# Optimal Redistricting in North Carolina

Luca Mingardi, Alexandru Socolov, Luis Honsel
lucam@mit.edu socolov@mit.edu honsell@mit.edu

April 2020

## 1 Introduction

One of the necessary conditions for democracy is a fair election procedure. In the US, however, gerrymandering has been flagged to pose a threat to fair elections in which one party gets the right to draw the voting district lines, potentially in their favor. By cracking and packing the voters, one can get a map that results in vastly different election outcomes that distort the true population preferences. In this project, we take a computational approach to redistricting proposing three algorithms for automatically drawing the voting district maps.

## 2 Motivation and Challenges

The importance of our project stands in the possibility of applying it to different situations. For example, it could be used as a baseline plan for judges and decision makers, or it could directly be exploited as a good draft from which starting to draw the new district plan.

A valid redistricting needs to be compliant with a number of laws and indications, thus we have considered four different metrics to evaluate our approaches:

- Compactness
- Population share
- Voting share
- Racial share

Then, we have defined a convex function having as objective a combinations of the above metrics, with the aim of optimizing it to find a global minimum. In reality, because of path disconnectedness, this becomes an extremely complex problem[1], which is currently unsolved. Thus, we have developed three heuristics in order to tackle this problem and find a good solution for it.

[1] https://arxiv.org/abs/1808.08905

# 3 Data

Given the nature of our project, it was of critical importance to find some granular data having information about voting trends, together with a shapefile to be flexible and precise when drawing new district borders. While searching online, we found that North Carolina is one of the States that is more open about data sharing, and particularly, we found an amazing repository [2] with very detailed information about the VTDs of the State.

The voting district (VTD) is the most granular polling area that we have been able to find in the form of a shapefile. The data was obtained from the North Carolina General Assembly's website and processed by members of the Metric Geometry and Gerrymandering Group (MGGG). Specifically, demographics data has been aggregated from the census block level to VTDs using the General Assembly's Block Level Key. The file includes the following information about the 2692 VTDs in North Carolina:

- VTD identifier, area, perimeter and shape
- Total population, voting population, distribution of races
- Voting data for the Presidential and Senate elections from 2008 to 2016
- District assignments from 2011, 2016 and Judge plan

The specific aim of our project has been to analyze the 2016 district plan, trying to improve it over different metrics. The detail and accuracy of the data has really made possible to work from a very small geographic entity to form new district shapes.

# 4 Methods for redistricting

## 4.1 Weighted K-Means

The first method for assigning the VTDs to districts is Weighted K-Means [3]. Inspired by the unsupervised machine learning technique K-Means, the algorithm weighs the distances between any two points by population share, racial group share and political view differences.

### Algorithm

The algorithm works as follows:

0. Find the center of each VTD characterized by longitude and latitude.
1. Pick n random centers as centroids and initilize the weights $w_k = 1/K, \forall k$ where $K$ is the number of districts to be drawn.

[2] `https://github.com/mggg-states/NC-shapefiles`

[3] Guest, O., Kanayet, F. J., Love, B. C. (2019). Gerrymandering and computational redistricting. *Journal of Computational Social Science, 2*(2), 119-131.

2. Assign all the other blocks to the centroid that minimizes the weighted distance. The distance between VTD $i$ and centroid of cluster $k$ is calculated as:

   $D_{ik} = \frac{w_k}{\sum_{k=1}^{n} w_k} * Distance_{ik}$

   Where $\alpha, \beta, \gamma \in [0, 1]$ are parameters controlling the relative importance across the three metrics of fairness. $Distance_{ik}$ is the great-circle distance which respects the curvature of the Earth.

3. Re-calculate each centroid as the mean latitude and longitude within each cluster.

4. Update the weights according to the following formula:

   $w_k = |population_k|^{\alpha} * |black\_population_k|^{\beta} * |democratic\_share_k|^{\gamma}$

   Here, $population_k$ is sum of the population of VTDs currently assigned to cluster $k$. Similarly, $black\_population_k$ is sum of the black population. Lastly, $democratic\_share_k$ is sum of democratic votes in cluster k VTDs divided by the total votes across these VTDs in 2016 gubernatorial elections.

   For convergence issues, we updated the weights gradually, controlling how fast we change the weights with $\beta$. In particular,

   $w_k^{new} = \beta * w_k^{newly\_calculated} + (1 - \beta) * w_k^{old}$

5. Repeat 2 and 3 until the coordinates of the centroids don't change much:

   If $\sum_{i=1}^{2} |\frac{c_{ik}^{new} - c_{ik}^{old}}{c_{ik}^{old}}| < \epsilon$ for at least one k (cluster) then terminate

### Results

Since the model takes multiple parameters as inputs, we experiment with $\alpha, \beta$ and $\gamma$ that represent the relative importance of having equal populations vs racial group populations vs partisan preferences across districts. Depending on these parameters, we arrive at the following maps (Figure 5).

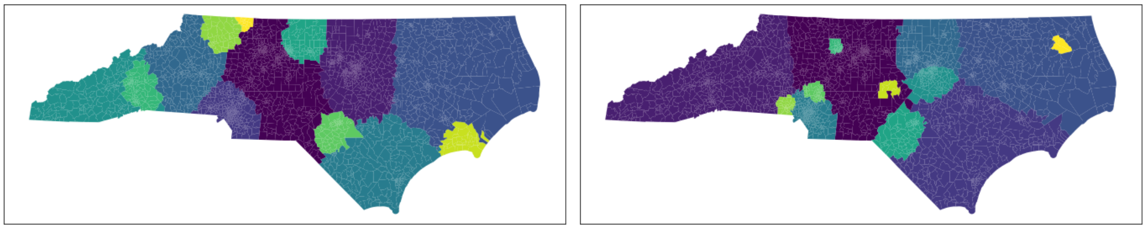


Figure 1: NC Redistricting maps according to Weighted K-Means algorithm. Left: weighted by population differences only, $\alpha = 0.9$. Right: weighted by population and partisan differences, $\alpha = 0.5$ and $\gamma = 0.9$

### Evaluation

Among the advantages of this method are:

- Guaranteed contiguity. Since the points are assigned to the nearest centroid based on some distance, the districts are never split apart. In other words, there is always a path connecting any two VTDs in a cluster.
- Great compactness. A property of the great-circle distance metric, the clusters are nudged to be of circular shapes, which translates into a good performance on the compactness score (more on this in Results).
- Parameters represent priorities. Depending on the policy-makers relative preferences for what measure of fairness they want to prioritize, $\alpha, \beta$ and $\gamma$ can accommodate that.
- Computationally quick. The algorithm solves in roughly 20 seconds without huge efforts to optimize the code at the development stage.

On the other hand, the drawbacks of the Weighted K-Means algorithm are:

- Unstable map drawings. Due to the random initialization of the first n centroids, the algorithm results in different maps every time its run unless we fix the seed.
- Sometimes the algorithm draws districts inside other districts, which is considered illegal. This usually happens when one of the parameters is close to one, e.g. map on the right in Figure 5.
- Over-prioritizes circular shapes. Sometimes rectangular shapes are as preferred as circular ones if they result in fairer population distributions.

We will return to evaluating the maps across multiple metrics in Results. For now, we proceed with two more alternative approaches for computational redistricting.

## 4.2 2-OPT with random start

This method uses a classical heuristic algorithm widely used in optimization, the 2-OPT algorithm. Basically, this method starts from one feasible solution and explores all the neighboring ones that improve our cost function. Before going in detail into the 2-OPT algorithm, let's first see how we generate feasible solutions to start with.

**Algorithm: Random Start**

The Random start algorithm assign 13 random points to 13 distinct districts. At each iteration, each district gets added a new point, selected randomly among its neighboring available point. The algorithm stops when there are no available points left.

0. Let S be the set of available point. S is comprised of all the states VTDs for now. We randomly select 13 points among S that will define the 13 distinct clusters.
1. Remove those 13 points from S
2. While S is not empty :

    (a) For each cluster look at the neighboring points still in S
    (b) Randomly select one of these points and add it to the cluster
    (c) Remove this point from S

The following figures give you the kind of redistricting you obtain though this method.

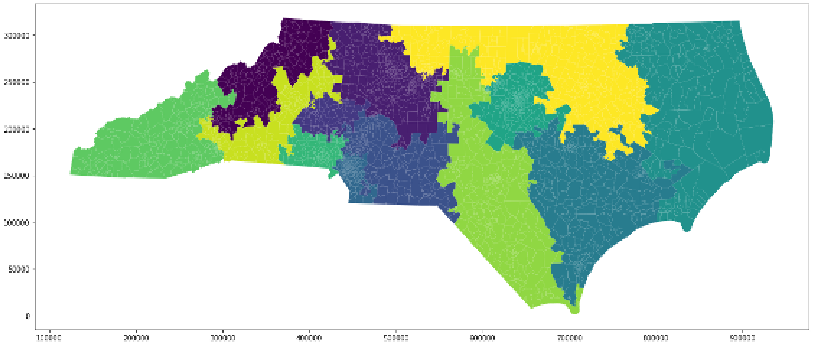


Figure 2: NC redistricting obtained through random start

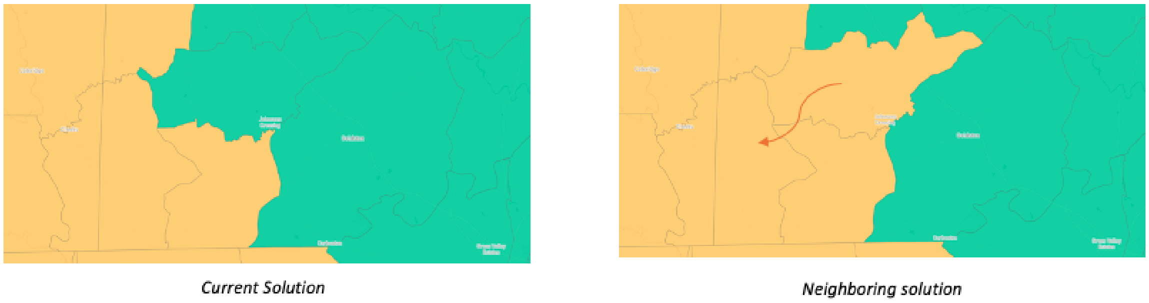
**Algorithm: 2-OPT**

Once we have our feasible solutions to start, we can use the 2-OPT algorithm. The main idea is to explore all neighbouring solutions of our current solution. A neighbor will be a solution that only differs by one point. If the neighbor has a better score, it will be your new current solution. The algorithm runs until there are no more improvements possible.

0. Let V be our current solution

1. While there are changes in the following loop

    (a) Look at every neighbor N of V
    (b) If N has a better score than V, $V := N$
    (c) Otherwise move to another neighbouring solution



*Current Solution*

*Neighboring solution*

Figure 3: Example of a neighbouring solution

### Evaluation

This method presents different advantages, among which:

- Guaranteed contiguity: given the way the random start algorithm works, adding VTDs to clusters one at a time, and also given the way the 2-OPT algorithm works, we are assured that there will always be a path between any two points inside a cluster.
- The optimization part allows this method to provide better score than Weighted K-means or random start methods.
- This model can provide different outputs depending on the metric we wan to prioritize.

On the other hands, some of the drawbacks are:

- Locally optimal solution: Given that the 2-OPT method only examines the neighbours of a current solution, we can easily get stuck in a locally optimal solution and be far from a globally optimal one.
- Because of the way the districts are built, this methods does not perform as well as Weighted K means for the compactness score.
- This method is highly dependent on the starting point and we observe a high variance at each new output of the algorithm.

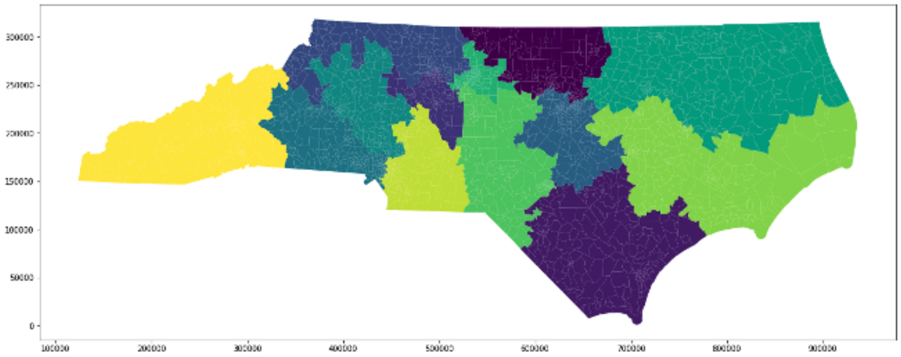


Figure 4: NC redistricting obtained through 2-OPT

## 4.3 Simulated annealing

As we have seen previously, one of the main drawbacks of the 2-OPT method is that it is very locally optimal. While the 2-OPT algorithm only accepts points that lower the objective function, the simulated annealing algorithm accepts not only new points that lower the objective, but also, with a certain probability, points that raise the objective. By doing this, the algorithm avoids being trapped in local minima, and is able to explore globally for more possible solutions. The probability is calculated using a temperature parameter, that will decrease through iterations, reducing the extent of the search and thus assuring that the algorithm will reach a local optimum at some point.

### Algorithm

The simulated annealing method needs a feasible solution has an input. We use the random start algorithm to generate it.

Input: Feasible solution V, number of iterations $t_{max}$, initial temperature $T_0$, temperature parameter $\alpha$.

0. Let V be our current solution

1. For t ranging from 1 to $t_{max}$:

   (a) Look at every neighbor N of V

   (b) If N has a better score than V, $V := N$

   (c) Otherwise, if we note C(V) and C(N) the costs of solutions V and N, do $V := N$ with a probability

   $$e^{-(C(V)-C(N))/T(t)} \tag{1}$$

   where $T(t)$ is the temperature at period $t$ defined as $T(t) = T_0 * \alpha^t$

### Evaluation

This method shares the sames advantages and drawbacks as the 2-OPT ,method. However, it performs much better on every metric, and can be seen as an improvement of the previous method. More on the performance in the Results section.

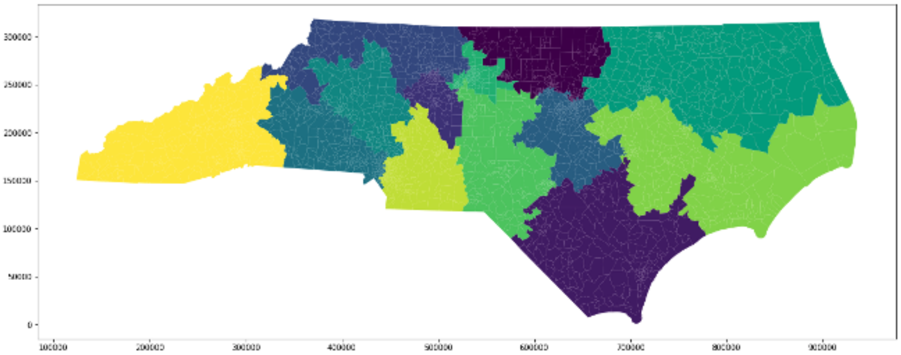


Figure 5: NC redistricting obtained through Simulated Annealing

## 5 Metrics

In order to evaluate how good how redistricting is, we have defined four metrics:

- Compactness score $= 1 - \frac{\text{Area of District}}{\text{Area of the Conved Hull of the District}}$

- Population share score $= \frac{|\text{Uniform population share - District population share}|}{\text{Uniform population share}}$

- Voting share score = | Democratic voters share in state - Democratic voters share in district |
- Racial share = | Black population share in state - Black population share in district |

All of these metrics are designed for a minimization problem, as they achieve a better performance as they decrease.

# 6 Results

In the table below, it is possible to observe the performance of the different approaches we have followed, with the details for the metrics that we have just defined

| Plan / Score | Compactness | Population | Racial | Democratic |
|---|---|---|---|---|
| Weighted K-Means | **0.16** | 9.11 | 1.38 | 1.26 |
| 2-OPT | 0.26 | 2.61 | 0.83 | 0.68 |
| Simulated Annealing | 0.25 | 2.63 | **0.82** | **0.67** |
| 2016 Plan | 0.29 | 0.02 | 0.84 | 0.91 |
| Judge Plan | 0.22 | **0.01** | 0.97 | 0.99 |

According to the metrics that we defined, Weighted K-Means is the best plan for compactness, the judge plan for population score and simulated annealing for racial and democratic score. It is very interesting to observe that the 2016 plan isn't the best plan in any of the metrics, thus it was very important to change and improve it.

In Figure 6, we plotted the different seat-vote curves for the different approaches we proposed.

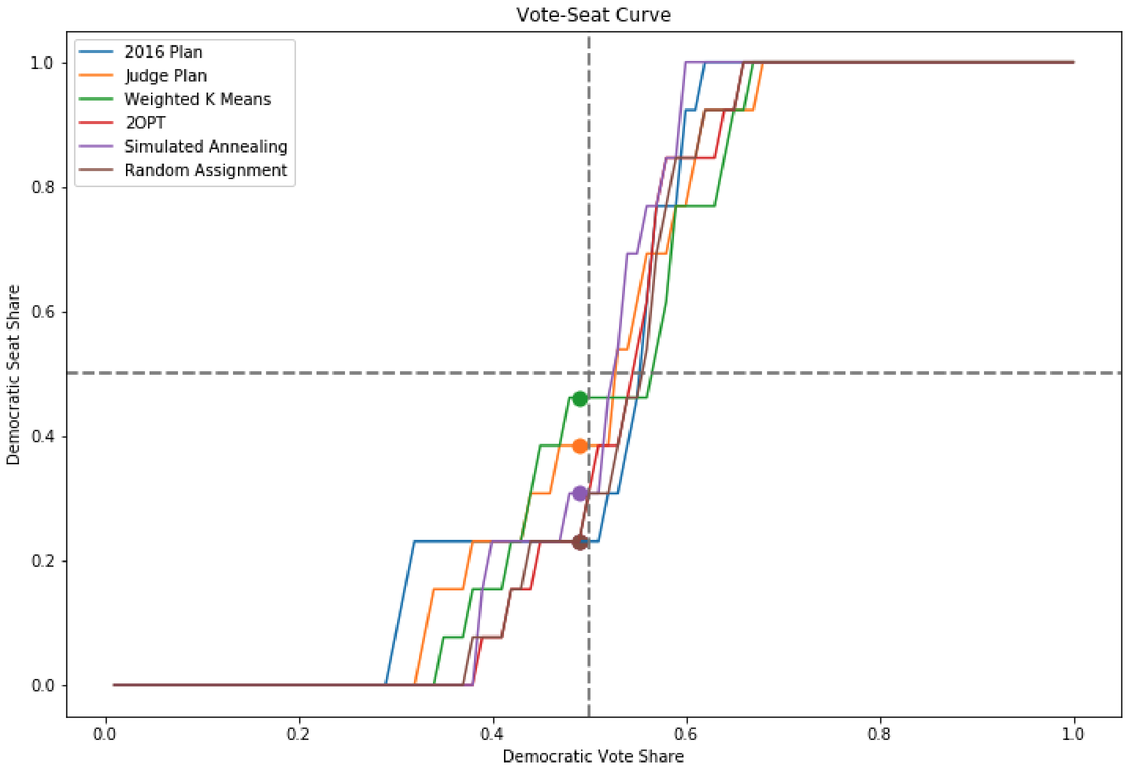


Figure 6: Seat-Vote curve for the different plans

The Seat-Vote curve is a standard measure of voting fairness, as it shows the outcome of an election and also the counterfactuals of it, meaning what would have happened if there would have been a certain swing in voting choices. This metric is based on the idea that ideally, when a party gets 50% of the votes, then it should take also 50% of the seats available. It's strongest flaw is that is assumes that vote swings happen uniformly in the population, which is usually not the case. However, it can still be useful to interpret the results of an election, and we have used it to evaluate the district plans that we got with our models.

From Figure 6, we can see that Weighted K-Means is the one providing the fairest result according to this metric (48.95% votes, 46.15% seats), followed by Judge Plan (48.95% votes, 38.46% seats), Simulated Annealing (48.95% votes, 30.76% seats) and then there is a tie between 2-OPT, the Random Assignment and the 2016 Plan (48.95% votes, 23.07% seats).

## 7 Conclusion

Redistricting is widely known to be an extremely complex problem, which gets even more complicated when you have to take into consideration also socio-demographics factors. In this project, we acknowledge that we have not provided a final solution for this matter, but we believe that we have proposed three creative and useful ways to start looking at this problem. One of the strength of our work is that it is highly flexible in the way in which it optimizes over the different metrics defined. This means that if the government wants to give more importance to one of them, it is very possible to do so, thus allowing the creation of different shapes accordingly. Alternatively, the maps generated by out algorithm could be used as a ”warm start” to redistricting by people or other algorithms. This way we hope to nudge the decision makers to start from more equitable scenarios rather than a blank canvas.