Drawing Fair Congressional Districts with Mixed-Integer Programming

Nicholas Beasley and Christopher En

April 2019

Abstract

We develop a model that uses mixed-integer programming to draw Congressional district maps. States are broken into discrete pieces and assigned to districts in a way that satisfies legal requirements and minimizes gerrymandering metrics. Our method is effective at drawing maps that are fair, but we find that enforcing contiguity is a major barrier to getting realistic results.

1 Introduction

In most US states, state legislatures are responsible for drawing Congressional districts after each census [1]. This allows the political party in control of the legislature to draw maps that allow it to win a disproportionate amount of seats, a practice referred to as gerrymandering. This is often done by drawing districts that pack many of the other party's voters into uncompetitive races. For example, in the 2018 midterm elections in North Carolina, Republicans received 50.4% of the vote but won 9 out of 13 Congressional districts [2].

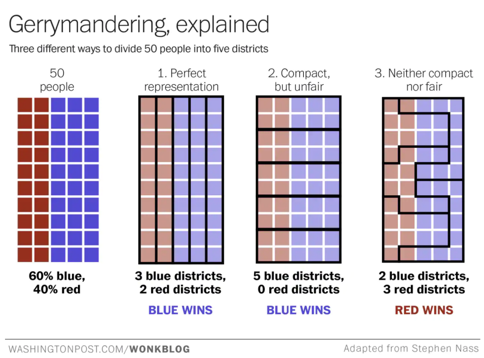

Figure 1: A simplified example of gerrymandering [3]

However, in recent years, there has been a movement to put the redistricting process in the hands of independent commissions. This practice has even been implemented in the most populous state in the US, California [1,4]. While this represents progress in combating gerrymandering, the process through which these commissions draw maps is tedious and still susceptible to partisan influence. When drawing new maps in 2011, the Redistricting Commission in California held 34 public hearings and received 20,000 written comments, including 2,000 potential maps [4]. Because of this, the Commission was not able to issue a final report until August, despite being formed in January.

To reduce the potential for partisan influence and the amount of time needed for the district-drawing process, we formulate it in a way that can be solved with computers. By discretizing states into small pieces (towns in the case of Massachusetts), we can write a linear program that assigns these pieces to different districts in a way that satisfies legal requirements (such as having roughly equal populations in each district) while optimizing some metric of gerrymandering. Solving the resulting LP should return an allocation of towns to districts representing a fair Congressional map.

1.1 Gerrymandering Metrics

Due to the prevalence of gerrymandering, political scientists have devised several metrics for measuring it. We found that the main voter-based metrics were **partisan bias** and **efficiency gap**. The partisan bias is the divergence in the number of seats between parties in a hypothetical election where each party got 50% of the vote [5]. The efficiency gap is the difference in the number of "wasted votes" for each party, divided by the total number of votes [5,6]. A wasted vote is defined as a vote cast for a losing candidate, or a vote cast for a winning candidate in excess of the 50% needed to win. We mainly work with these metrics in our paper because they are easier to compute and formulate as LPs.

Several geometric measures of gerrymandering also exist. They include the **Polsby-Popper ratio** and the **convex hull ratio**. The Polsby-Popper ratio is defined as $\frac{4\pi A}{P^2}$ (A =area, P =perimeter); it assumes that a circle is the ideal shape for a district (the ratio is 1), and penalizes districts that have high perimeter but low area [7]. On the other hand, the convex hull ratio is $\frac{A}{A_0}$, where A_0 refers to the area of the convex hull of a district [7]. While these measures are reasonable, we decided not to explore them in this paper because of the difficulty of computing the perimeter of districts (as well as computing the area of the convex hull of these districts).

2 Model

2.1 Partisan Bias

Our first model attempts to minimize partisan bias, as well as minimizing the sum of the "diameters" of each district as a tiebreaker between solutions. In describing our model, we denote the set of towns as T and the set of congressional districts as D.

Our parameters are:

- $p_i,\ \forall i \in T$: the population of town i.
- $r_i, d_i,\ \forall i \in T$: the number of Republicans/Democrats in town i in a hypothetical 50-50 race. In our model, this is done by shifting the partisan breakdown of independent voters.
- $c_{ij},\ \forall i, j \in T$: the distance between towns i, j for all pairs of towns.
- We also have big-M and little-m parameters, as is explained below.

Our decision variables are:

- $z_i^j \in \{0, 1\},\ \forall i \in T, j \in D$: whether town i is in district j.
- $dw_j, rw_j \in \{0, 1\},\ \forall j \in D$: whether Democrats/Republicans win (i.e. have more voters in) district j.
- $b \geq 0$: the partisan bias of the state
- $l_j \geq 0,\ \forall j \in D$: the distance between the two farthest towns in district j.

The first constraint is to ensure each town is assigned to exactly one district.

$$\sum_{j \in D} z_i^j = 1, \forall i \in T \tag{1}$$

To force each district to have a roughly equal population, we have

$$\sum_{i \in T} z_i^j p_i \leq maxpop, \forall j \in D \tag{2}$$

$$\sum_{i \in T} z_i^j p_i \geq minpop, \forall j \in D \tag{3}$$

For Massachusetts, we initially let $maxpop = 735,000$ and $minpop = 720,000$. These numbers come from the fact that the total population (according to our data) is 6,547,695 people, for an average of 727,521.67 people per district. To increase the likelihood of a feasible solution, we have forced the population of each district to be within a relatively large interval of 720,000 to 735,000 people.

Next, we have constraints to identify which party won each district.

$$dw_j + rw_j = 1, \forall j \in D \tag{4}$$

$$10^6 dw_j \geq \sum_{i \in T} z_i^j (d_i - r_i), \forall j \in D \tag{5}$$

$$10^6 rw_j \geq \sum_{i \in T} z_i^j (r_i - d_i), \forall j \in D \tag{6}$$

We use 10^6 as a big-M value, as it is greater than the population of any one district. Thus, when dw_j or rw_j are 1, the second and third inequalities will always be satisfied, respectively. The first constraint says that exactly one party wins each district. The second constraint says that if Democrats win the district (by having more voters), then dw_j must equal 1, indicating a Democratic victory. This then forces $rw_j = 0$. Similarly, if Republicans win, then rw_j must equal 1 and $dw_j = 0$. (Note: Ties are extremely unlikely, but they are technically allowed to go either way).

Then, the partisan bias constraint is given by

$$b \geq \sum_{i \in D} dw_i - \sum_{i \in D} rw_i \tag{7}$$

$$b \geq \sum_{i \in D} rw_i - \sum_{i \in D} dw_i \tag{8}$$

The partisan bias must be larger than the difference in the number of seats, whether Democrats win more seats or Republicans win more seats.

We have a large set of constraints to identify the correct value of l_i for each district, given by

$$l_i \geq c_{s,t}(z_s^i + z_t^i - 1), \qquad \forall i \in D,\ s, t \in T \tag{9}$$

If towns s and t are both in district i, then l_i must be greater than or equal to the distance c_{st} between the towns. Otherwise, the constraint is trivially satisfied.

Finally, the objective is given by

$$\min_{z,dw,rw,b,l} b + m \sum_{i \in D} l_i \tag{10}$$

The goal is to first minimize the partisan bias, then to minimize the sum of the "diameters" of each district. Since there are 9 districts in Massachusetts, any optimal solution should have a partisan bias of 1, with one party winning 4 seats and the other winning 5. So, we want to choose a little-m value such that $m \sum_{i \in D} l_i$ is at least an order of magnitude below 1. A valid choice would be $m = 10^{-4}$.

2.2 Efficiency Gap

The second model attempts to minimize the efficiency gap. Our sets are T, the set of towns, and D, the set of congressional districts.

Our parameters are:

- $p_i,\ \forall i \in T$: the population of town i.
- $r_i, d_i,\ \forall i \in T$: the number of Republicans/Democrats in town i.
- $c_{ij},\ \forall i, j \in T$: the distance between towns i, j for all pairs of towns.

Our decision variables are:

- $z_i^j \in \{0,1\},\ \forall i \in T, j \in D$: whether town i is in district j.
- $dw_j, rw_j \in \{0,1\},\ \forall j \in D$: whether Democrats/Republicans win (i.e. have more voters in) district j.
- $dwastedif_j, j \in D$: Democrats' wasted votes minus Republicans' wasted votes in district j.
- $l_j \geq 0,\ \forall j \in D$: the distance between the two farthest towns in district j.
- $x \geq 0$: a variable that will remove an absolute value from our objective.

The first six constraints are exactly the same as in the partisan bias model. Each town must be in exactly one district:

$$\sum_{j \in D} z_i^j = 1, \forall i \in T \tag{11}$$

Since each district must have roughly equal population,

$$\sum_{i \in T} z_i^j p_i \leq maxpop, \forall j \in D \tag{12}$$

$$\sum_{i \in T} z_i^j p_i \geq minpop, \forall j \in D \tag{13}$$

The bounds chosen are identical to the bounds chosen in the partisan bias model.

In addition, we once again want to identify which party won each district.

$$dw_j + rw_j = 1, \forall j \in D \tag{14}$$

$$10^6 dw_j \geq \sum_{i \in T} z_i^j (d_i - r_i), \forall j \in D \tag{15}$$

$$10^6 rw_j \geq \sum_{i \in T} z_i^j (r_i - d_i), \forall j \in D \tag{16}$$

$$\tag{17}$$

We use 10^6 as a big-M value, as in the partisan bias model.

Finally, we need constraints that will ensure that $dwastedif_j$ accurately measures the difference in the number of wasted votes in each district.

$$dwastedif_j \leq \frac{1}{2} \sum_{i \in T} z_i^j (d_i - 3r_i) + 2 \cdot 10^6 rw_j, \forall j \in D \tag{18}$$

$$dwastedif_j \geq \frac{1}{2} \sum_{i \in T} z_i^j (d_i - 3r_i) - 2 \cdot 10^6 rw_j, \forall j \in D \tag{19}$$

$$dwastedif_j \leq \frac{1}{2} \sum_{i \in T} z_i^j (3d_i - r_i) + 2 \cdot 10^6 dw_j, \forall j \in D \tag{20}$$

$$dwastedif_j \geq \frac{1}{2} \sum_{i \in T} z_i^j (3d_i - r_i) - 2 \cdot 10^6 dw_j, \forall j \in D \tag{21}$$

If Democrats win a district, then the latter two constraints imply the difference in wasted votes is in some huge interval, so they are non-binding. On the other hand, the first two constraints imply that:

$$dwastedif_j = \frac{1}{2} \sum_{i \in T} z_i^j (d_j - 3r_j)$$

The RHS is exactly the difference between the number of wasted Democratic votes (half the difference between Dem and Rep votes) and the number of wasted Republican votes (all of the Rep votes).

Similarly, if Democrats lose a district, then the first two constraints imply the difference in wasted votes is in some huge interval, so they are non-binding. On the other hand, the latter two constraints imply that:

$$dwastedif_j = \frac{1}{2}\sum_{i\in T} z_i^j(3d_j - r_j)$$

which is the difference between Dem. and Rep. wasted votes in a Dem. loss.

Our last set of constraints is the same as in the partisan bias model. They make sure l_i is defined correctly:

$$l_i \geq c_{s,t}(z_s^i + z_t^i - 1), \qquad \forall i \in D,\ s,t \in T \tag{22}$$

Finally, recall that the efficiency gap is the absolute value of the sum of $dwastedif_j$ over $j \in D$ (the absolute value of all Dem. wasted votes minus all Rep. wasted votes), divided by the number of voters. To eliminate the absolute value, we add two final constraints:

$$x \geq \sum_{j\in D} dwastedif_j \tag{23}$$

$$x \geq -\sum_{j\in D} dwastedif_j \tag{24}$$

Then our objective is:

$$\min \frac{x}{\sum_{i\in T}(r_i + d_i)} + m\sum_{i\in D} l_i \tag{25}$$

We minimize the difference in wasted votes divided by the number of total votes (the efficiency gap) plus a constant times the sum of the "diameters" of each district! This second term means the same thing as in the partisan bias model, and adjusting the value of m changes the relative importance of the two objectives.

3 Results

We ran our model on Massachusetts, which is divided into 351 towns and has 9 Congressional districts.

3.1 Data Collection

To run our model, we needed to find data for the values of our parameters (such as population) for Massachusetts towns, the atomic units of our model. Population data was taken from the 2010 US Census (through Wikipedia), and the voter data came from the Massachusetts Secretary of State's office [8]. The number of registered Democrats/Republicans was used as a proxy for actual election results.

Next, we collected data on the adjacency of towns. Unfortunately, this data was not available online, so we had to construct it by hand. Because the matrix was very sparse, we went through the 351 towns and for each town, wrote the names of the towns which were adjacent to it (by looking at a map of MA, also from the Secretary of State's office) [9]. After we made this list, we converted it to a matrix. If this data is not available online, it could represent a serious bottleneck for extending the model to larger states, such as California.

Finally, to get data on the distance between pairs of towns, we scraped Google for the coordinates of each town. We assumed that each town was represented by the coordinates of its approximate town center. Once we had the coordinates, we were able to easily convert pairs of coordinates into pairwise distances between town centers.

3.2 Partisan Bias Model

We implemented our models in Python and used Gurobi as our solver. Since our LP was designed to return an allocation of towns to districts, we wrote a short program in R to take the list of towns in each district and create a map of MA with each district labeled and shaded with a different color.

The model shows promise in being able to identify districts which are more equitable. For example, the following map results in a partisan bias of 1 seat, which is the best that can be done with an odd number of seats.

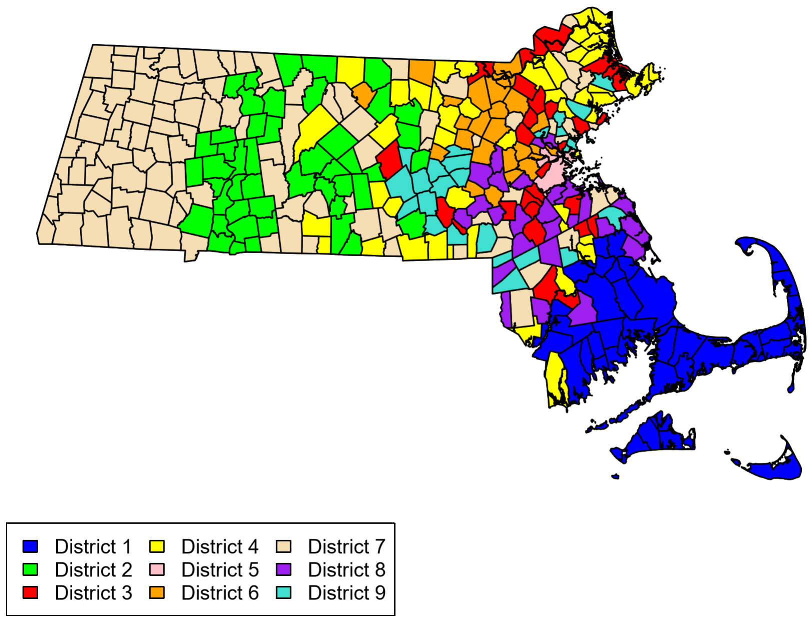

Figure 2: Sample district map minimizing partisan bias and sum of diameters of districts.

We ran this model for approximately 8 hours and it shows progress in constructing reasonably shaped districts (for example, district 1 and district 5, which is just Cambridge and Boston). Unfortunately, our model did not find the optimal solution; even after 8 hours, the solver still indicated a large gap between the incumbent solution and the best lower bound (1.09111 and 1.00335, respectively). Some districts are also clearly larger than they need to be. We discuss our issues with computational power in section 4.

To circumvent some of our issues with computing power, we tried creating constraints to limit the diameters of districts. We hoped that this would result in a more reasonable map without needing a large amount of computational resources. To give an example, by forcing two districts to be under 40km in diameter, four districts to be under 70 km, and the remaining three districts to be under 110 km, we can get an improved map:

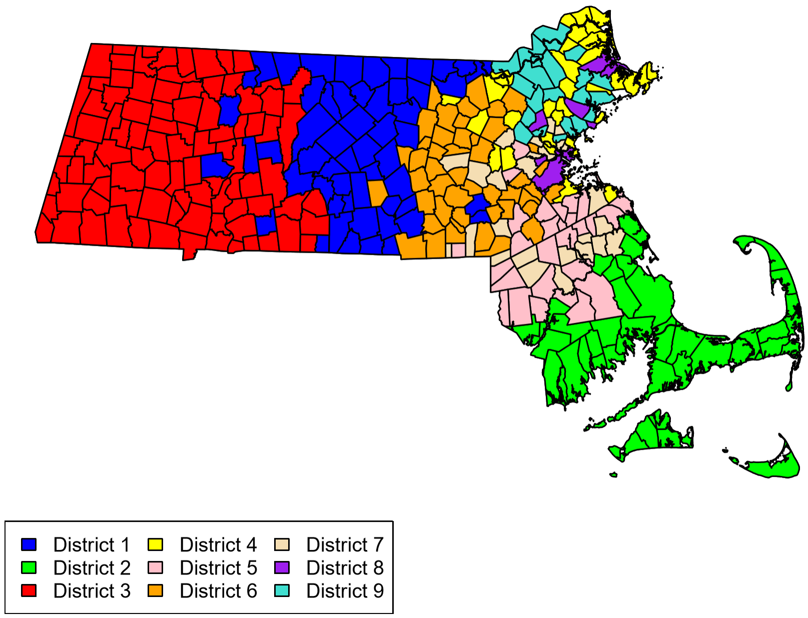

Figure 3: Sample district map minimizing partisan bias and constraining district diameters

However, notice that the districts are still not all contiguous; enforcing this is very non-trivial and will be discussed in section 4.

3.3 Efficiency Gap Model

We performed the same analysis as for partisan bias. It turns out that the lowest efficiency gap occurs when Democrats win all 9 districts, and the efficiency gap is 1.47%. This is a by-product of the fact that the breakdown of registered voters (used as a proxy for votes) is very close to 75% Democratic and 25% Republican. In a scenario where exactly 75% of voters are Democrats, an efficiency gap of 0 can be achieved if Democrats win all the seats (since both parties waste 25% of the vote). Thus, the main issue is trying to draw contiguous, reasonably shaped districts where Democrats win all 9 seats.

As in the partisan bias model, using our original model (where the sum of the diameter of each district is in the objective) leads to an extremely long runtime. Even after 20 minutes of running the model, the current solution is nowhere near optimal:

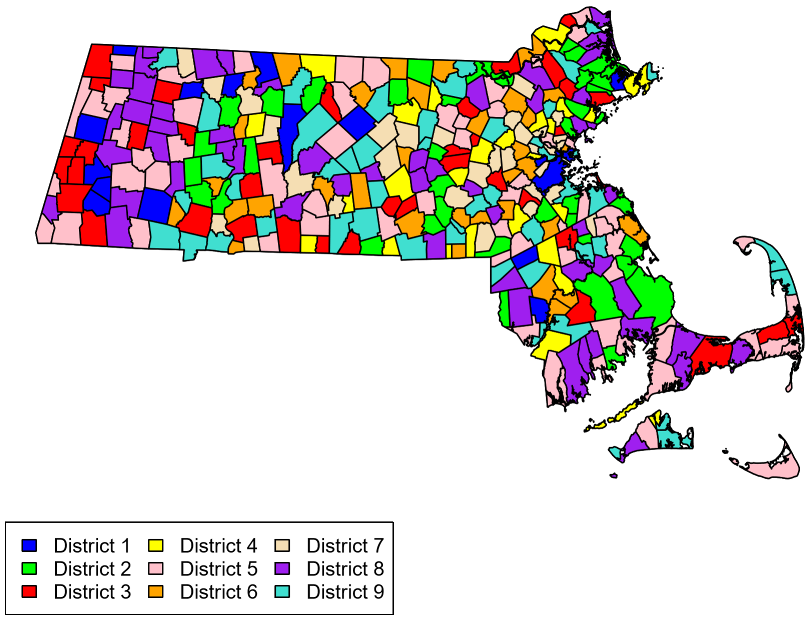

Figure 4: Sample district map minimizing sum of efficiency gap and sum of diameters of districts.

Also similarly to the partisan bias model, constraining the size of each individual district seems to be a promising approach. If we provide the same constraints as before (two districts under 40km in diameter, four districts under 70 km, and the remaining three districts under 110 km), we get the following map:

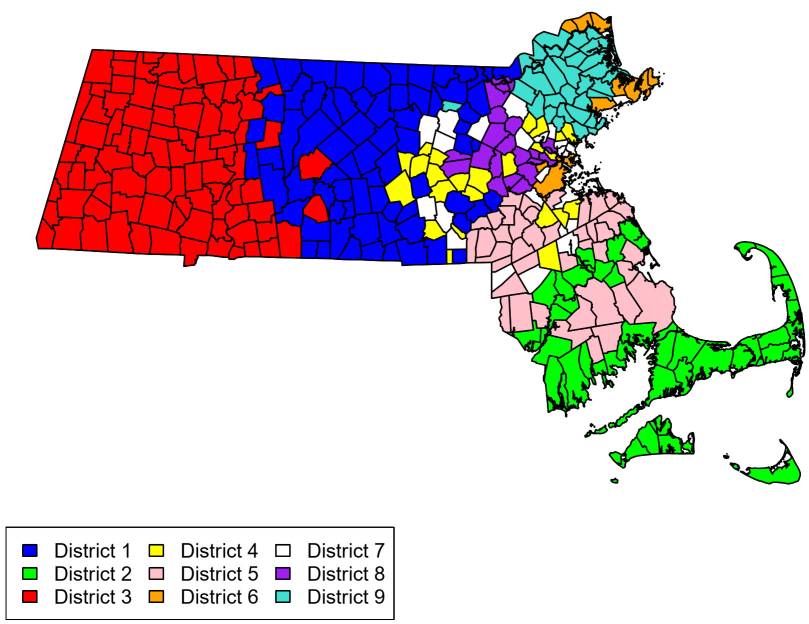

Figure 5: Sample district map minimizing efficiency gap and constraining district diameters

4 Discussion

A major roadblock in our project was computing power. Our original model did not find the optimal solution even after several hours, and if we stopped the solver during this time, the current solution was far from optimal. Most districts were still constructed from towns scattered across the map. We tried to solve this problem by using AWS servers for computation, but because Gurobi's academic license requires you to be on an academic network, this failed. We also tried to write the model in CVXOPT instead (which doesn't require a license), but the model was too large, resulting in a memory error. This is because CVXOPT requires you to write out the constraint coefficients in matrix form.

Another roadblock was trying to force districts to be contiguous (i.e. consist of one continuous piece). While this is only a legal requirement in 23 states, it is essentially an unwritten law, as nearly all districts are drawn this way [10]. Minimizing the sum of the diameters of the districts seems like a good approach but takes more computing power than we had. Constraining the sizes of each individual district gave fairly good results, and by performing a closer examination of the population of each district and looking at the sizes of current districts as a baseline, these constraints could be refined even more.

Rigorously enforcing contiguity in integer programs is an area of active research and has been applied to fields like forestry. One promising approach is to look at each pair of points in a district and force the district to include at least one point in each cut separating the two points [11]. Another possible approach is to use flow constraints [12]. Unfortunately, these approaches are computationally expensive (they require many constraints) and so they would run into the same roadblocks as our current approach. When using the flow constraints for contiguity from [12] on District 1 only, we were able to generate some examples of continuous districts (see below).

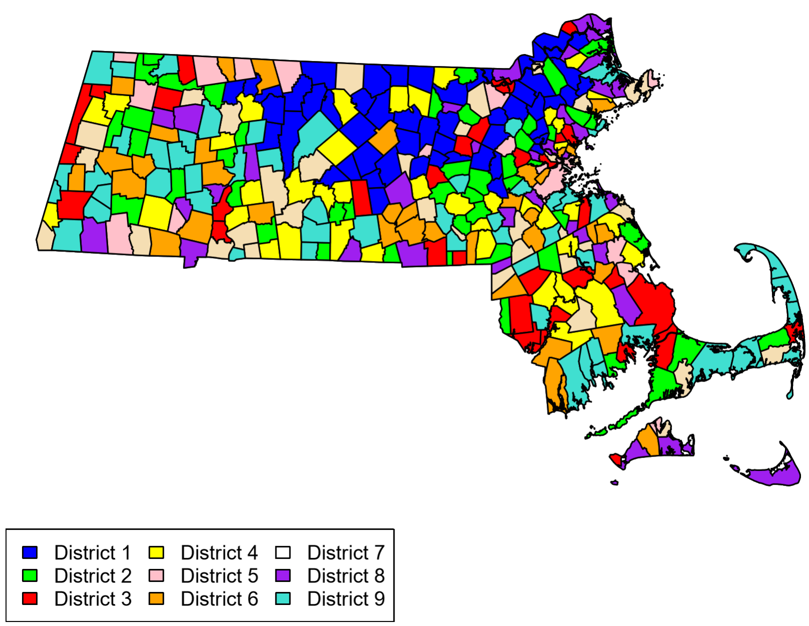

Figure 6: Sample district map using flow contiguity constraints on District 1

These constraints can be generalized to create multiple contiguous districts. For example, by creating a second set of flow constraints for District 2, we were able to create the following map, where both Districts 1 and 2 are contiguous.

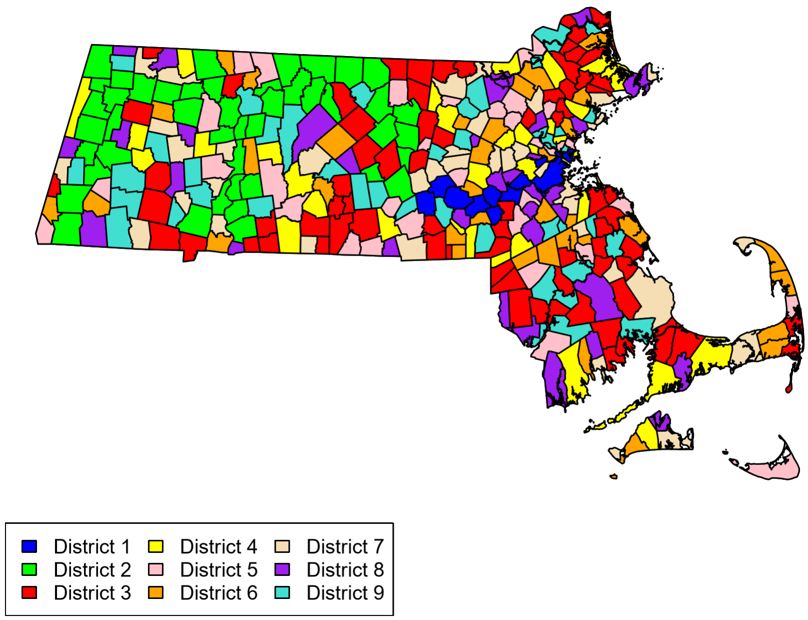

Figure 7: Sample district map using flow contiguity constraints on Districts 1 and 2

However, there are two issues with this approach. The first is that just enforcing contiguity without any restrictions on the size/shape of a district will sometimes create very long or irregular shapes that do poorly on geometric tests like the convex hull or Polsby-Popper ratio (for example, Districts 1 and 2 in the previous figure). Contiguity constraints must be combined with other constraints, like limits on the diameter of a district.

Furthermore, Gurobi takes a long time to generate any feasible solutions; the model can run for 10+ minutes without generating an incumbent feasible solution. One way to address this issue is to set Gurobi's "MIPFocus" parameter to 1; this makes the solver focus more on finding feasible solutions. Another technique we used was to feed Gurobi a partial solution, where we indicated the flow "sink" for each district and asked it to check the resulting family of solutions for an incumbent. This was used to generate the previous solution by feeding Boston and Pittsfield as the "sinks" for Districts 1 and 2, respectively. A disadvantage is that this requires us to guess an acceptable sink for each district, which may be extremely hard to do for more than a few districts. Finding a more efficient way to generate contiguous districts represents a major area for future research.

5 Conclusion

Drawing congressional districts to minimize two important voter-based measures of gerrymandering (partisan bias and efficiency gap) can be modeled as a linear program. States can be broken up into discrete units (such as towns in Massachusetts, or counties in other states) and assigned to districts in a way that is fair to both parties. By implementing the model in Python, we were able to generate sample maps that minimized both these metrics, showing the potential effectiveness of our model as a redistricting tool. The main obstacle we encountered was trying to draw districts that were contiguous, which is more or less a legal requirement. Our attempt to do this by minimizing district "diameters" was infeasible given our computing resources, and other approaches, such as giving each district a maximum diameter, have drawbacks. One promising approach for future research was using flow constraints; this provides a rigorous guarantee that districts will be contiguous but has drawbacks in terms of computational complexity.

6 References

[1] J. Levitt. *All About Redistricting – Who Draws the Lines*, Loyola Law School. http://redistricting.lls.edu/who-fed10.php (2019).

[2] M. Astor & K. Lai. *What's Stronger than a Blue Wave? Gerrymandered Districts*, New York Times. https://www.nytimes.com/interactive/2018/11/29/us/politics/north-carolina-gerrymandering.html (29 November 2018).

[3] C. Ingraham. *This is the Best Explanation of Gerrymandering You Will Ever See*, Washington Post. https://www.washingtonpost.com/news/wonk/wp/2015/03/01/this-is-the-best-explanation-of-gerrymandering-you-will-ever-see/?utm_term=.bf1d7003b6c5 (1 March 2015).

[4] *Final Report on 2011 Redistricting*, State of California Citizens Redistricting Commission. https://wedrawthelines.ca.gov/wp-content/uploads/sites/64/2011/08/crc_20110815_2final_report.pdf (15 August 2011).

[5] N. Stephanopoulos & E. McGhee, *Partisan Gerrymandering and the Efficiency Gap* (Public Law and Legal Theory Working Paper No. 493, 2014). https://chicagounbound.uchicago.edu/cgi/viewcontent.cgi?article=1946&context=public_law_and_legal_theory.

[6] M. Duchin (2018). *Gerrymandering metrics: How to measure? What's the baseline?* https://arxiv.org/pdf/1801.02064.pdf

[7] I. Smythe, *Mathematics and Politics – Lecture 15.* http://pi.math.cornell.edu/~ismythe/Lec_15_web.pdf

[8] W. Galvin. *Enrollment Breakdown as of 10/17/2018*, Massachusetts Secretary of State's Office. https://www.sec.state.ma.us/ele/elepdf/enrollment_count_20181017.pdf (17 October 2018).

[9] W. Galvin. *Massachusetts Cities and Towns*, Massachusetts Secretary of State's Office. http://www.sec.state.ma.us/cis/cispdf/city_town_map.pdf (27 February 2019).

[10] J. Levitt. *All About Redistricting – Where the Lines are Drawn*, Loyola Law School. http://redistricting.lls.edu/where-state.php (2019).

[11] R. Carvajal, M. Constantino, M. Goycoolea, J. Vielma, & A. Weintraub (2013). *Imposing Connectivity Constraints in Forest Planning Models.* Operations Research, 61(4), 824-836. http://www.mit.edu/~jvielma/publications/Imposing-Connectivity-Constraints.pdf

[12] T. Shirabe (2005). *A Model of Contiguity for Spatial Unit Allocation.* Geographical Analysis, 37(1), 2-16. https://onlinelibrary.wiley.com/doi/epdf/10.1111/j.1538-4632.2005.00605.x