Incorporating Risk into Algorithmic Redistricting

Assessing and Optimizing Redistricting Algorithms with a Bias-Variance Trade-Off

Ivan Sinyavin and Owen Xu Li

December 12, 2024

Introduction

- Redistricting redraws electoral district boundaries, typically after each decennial census.
- Gerrymandering manipulates boundaries to benefit a particular party by:
  - **Packing**: Concentrating opposition voters in a few districts.
  - **Cracking**: Diluting opposition voters across many districts.
- Our goal: Develop an algorithm that explicitly optimizes for a specific party's objectives using:
  - Historical voting data.
  - Measures of expectation and variance.
  - Optimization for short-term and long-term political success.

Past Voting Behavior in Congressional Elections

- **Dataset:**
  - 2012, 2016, and 2020 presidential elections used as proxies for congressional voting.
- **PVI Baseline:**
  - Calculated using 2008 presidential election results for the 2010 redistricting cycle.
  - Serves as a predictive baseline for newly drawn districts post-2010 census.
- **Temporal Shifts:**
  - Changes in voting patterns from 2012 to 2020 reflect temporal shifts in political control post-redistricting, how initial voter distributions ($t = 0$) evolve,
    - Sharp impact of redistricting in immediate elections (e.g., 2012).
    - Diminishing influence of initial redistricting plans as partisan compositions shift over time (e.g., by 2020).

Fitting Logistic Regressions (Part 1)

- Adjusted vote shares to isolate local dynamics by removing nationwide shifts:

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

- Mean movement removed to account for national trends:

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

Fitted Coefficients:

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: Fitted Logistic Regression Coefficients.

Fitting Logistic Regressions (Part 2)

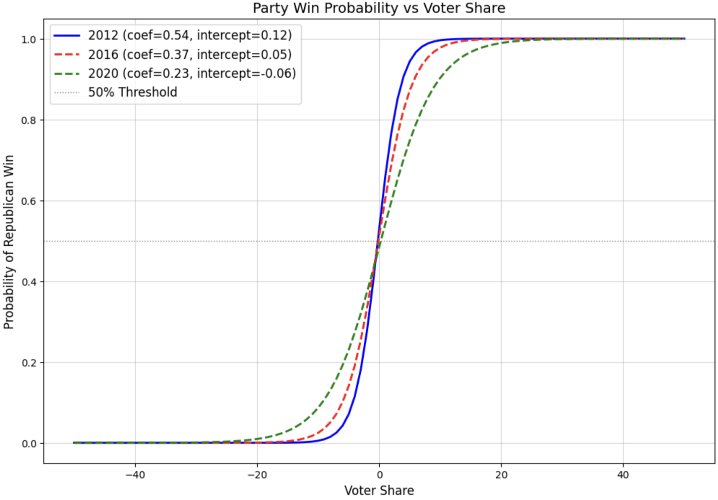


Figure: Logistic Regression Curves for 2012, 2016, and 2020 (Adjusted).

Broad Overview of Algorithm

Algorithm Steps

1. **Initial Partition:** 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. Use Voronoi Tessellation with a bias toward population.
2. **Boundary Node Selection:** Randomly select a boundary node v that lies on the border between two districts D_i and D_j.
3. **Reassignment:** Reassign v from its current district D_i to an adjacent district D_j.
4. **Evaluation:** Compute the impact of the reassignment on the overall objective function. Accept the reassignment if:
    - It improves the objective function, or
    - It satisfies the Metropolis-Hastings acceptance criterion.

Voronoi Tessellation for Initial Districts

- Tessellation:
  - Partitions a space into regions based on a set of seed points.
  - Each region contains points closer to its seed than any other seed.

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

  where V_i is the region around seed s_i.
- Process:
  - Use population density to bias the selection of seed points.
  - Apply tessellation to assign each pixel to its closest seed.
- Target Population:

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

  where P_{total} is the total population and N is the number of districts.

Voronoi Tessellation Map of Florida

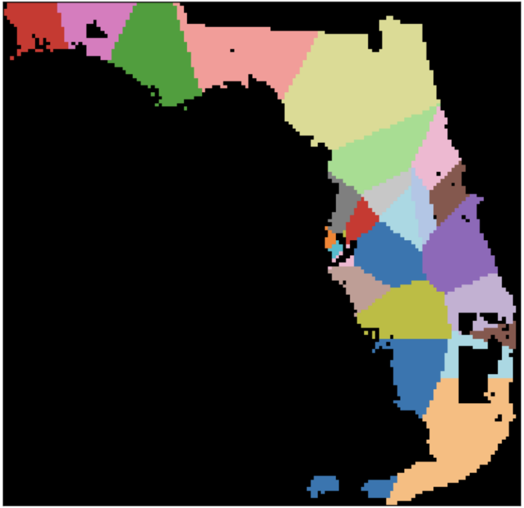


Figure: Population-biased Voronoi Tessellation applied to Florida.

Objective Function

Objective Function

The objective function is defined as:

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

where:

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

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

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

Political Favorability: Overview

- Most existing algorithms do not explicitly optimize for political favorability or incorporate measures of risk/reward. We construct a parameter explicitly optimizing for a specific political party.

$$\mathbb{E}[D_{\text{won}}] = \sum_{i=1}^{n} P(\text{win}_i), \quad \text{Var}[D_{\text{won}}] = \sum_{i=1}^{n} .P(\text{win}_i(t)) \cdot (1 - P(\text{win}_i(t)))$$

- Final Political Favorability Term:

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

- Variance penalty approximated via logistic curve adjustment:

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

Larger T: flatter curve, penalizing variance over time.

Accepting or Rejecting the New Map

- For each iteration:
  - Randomly select a boundary node v.
  - Consider moving v from district i to adjacent district j.
  - Compute change in objective function:

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

- Acceptance Probability:

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

- Update temperature:

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

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

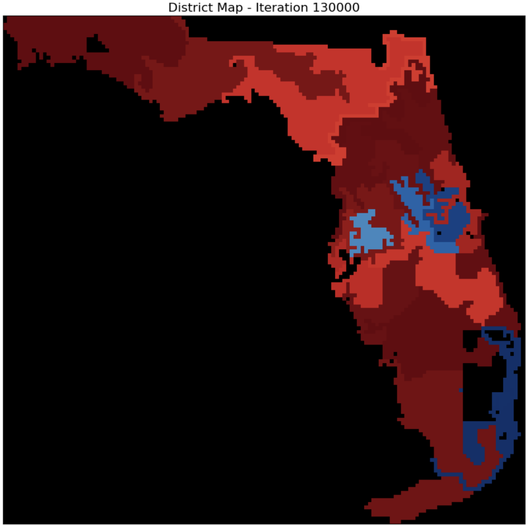
Refined Cooling Schedule

- Initial approach:
  - $T_0 = 1000$, cooling factor $\alpha = 0.995$.
  - Observed premature convergence due to rapid decay of temperature.
- Dynamic Cooling Schedule:
  1. Start with $T_0 = 1000$ and cool using:

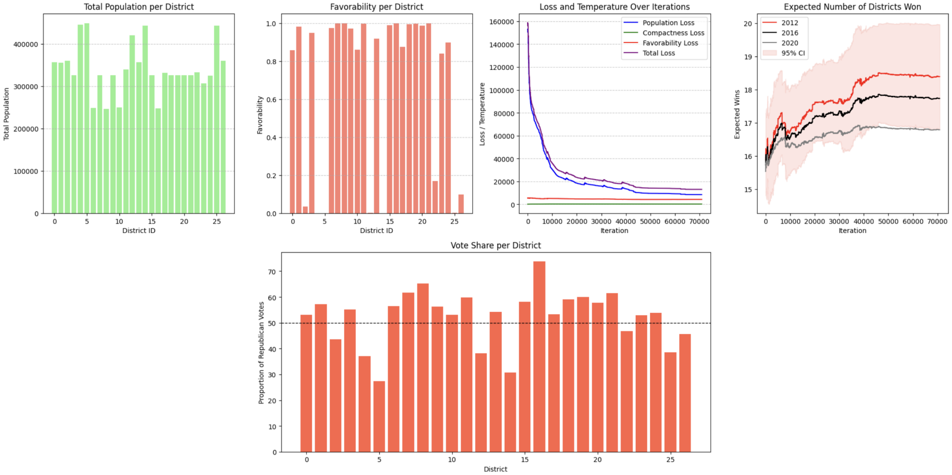
$$T_k = T_0 \times \alpha^k.$$

  2. If $T_k < 10^{-8}$, reset $T_k = 400$.
  3. Resume cooling with the same factor $\alpha = 0.995$.
- Pros of Dynamic Cooling:
  - Prevents premature convergence to suboptimal solutions.
  - Introduces variability by accepting "worse" moves.

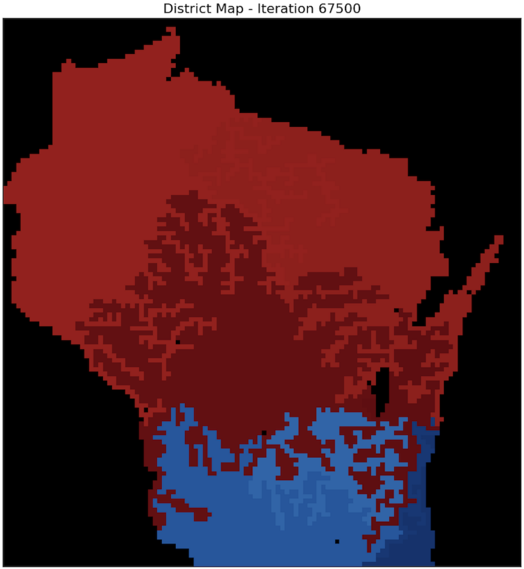
Redistricting Results for Florida - Republican



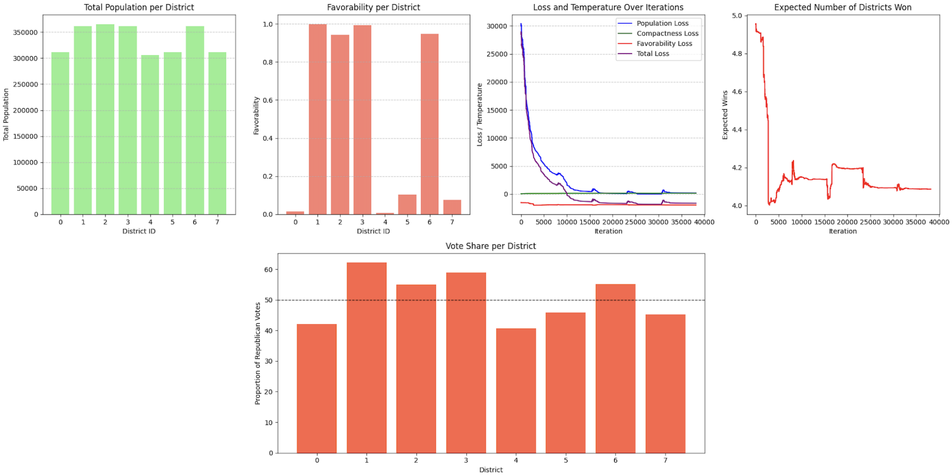
Redistricting Results for Florida - Republican



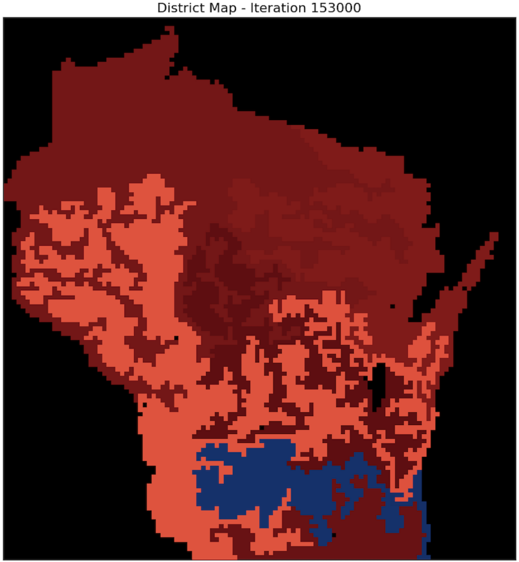
Redistricting Results for Wisconsin - Democratic



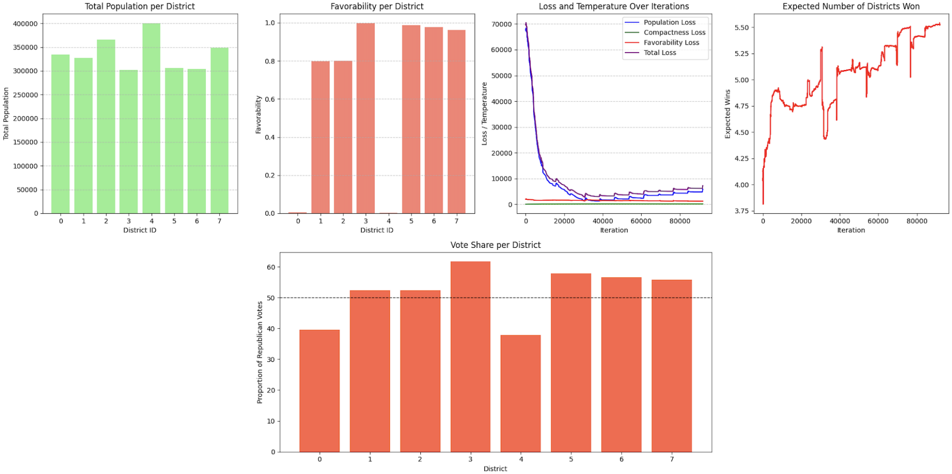
Redistricting Results for Wisconsin - Democratic



Redistricting Results for Wisconsin - Republican



Redistricting Results for Wisconsin - Republican



Conclusion and Future Work

- We introduced a different approach to redistricting by integrating partisan objectives, population equality, and compactness.
- Our algorithm uses historical voting data measures of expected risk/reward.
  - Case studies in Florida and Wisconsin, optimizing districts for partisan objectives while maintaining constraints.
  - Incorporates variance penalties and long-term favorability, which allows the algorithm to optimize against demographic and electoral changes.
- Future work includes refining variance penalties, compactness measures (e.g., Reock or Polsby-Popper), and integrating demographic/geospatial dynamics.