Unbiased Congressional Districts

Alex Benn
abenn1@swarthmore.edu

David German
dgerman1@swarthmore.edu

Abstract

This paper presents a strategy for dividing a state into congressional districts using a modified version of Smith and Ryan's recursive shortest splitline algorithm (2007). Our strategy reduces the cost of the computation by approximating the population of a ZIP Code Tabulation Area (ZCTA) as a point mass at its population centroid. If there are N ZCTAs in the state with E total edges, and k districts to be created, our algorithm runs in expected $O((N^2 + E) \cdot \log k)$ time.

1 Introduction

The determination of congressional district boundaries within a state is often a controversial process. Typically, any solution that gives contiguous, equipopulous districts is legal. Therefore, the majority party in the state legislature has wide latitude to draw districts that bolster its chances in congressional elections. Such gerrymandered districts will typically disperse regions of strong support for the opposition party among the districts so that the majority will control most or all seats.

We adopt the following definitions. An *unbiased districting* is a division of a region into equipopulous districts that is developed without any information about the residents' voting preferences. A *gerrymandered districting* is a division of a region into equipopulous districts such that the region's majority party is the majority party in every district.

2 Related Work

The literature contains several proposals for generating or maintaining unbiased districts that optimize some geometric criterion.

Given an initial districting, Helbig et. al. heuristically maximize district compactness using an iterative linear programming technique (1972). In their method, the region is discretized into m existing political zones, such as counties or census tracts, that are small compared to the whole state and approximately equipopulous. They then proceed as summarized in Algorithm 1. Constraint A is the one-vote constraint: each zone belongs to exactly one district. Constraint B is the equipopulation constraint. It relies on the assumption that zones are equipopulous, but this restriction could easily be lifted by rewriting the constraint as an inequality on the population sum, with some deviation tolerance. The value function to be minimized is simply the summed population-weighted distance of zones from the centroids of their districts. Since finding a minimum by brute force takes $O(2^m)$ time, the transportation algorithm is used; the details are out of the scope of this paper. Note that this method does not guarantee that districts will be contiguous.

Macmillan presents an innovative solution to the contiguity problem (Macmillan, 2001). Macmillan first presents the connectivity method of checking contiguity of a region, which is a streamlined, but still $O(m^3)$, version of connectivity matrix multiplication. Macmillan improves on this algorithm by initially assuming that the regions in the districting are already connected, and what is being examined is the decision to either add a single zone to or remove a single zone from the current region. Since the only way the region can become disconnected is if adding or removing the zone breaks contiguity, we need only examine the single zone and the zones it borders. Using Macmillan's switching point technique, it can be determined quickly whether the operation breaks contiguity. An initial districting can then be updated in response to population movement

Algorithm 1 Helbig et. al. linear transport

```
Let $\epsilon$ be an arbitrary stagnation threshold.
Let $m$ be the total number of zones.
Let $n$ be the number of districts.
$g = m/n$
Let $(u_i, v_i)$ be the centroid of zone $i$.
Let $p_i$ be the population of zone $i$.
repeat
  for All districts $j$ do
    Compute $(a_j, b_j)$, the centroid of district $j$.
    for All zones $i$ do
      $d_{ij} = \sqrt{(a_j - u_i)^2 + (b_j - v_i)^2}$
      if zone $i$ is in district $j$ then
        $x_{ij} = 1$
      else
        $x_{ij} = 0$
      end if
    end for
  end for
  constraint A: $\sum_j x_{ij} = 1$ for all $i$.
  constraint B: $\sum_i x_{ij} p_i = g$.
  Using the transportation algorithm, minimize $\sum_i \sum_j (x_{ij} \cdot d_{ij} \cdot p_i)$ subject to constraints A,B.
  for All districts do
    Let $(\bar{a}_j, \bar{b}_j)$ be the centroid of the new district $j$.
  end for
until $|\bar{a}_j - a_j| < \epsilon$ and $|\bar{b}_j - b_j| < \epsilon$ for all $j$.
```

by simulated annealing. In this process, donor districts with excess population and recipient districts with a population deficit are selected at weighted-random by the size of the deviation. A zone to transfer is selected at weighted-random by the size of the resulting deviation. So long as the transfer would not break contiguity, it is accepted if it strictly improves deviation, and probabilistically accepted based on the current annealing temperature otherwise. Macmillan's algorithm does not optimize for compactness or any other geometric property, and therefore may yield results as unattractive as a gerrymander.

The shortest splitline algorithm of Smith and Ryan recursively divides the region to be districted into two regions (Smith and Ryan, 2007). If the region being divided has an even number of congressional seats, the two child regions are equipopulous. If it has an odd number, then one region will be large enough to have one more seat than the other. Smith and Ryan compute splitlines on a grid sampling of population. The details of their implementation are only given as uncommented C source code, and are therefore not summarized here.

Our literature search did not discover any existing work explicitly concerned with optimal gerrymandering. However, the generalized equitable ham sandwich algorithm of Bespamyatnikh et. al. is applicable (1999). This algorithm splits a set of $r \cdot p$ red points and $r \cdot q$ blue points into r regions with p red points and q blue points using a divide-and-conquer strategy. Using voter registration rolls, registered Democrats (blue points) and Republicans (red points) can be identified with geographic locations. Letting r equal the size of the state's congressional delegation, the majority party will control every district in the equitable ham sandwich subdivision.

3 Implementation

We have implemented a shortest splitline algorithm using the basic concepts developed by Smith and Ryan (2007), but with the adapted implementation summarized in Algorithm 2. The algorithm produces districts that are contiguous, and equipopulous to a variable tolerance parameter ϵ. Degeneracy handling is not included in the pseudocode, but has been implemented and is discussed further in Section 3.3.

Smith and Ryan use a raster model for the region: boundary points are represented by pixels, and a splitline starts and ends at the centers of pixels. This model has the problem of fixed and arbitrary granularity: edges cannot shift by less than one pixel at either end, and the granularity is set by the resolution of the original raster image of the region. If the original image is too small, the resulting splitline may be unsatisfactory in population distribution, but if the original image is too large, the algorithm takes much longer.

Instead of working within the limitations of a raster image, we chose to use the ZIP code as our fundamental element. Populations are stored by ZIP code rather than by pixel. Splitlines start and end on ZIP code population centroids, and the determination of the population of each district is calculated by summing the populations of the ZIP codes whose centroids fall within that region. The outer boundaries of the ZIP code zones comprise the border of each district.

3.1 Data Set

ZIP codes have a number of virtues as fundamental elements for the software. They are generally sized to contain roughly even populations, so areas of high population density are broken down into smaller ZIP codes. However, ZIP code regions are determined arbitrarily by the post office and tend to follow postal routes rather than being shaped according to any metric of geographical cohesiveness.

The US Census has found it useful to map out the regions containing all addresses in each ZIP code, called Zip Code Tabulation Areas (ZCTAs). Map overlays in standard formats are available online from the US Census's website for all fifty states.[1] Our implementation extracts data from the ArcView Shapefile format.

Grubesic and Matisziw's use of census ZCTA maps for epidemiological studies (2006) helpfully clarified that ZCTAs in which three numbers are followed by an `HH` represent water surface, and those with an `XX` represent unpopulated land. For example, any water feature mostly within the `303` ZIP3 ZCTA would be labeled `303HH`.

[1] `http://www.census.gov/geo/www/cob/z52000.html`

Algorithm 2 ZCTASplit

```
Let $\epsilon$ be an arbitrary population deviation tolerance.
Let $k$ be the number of districts.
Let $t$ be the population.
Let $R$ be the set of ZIP codes in the region.
if $k = 1$ then
  return $R$
end if
for Each edge $e$ of a ZIP in $R$ do
  If $e$ adjoins only one ZIP, it is in the set of possible boundary edges $B$.
end for
Find the northwesternmost edge $p$ in $B$. {$p$ must be on the outer boundary.}
$Z \leftarrow \emptyset$
Let $c$ be the edge following $p$ on the ZIP $z$ that $p$ adjoins.
while $c \neq p$ do
  $Z \leftarrow Z \cup \{z\}$
  while $c \in B$ do
    $c_n \leftarrow$ the edge following $c$ on $z$.
    $c \leftarrow c_n$
  end while
  $z \leftarrow$ the ZIP that $c$ adjoins that is not $z$.
  $c_n \leftarrow$ the edge on $z$ in $B$ that shares a vertex with $c$.
  $c \leftarrow c_n$
end while
$g \leftarrow \lfloor k/2 \rfloor \cdot t$ {$g$ is the goal population.}
$l \leftarrow \infty$
for All $z_1 \in Z$ do
  for All $z_2 \in Z$ do
    Draw a splitline connecting the centroids of $z_1$ and $z_2$.
    $a \leftarrow$ the population above the line.
    $d \leftarrow$ the distance between the centroids.
    if $(1-\epsilon)g \leq a \leq (1+\epsilon)g$ and $d < l$ then
      $l \leftarrow d$.
      $s_b \leftarrow (z_1, z_2)$
    end if
  end for
end for
if $s_b$ is not defined then
  error no acceptable splitline found
else
  $U \leftarrow$ ZCTASplit($\epsilon$, $k = \lfloor k/2 \rfloor$, $t = g$, $R =$ ZIP codes above the splitline)
  $O \leftarrow$ ZCTASplit($\epsilon$, $k = \lceil k/2 \rceil$, $t = t - g$, $R =$ ZIP codes below the splitline)
end if
return $U \cup O$
```

Since the ZCTA maps simply delineate each ZCTA by one or more labeled polygons, no neighborhood information can be directly determined from the ZCTA shapefile. This complicates finding the boundary of a region to split, an issue that is dealt with in Section 3.2.

We have been testing our implementation with publicly available voter rolls from the state of Georgia.[2] These voter rolls contain ZIP code, latitude, longitude, and party registration. All voter entries for a single ZIP code are placed within that ZIP code, and the centroid of the population of that ZIP code is found. This population centroid is used to determine the side of a splitline on which each ZIP code falls.

3.2 Strategy

Algorithm 2 can be divided into two phases. First, ZIP codes on the boundary of the region are identified. A ZIP is a candidate to be on the boundary if it has at least one edge that no other ZIP code in the region shares. Since the census ZCTA polygons may not perfectly tessellate the state, a walk is performed to identify the actual edge. This walk begins at the northwesternmost edge in the entire set of candidate boundary edges, which bounds some boundary ZIP z_0. The edges of z_0 are traversed in the cyclical order extracted from the shapefile until an edge e that also bounds some other ZIP z_1 is reached. The walk then moves onto z_1. One edge adjacent to e bounding z_1 is in the set of candidate boundary edges. The walk returns to the boundary by proceeding in the direction of that edge. When the walk revisits the first edge, all visited ZIPs z_n are returned.

Then, for every pair of boundary ZIPs (z_a, z_b), a splitline is drawn between the population centroids. Each ZIP population centroid in the region is classified as above or below the splitline, and the populations in each half are summed. If the ratio of population above the splitline to below the splitline is within some arbitrary tolerance of the equipopulous ratio, the splitline is a legal candidate. The shortest legal candidate splitline is accepted, and the algorithm recurses on the two ZIP sets returned.

[2] `http://grso.uga.edu/voter/`

3.3 Degeneracies

3.3.1 Rivers and Lakes

`HH` regions representing rivers can wander from the border across large swaths of a state, making it difficult to identify ZIP codes on the boundary. Simply removing the water from the dataset directly places a large number of interior ZIP codes on the apparent boundary, which is no improvement. When a split is made with water preserved, allocating water that crosses the boundary to one side or the other produces similar problems upon recursion.

The solution is to allocate the entire body of water to both regions. The boundary walk then proceeds along the edge of the water until it eventually returns to the proper boundary on the other side. The resulting boundary will thus include protrusions that are not actually part of the intended district. While potentially bizarre in appearance, these protrusions are not disruptive to the algorithm because they contain no population. Since regions will eventually contain water that is entirely disconnected from their actual land area, the boundary edge traversal must begin on the northwesternmost edge in B that adjoins a populated ZIP in the region.

3.3.2 Eccentric Splitlines

Consider the line containing the segment that connects the centroids of two boundary ZIP codes z_1 and z_2. If this line crosses the region boundary at any other ZIP code, the subsets of ZIPs produced by splitting on the line may be misidentified. We call such lines eccentric splitlines.

As a simple example, consider z_1 and z_2 on the boundary of a circular region, with populations distributed such that the line connecting the centroids is perpendicular to the local boundary. A splitline from z_1 to z_2 appears to divide the region in half, when it actually should separate z_1 and z_2 from all other ZIPs.

To work around this degeneracy, we simply reject any eccentric splitline. The check for eccentricity is fast. Recall that the boundary walk produces the boundary ZIP codes in order. After classifying all ZIP codes as above or below the line, we start at an arbitrary ZIP code on the boundary and look up its classification. We then consider each subsequent boundary ZIP in order. If the classifications of adjacent ZIPs differ, but neither is a boundary ZIP, the splitline is eccentric.

3.3.3 Discontiguity

ZIP code boundaries are stored in the shapefile as ordered lists of edges. Since the boundary-walk algorithm must be able to wrap around the edge list from the last edge to the first, each ZIP code cannot store more than one polygon. However, the dataset may include more than one polygon with the same ZIP code label. Examples of situations where this may occur include groups of islands with the same ZIP code, and ZIP codes that have a river cutting through the middle.

The solution to this problem is to symbolically perturb the dataset by re-labeling multiple polygons that have the same ZIP code. Data is read in a linear fashion, so the first polygon with the ZIP code 30304 will keep that label, but subsequent polygons will be re-labelled 30304I, 30304II, etc. Population data is only inserted into the polygon that happens to have been inserted first, and therefore happens to be labeled 30304. Since the population centroid is used to determine splitline information, this has no negative effect on the accuracy of the resulting splitline.

3.3.4 High-Degree Vertices

There are two other degeneracies that result from peculiarities of the ZCTA dataset. These must be dealt with on a case-by-case basis.

The first situation that may arise is a polygon making contact with the boundary at a single vertex. This is a special case of a situation called a high-degree vertex. The implementation is designed to deal with vertices that are incident to at most three edges: we walk the boundary of the current polygon until we run into another polygon on the boundary, then we jump to that polygon and keep walking. If the abutting polygon only touches the boundary at a single point, however, then the boundary continues on some other polygon that also touches that vertex. Since the algorithm does not know how to get to that next polygon, it must search through all edges in the candidate boundary edge list to find the one that continues from this vertex. It is important to look for an incident edge that has not yet been traversed.

Another degeneracy occurs when a polygon has only one edge on the boundary. The implementation determines the direction to walk along each polygon by looking one edge ahead and one edge behind. If the first boundary edge is also the last boundary edge, then the implementation cannot determine this direction. It must then examine the vertices; the direction selected is that which carries the walk away from the last vertex seen.

3.4 Asymptotic Analysis

Say the dataset has N ZCTAs and E edges, and we wish to form k districts. Let us first consider the topmost level of recursion. The algorithm first examines each edge to determine whether it appears in more than one polygon in order to build a list of candidate boundary edges. This takes $O(E)$ time. Next, we traverse the polygons that are incident to the boundary edges to cull for internal water features and polygon edge misalignments. Our algorithm only walks those polygons that sit on the boundary of the region. For the next step, we make two assumptions. First, we assume that approximately $\sqrt{N}$ ZCTAs lie on the boundary, and thus $\sqrt{N}$ polygons. Second, we assume that all polygons in the dataset contain an approximately constant number of edges. walking the boundary is then an $O\left(E + \sqrt{N}\right)$ operation at each level.

The final portion of the algorithm examines all candidate splitlines, that is, the set of lines connecting any pair of boundary ZIP code population centroids. Assuming there are $\sqrt{N}$ centroids, there will be $O\left(\sqrt{N}^2\right) = O(N)$ splitlines. For each splitline, we must calculate the population above and below that splitline, taking $O(N)$ time. Thus the runtime for this portion of the algorithm takes $O\left(N^2\right)$ time.

Each level of recursion must execute both of the above operations. There are $log(k)$ levels of recursion, so the expected runtime for the overall algorithm is $O\left((N^2 + E)\log k\right)$. The worst-case runtime occurs when all ZCTAs lie on the boundary of the region to be split, resulting in a runtime of $O\left((N^3 + E)\log k\right)$.

4 Results

We have tested our software on the state of Georgia. Since our implementation is proof-of-concept, we let the registered voters of Georgia approximate the population. Our software produces lists of ZIP codes by district, and generates KML output for visualization with Google Earth.

Georgia has 13 congressional districts. Our software successfully computes a shortest splitline districting for population variation tolerances of 3% and higher. Since there are only about 1000 ZIP codes in Georgia, the regions at low levels of recursion can have very few non-eccentric splitline candidates, resulting in failure to find an acceptable splitline at tighter tolerances. If better population equity is required, smaller tabulation regions should be used.

Figure 1 shows the results with a 3% tolerance, and Figure 2 shows the results with a 5% tolerance. As expected, the splitlines are noticeably longer in Figure 1, particularly in the north of the state.

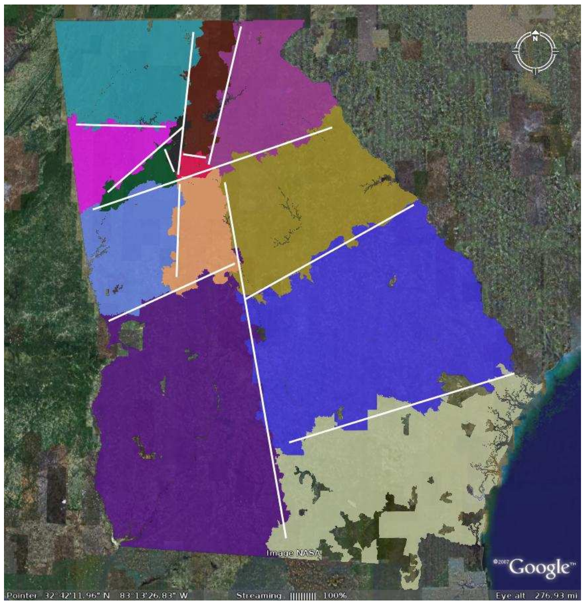

Figure 1: 13 districts attempted, 3% tolerance.

Of the approximately 3.5 million registered voters in the Georgia data, only 102,227 have a declared party affiliation. 63% of those with a declared affiliation are Democrats; consequently, Democrats appear to carry 10 of the 13 districts at 5% population tolerance. Since so few voters have formal affiliation, this result has no value as a prediction of actual outcomes.

5 Conclusion

The software we have developed successfully districts Georgia. Since there are relatively few ZIP

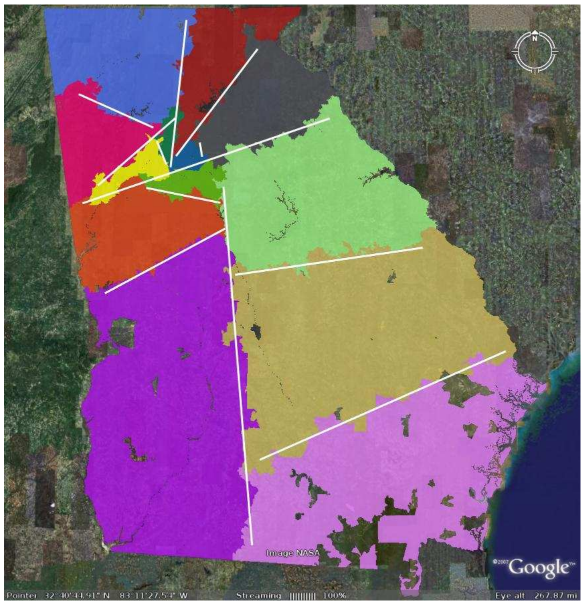

Figure 2: 13 districts, 5% tolerance.

codes per state, it runs quickly in practice: Georgia, with about 1200 ZIPs, is districted in 210 seconds on a modern workstation. Qualitatively, the results are reasonable. The Atlanta area is partitioned into small, compact districts, with larger but still simple polygons outstate.

The districts produced by using ZCTAs as the fundamental zone are certainly preferable to gerrymandering, but have some quality issues. As seen in Georgia, high equipopulation precision is not attainable with such large quanta. Moreover, the degeneracy handling required to cope with idiosyncrasies of the ZCTA data set can distort results. In particular, the eccentric splitline handling will reject most lines that closely parallel a region boundary, even if that line would be a desirable solution. Finally, ZIP boundaries themselves are arbitrary, irregular, and potentially open to political manipulation.

The natural solution is to replace ZCTAs with some other, finer tessellation of convex polygons. In the extreme, this tessellation could be the Voronoi diagram on the set of all voters. However, the expected quadratic and possible cubic running times will penalize performance severely for higher-frequency sampling. It may be possible to remove a factor of N from the runtime by building a quadtree index of voters, with each node storing the entire population descended from it.

6 Acknowledgments

We thank Andy Danner for advising our project.

References

Micah Altman. 1997. Is automation the answer: The computational complexity of automated redistricting. *Rutgers Computer and Law Technology Journal*.

Sergei Bespamyatnikh, David Kirkpatrick, and Jack Snoeyink. 1999. Generalizing ham sandwich cuts to equitable subdivisions. In *SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, pages 49–58, New York, NY, USA. ACM.

Tony H. Grubesic and Timothy C Matisziw. 2006. On the use of zip codes and zip code tabulation areas (zctas) for the spatial analysis of epidemiological data. *Int J Health Geogr.*, 5(58).

Robert E. Helbig, Patrick K. Orr, and Robert R. Roediger. 1972. Political redistricting by computer. *Commun. ACM*, 15(8):735–741.

Jorg Kalcsics, Stefan Nickel, and Michael Schroder. 2005. Towards a unified territorial design approach - applications, algorithms, and gis integration. *Journal of Geographical Systems*, 13(1):1–56.

W. Macmillan. 2001. Redistricting in a gis environment: An optimisation algorithm using switching-points. *Journal of Geographical Systems*, 3(2):167–180.

Warren D. Smith and Ivan Ryan. 2007. Gerrymandering and a cure - shortest splitline algorithm. `http://www.rangevoting.org/GerryExamples.html`.