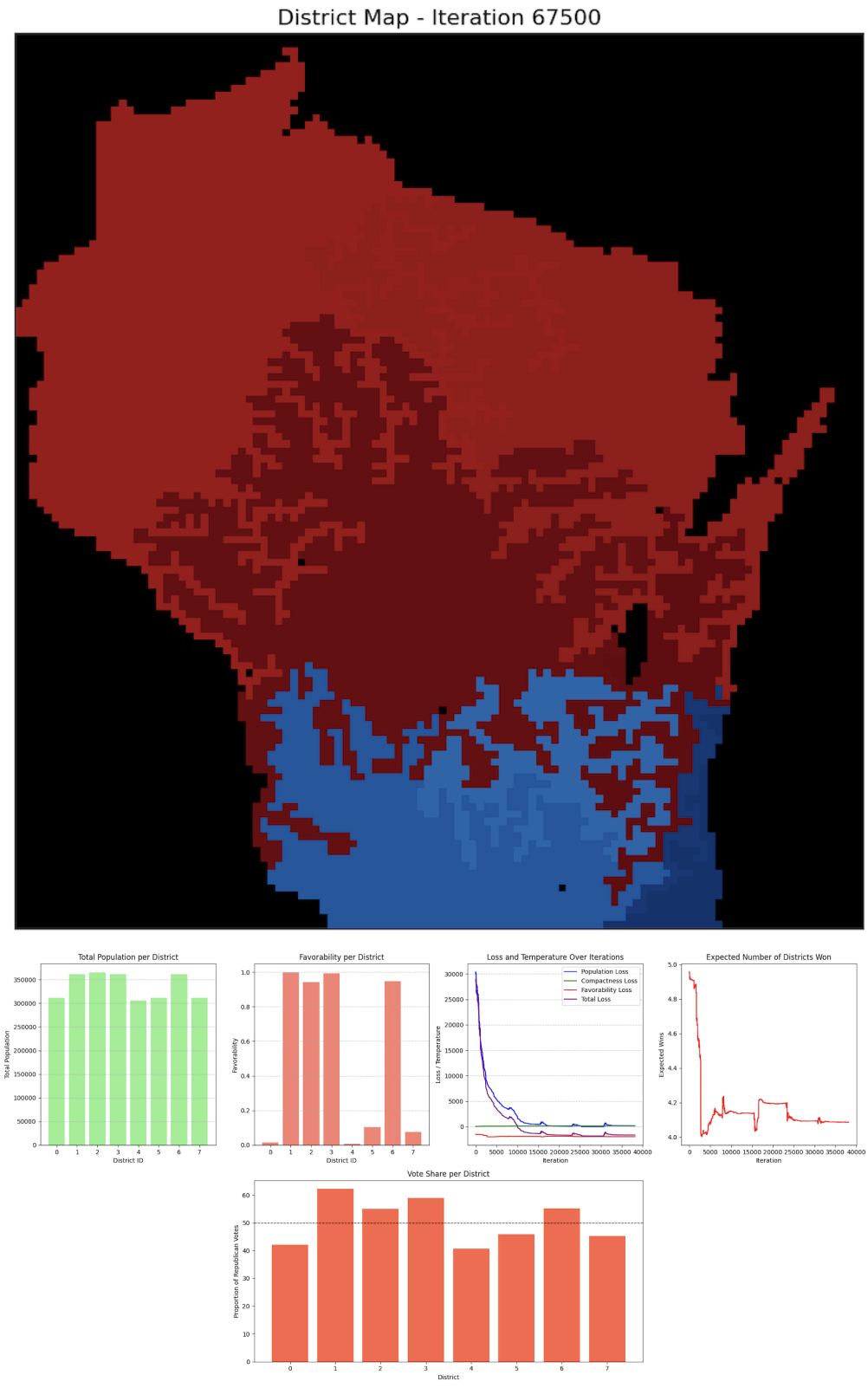

Figure 5: Redistricting Results for Wisconsin, favoring Democrats

4.3 Wisconsin, favoring Republicans

The districting map in **Figure 6** represents the results of an optimization process targeting eight districts, with the objective of favoring Republican outcomes, as indicated by the positive favorability loss parameter ($\gamma > 0$). Unlike the prior scenario, the algorithm aims to maximize the expected number of Republican-majority districts. The "Expected Number of Districts Won" plot shows a steady increase in expected Republican wins, stabilizing near 5.5 districts.

The map illustrates a clear partisan division, with Democratic-majority districts (blue) concentrated in the southeastern region, likely urban centers, while Republican-majority districts (red) dominate the rural and northern areas. The population per district, as shown in the bar chart, is relatively balanced, with most districts near the target population of 350,000. The algorithm allows minor deviations to prioritize partisan favorability while maintaining population parity.

The favorability chart reveals a relatively polarized distribution, with five districts showing Republican-majority probabilities close to 1 and three districts exhibiting Democratic-majority probabilities near 0. The vote share per district confirms this alignment, as Republican votes exceed 50% in five districts. Few districts hover near the 50% threshold, indicating that the algorithm penalizes competitive districts, favoring decisive partisan outcomes.

The loss dynamics demonstrate the interplay between the three optimization objectives. Initially, the total loss (purple line) is dominated by population loss, as districts vary greatly in size. As iterations progress, population loss decreases rapidly as district sizes equalize. Compactness and favorability losses decline more gradually, reflecting incremental improvements in district shapes and partisan alignment. The plateau in total loss indicates convergence to a stable solution, while temperature resets (visible as minor fluctuations in the loss curve) allow the algorithm to escape local minima.

Finally, the "Expected Number of Districts Won" plot highlights the algorithm's success in optimizing Republican outcomes. The expected Republican-majority districts increase steadily, stabilizing at approximately 5.5 by the final iterations. This steady improvement reflects the algorithm's effectiveness in redistributing voters to align district outcomes with the partisan objective. The results underscore the flexibility of the optimization framework in achieving specific political goals while balancing population and compactness constraints.

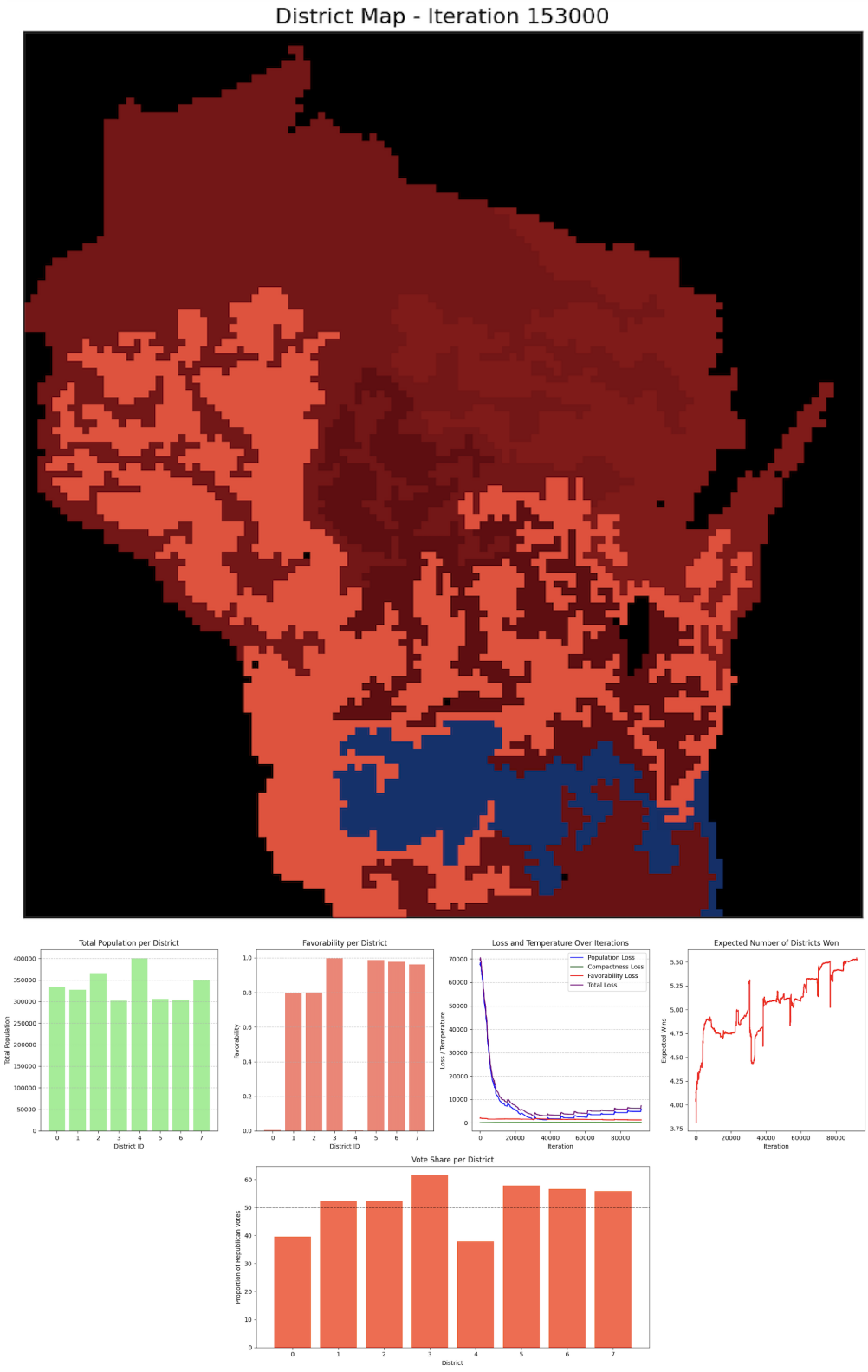

Figure 6: Redistricting Results for Wisconsin, favoring Republicans

5 Conclusion and Future Work

The results from our case studies in Florida and Wisconsin demonstrate the flexibility and effectiveness of the proposed framework. In Florida, our algorithm successfully optimized districts to favor Republican outcomes, exceeding a targeted number of Republican-majority districts while balancing compactness and population parity. Similarly, the Wisconsin case studies illustrate how the algorithm can be adapted to favor either Democratic or Republican objectives.

Notably, the inclusion of variance penalties allowed our algorithm to product district plans more resilient to demographic and electoral changes. This approach addresses a gap in the literature by prioritizing both immediate and sustained partisan advantages.

Future work could explore refining the variance penalty and cooling schedule mechanisms to further improve convergence efficiency and solution quality. Additionally, incorporating demographic and geospatial changes into the optimization framework could further enhance the robustness of the model.

Moreover, the current compactness score, which penalizes high perimeter-to-area ratios, may not adequately constrain the irregularity of district boundaries, as seen in the presented maps. Implementing an alternative compactness measure—such as Reock or Polsby-Popper scores—could prioritize the creation of less clearly-gerrymandered, more visually and geometrically balanced districts.

6 Bibliography

References

Benjamin Fifield, K. I., Michael Higgins, & Tarr, A. (2020). Automated redistricting simulation using markov chain monte carlo. *Journal of Computational and Graphical Statistics*, *29*(4), 715–728. https://doi.org/10.1080/10618600.2020.1739532

McCartan, C., & Imai, K. (2023). Sequential Monte Carlo for sampling balanced and compact redistricting plans. *The Annals of Applied Statistics*, *17*(4), 3300–3323. https://doi.org/10.1214/23-AOAS1763

Voting & Team, E. S. (2018). *2016 Precinct-Level Election Results*. https://doi.org/10.7910/DVN/NH5S2I