Districting Done *Right*

Team 2099

February 15, 2022

Abstract

how do we do this front 3/4 page summary??? (need to write it)

1 Introduction

In compliance with the United States Constitution and other subsequently passed laws, the makeup House of Representatives is to closely mirror the population distribution of the country. With the exception that all states must be minimally represented by one person, seats are allocated such that the highest-population states will control a proportionately high level of power in the lower chamber. When sending multiple representatives, it is the individual state's responsibility to apportion its population into districts that, though not necessarily non-partisan, must resemble each other in total population. Malapportionnment, by enlarging a specific district to decrease its population's influence, is strictly illegal and unconstitutional.

New York, the second most populous state in the Union, elects bi-annually 29 delegates to the House, each representing his or her own congressional districts of roughly equal population. Within each state, however, there is no specific requirement of how these districts are drawn, so long as they remain in general parity regarding the number of inhabitants. Furthermore, each state is responsible for redrawing their own district lines as new census data becomes available and certain areas gain or lose relative popularity within the state. Alternatively, districts must be redrawn whenever seats are gained or lost as a result of a population shift within the country. Predictably, districting is politically a hotly contested issue, and, more pertinently, a generally difficult task to perform by hand.

Historically, districting practices, left up to the state, have allowed mis-apportionment to improperly and untruthfully shift political power. Perhaps

Figure 1: Arizona's 2nd congressional district, widely thought to be gerrymandered. (Source: NationalAtlas.gov)

the most well known incident centered around Massachusetts Governor Elbridge Gerry [1], who in the early 1800s approved a controversial districting plan which created a "salamander-shaped" district to group areas of political opposition under one national representative. This outraged the opposing party, who alleged that he was now given unfair advantage. Retroactively named, "gerrymandering" refers to any districting action which, via shape or size, intentionally seeks to dilute the relative power of political opposition. See figure 1 for an example of a currently existing, especially odd-shaped district.

Amongst political scientists, there does not exist a consensus regarding the effects gerrymandering [2]. While there is mathematical evidence that suggests gerrymandering can give an advantage to a majority party or philosophy [3], it is claimed that the effects of a successful gerrymander by use of shape in a state of high contention (and thus a state where a gerrymander would have the most effect), any gain would be relatively short lived and would eventually be negligible due to changing voting trends [4]. On the other hand, gerrymandering by skewing size can have tremendous effects to suppress the political influence of a group that is tightly contained geographically, such as the African-American population in the low-country of South Carolina.

Even within this Mathematical Contest in Modeling team there are differences in opinion with regard to whether or not districts should be determined by legislative procedures within a state or an impartial computer algorithm. Thus the motivation of this paper is not simply to develop such an algorithm, but also to consider whether or not the algorithm, if successful, should be employed at all.

This paper will first develop a mathematical model for the districting problem conceptually as expanding "bubbles" in a fixed volume, with each bubble containing equal mass (analogous to population). We then develop an algorithm based on this model, where the bubbles and fixed volume translate to districts and state borders respectively. Literature suggests that our algorithm fits well into the multi-start constructive heuristic classification, but it also contains greedy and probabilistic elements [5] [6]. While there exist many heuristic algorithms for solving the districting problem [7] [8] [9], our algorithm was conceived independently and, according to our research, contains nuances not seen in existing methods. Finally, the implementation is applied to the state of New York, such that the results may be compared actual districts.

2 Model Fundamentals

2.1 Solution Criteria

As expected, over the past few decades many cases related to congressional districting have been heard by the United States Supreme Court. /citewatson Verdicts and decisions arising from these trials include a strict consideration of malapportionment as illegal: districts within a state must contain near-equal population, but no other characteristics have been clearly defined. While the courts do require that districts should be "reasonably contiguous and compact," this ambiguous statement is often ignored, as it is not immediately clear how to define "reasonably" contiguous. As a manner of practice, it is expected that districts will be drawn in good faith, such that political gain is not a consideration in the placement of the individual boundaries.

Upon careful consideration, we make the assumption that a "good" and "fair" districting scheme accounts for three essential criteria: equal population amongst districts, contiguity, and compactness. The first two criteria are simple and non-ambiguous, while for the last criterion we define the following benchmark: $min(\frac{perimeter}{area})$. For a constant area, the shape that minimizes this simple expression is a circle, whereas a district shape subject to gerrymandering would have a much greater ratio. We define population equality and contiguity as our constraints, while our compactness benchmark defines the quality of a given solution.

Note that missing from our criteria are population demographics, which are frequently said to be used by gerrymandering politicians. The absense of such traits yields an unbiased districting algorithm.

2.2 Conceptual Model

Given our solution criteria specified above, an intuitive, analogous system was developed that featured expanding bubbles. Such shapes are obviously contiguous and attain the best "score" in the compactness benchmark. Thus, the political districting problem can be modelled as *bubbles* that are expanding inside a fixed area in the shape of the state border. The area

Figure 2: Two bubbles in a non-uniform fixed area, with the bottom having a greater density than the top. For all images, each bubble encompases equally sized populations. (a) Both bubbles are the same size. (b), (c) The top bubble expands more quickly than the bottom bubble, since the top of the fixed area is less dense. (d) The bubbles begin to deform due to collisions with the fixed area's boundary and each other. (e) The steady state when both bubbles have expanded to their maximum size.

of each bubble represents a single district, and its mass represents the contained population. As the bubbles expand, they not only try to retain their shape, but they also inadvertently deform when they reach a barrier, obstacle, or other bubble. Although all bubbles are artificially constrained to hold at any point the same mass, they will eventually cover the entire area of the state while retaining a reasonably contiguous and convex shape.

Now consider the case where a bubble's rate of expansion is determined by the number of people it takes in. In an area modeled as having a uniform density, these expanding bubbles would grow at the same rate and, in their steady state, partition the area into equally sized regions. But a realistic state, of course, does not have a uniform population density. If we weight the interoir of the state by population and keep constant the number of people that each bubble encompasses, then the bubbles would grow at varying rates, such that low density areas encourage rapid growth and high density areas encourage slow growth. In their steady state, the bubbles would partition the area into differently sized regions while maintaining the requirement that they all contain approximately the same number of people. This concept is illustrated in figure 2. The idea that each bubble should 1) grow such that the rate of people absorbed by the bubble is constant, and 2) try to retain its bubble-like shape, is what we attempt to enforce in our algorithm.

3 Details of the Districting Algorithm

Although, initially, it may seem plausible to run a physics-based simulation of several bubbles within a somewhat continuous environment, bubble deformation is extremely difficult to simulate on a large scale. Thus transforming the conceptual model into an algorithm that can be realistically implemented was necessary. The expanding bubble concept is mostly maintained as determined above, but several computational details and support structures were added as described in the following sections.

3.1 Design for "Expanding Bubbles"

The first step in our algorithm is to create an *environment matrix*, a two dimensional matrix containing data about the state to be districted, for the bubbles to expand in. We use a matrix of integers to define the interior area of a state, with other locations in the matrix being "off limits." Within the state's interior, each matrix element is assigned a number which represents the population within an equivalent square area in the state being modeled. Obviously, a higher number corresponds to a greater population within that unit square. An example of a simplified, theoretical state represented as a coarse ten by ten environment matrix is presented in figure 3a.

A second matrix, which we call the *bubble matrix*, is used to store the locations of each bubble by means of an index number (x) ranging between one and the desired number of districts. Each bubble is initialized as a single entry in this new matrix. The start location of each bubble is determined probabilistically via the values in the environment matrix, as shown in equation 1. $P(i,j)$ is the probability that bubble is placed at location (i,j), $R_{i,j}$ is the value of the environment matrix at location (i,j), S is the set of locations that lie within the interior area of the state, and n is a constant. A larger value of n corresponds with greater skewing of probability towards more populous areas, and the case $n = 0$ corresponds with every $(i,j) \in S$ having an equal liklihood of being picked. We desire most bubbles' starting locations to be near population centers because we make the assumption that these are good approximations for where most bubbles will end up in an optimal solution.

$$P(i,j) = \begin{cases} \dfrac{R_{i,j}^n}{\displaystyle\sum_{(i,j)\in S} R_{i,j}^n} & \text{for } (i,j) \in S \\ \\ 0 & \text{for } (i,j) \notin S \end{cases} \tag{1}$$

Following initialization, the bubble containing the fewest people begins to expand itself. The size of a bubble is defined as $F(x) = \sum_{(i,j)\in B_x} R_{i,j}$, where B_x is the set of locations in the environment matrix (which is determined by the values in the bubble matrix) that belong to the bubble with index x. Let the smallest bubble be x_{min} such that $F(x_{min}) = minF(x)$ over all x. The bubble x_{min} is expanded until it is strictly larger than all

Figure 3: Matricies from a 10x10 districting problem. (a) The environment matrix. (b) A good result of the algorithm. (c) Another good result of the algorithm, but a different scheme. (d) A bad result of the algorithm.

others, thereby giving the dubious role of "smallest bubble" to another. The process repeats until all $(i, j) \in S$ are associated with some bubble.

The primary method by which a bubble grows is to absorb an empty adjacent location. Since there are generally numerous possible locations to choose from, relative probabilities are assigned to each potential expansion spot such that the expanding bubble remains as compact as possible (in open space, this equates to a roughly circular shape). The most probably location for expansion will be that closest to the bubble's current center of area, defined as the average position (without regards to population) of every discrete location contained within the bubble. In descending order, less appealing expansion spots are assigned probabilities that decay in a constant manner. Currently, any arbitrary expansion spot is 50% more likely to be chosen than its slightly more distant peer.

In the event that a bubble is not adjacent to any empty locations, but it still needs to expand, it performs what the model colloquially refers to as a *steal*. More specifically, the bubble will assign to itself a location from a neighboring bubble, thus gaining possession of the enclosed population. This growing method is also a probabilistic step that collects possible expansion spots to which relative probabilities are assigned. Unlike the previous step, however, consideration is given to the original possessor of any would-be expansion spot. Specifically, all possible expansion spots are ordered such that steals are most likely to occur from the largest neighbors. On multi-pixel borders, the expansions spots are further ordered by proximity to the expanding bubble's center of area.

For the algorithm to end at what we consider a solution, two criteria must be met: all locations in the interior of the state are associated with some bubble, and the bubbles have roughly equal size. All bubbles will first expand until the whole of the state is filled (with perhaps some stealing taking place), To balance population sizes, they should continue to steal from each other by again continuously selecting the smallest bubble/district and expanding it. Note that at this point in the algorithm, a given district is expanded until it is 3% above the total population of the state divided by the number of districts.

The algorithm boosts small districts to (just over) the desired population size until each is nearly uniform. For the purposes of this condition, "nearly uniform" is defined to be the range $[-8\%, +12\%]$ of the total population divided by the total number of districts. The range is skewed towards values greater than the optimum due to algorithm design: although the system can force individual bubble/district population to rise, it must wait for it to fall. This fall-time can be extremely lengthy, truly representing the majority of the computation in the algorithm.

The algorithm, in its original form, is *heuristic* in that, even though the optimality of solutions cannot be guaranteed, it should still find good and feasible solutions. It is also *constructive* because it builds a solution from the ground-up and terminates upon discovery of one that meets all of the criteria and requirements explained previously. Furthermore, it exhibits *multistart* behavior since each time the algorithm is run, the start-locations of each bubble are determined probabilistically.

To have a reasonable chance of approaching an optimal solution for a given data set, the algorithm should be run multiple times. The results should be compared, and the best starting locations noted. To expand upon the previous point, the model is certainly *probabilistic*, as it uses probability at virtually every step. Finally, the model is somewhat *greedy* in nature because at every step, it strives to expand the bubbles in the best, most compact way possible (even though probabilities allow for nonoptimal expansions). The tyoes of algorithms mentioned above are well studied, but research has not found another algorithm with precisely this combination of characteristics.

3.2 Finding a Reasonably Good Solution

The result of the algorithm on a simplified case is shown in figure 3. In this simulation, three districts are planned for a theoretical state that fits into a $10x10$ grid. The state has two moderate population centers, one in the top left corner and one slightly below it (look where the values in figure 3a are large). Rarely, the algorithm might make a poor decision while expanding, as illustrated by the discontiguous solution in figure 3d. However, most commonly, the algorithm converges to a continuous and compact solution similar to the "peace sign" arrangement in figure 3c. Occasionally, the solution takes a different shape, like the "horizontal bar" arrangement in figure 3b. It is obvious that we would want to throw out solutions like 3d, but less clear is the deciding between 3b and 3c, both of which appear to be good solutions.

It becomes necessary, then, to develop some sort of quality measure. Of the three solution criteria described in section 2.1, only population equality is completely garuanteed to be satisfied. Therefore it makes sense that the benchmark we should use would test contiguity and compactness. We consider two: *circular* (equation 1111111) and *rectangular* (equation 22222222). The former is perimeter divided by area, where the optimal shape is a circle. The latter is area of the bubble divided by area contained within the smallest possible rectangular cover, where the optimal shape is a rectangle. In both cases, a greater value implies a more optimal solution. See figure *****need to make figure***** for an illustration.

Returning to the results of the $10x10$ case, figure *****need to make table of values***** shows the numerical result of both quality measures applied to figures 3b-d. While both benchmarks discount 3d as being a good solution, they don't come to a consensus with regard to 3b and 3c. The rectangular method favors a very rectangular solution in 3b, while the circular method favors an approximately round solution in 3c. So while it's still not clear as to which kind of solution would be better, we can at least discount the pathologically bad ones.

*****equations 111111111 and 222222222 go here-ish*****

3.3 Implementation Issues

While the algorithm runs well in the $10x10$ case, it runs significantly slower with larger environment matricies. Because of this slow runtime with any decently detailed dataset, changes to certain aspects of our algorithm were needed such that it would run faster. Previously, the smallest bubble, x_{min}, expanded until it surpassed the size of the next smallest bubble. In the final version of the algorithm, it expands until it is strictly greater than the largest of bubbles. Because the bubbles are now no longer expanding at approximately the same rate, some irregular shapes can occur. Compactness and contiguity are thus lost in the revised algorithm, which by design did not happen in the original algorithm. As a result, a suplimentary compactness algorithm should be designed to account for this tradeoff, even though time constraints do not provide for this.

4 Redistricting New York State: A Case Study

4.1 Details of Implementation on Actual Data

To test the algorithm developed in the previous sections, unbiased congressional districts are drawn, twenty-nine in all, in New York State. Similar to how the state government would proceed, the algorithm uses the latest Census data (in this case, from 2000) in determining how district lines should be drawn. This data, freely available from the New York State Data Center [10], is in the form of a *shapefile*, a commonly used geographical file format devised by the Environmental Systems Research Institute [11]. The shapefile is basically a set of connected polygons that represent *census tracts*, which are county subdivisions that the United States Census Bureau use to complete the decannual census [12]. The population of each census tract usually varies between 1,500 and 8,000, having been designed for homogeneity with respect to certain population characteristics when first delineated. Each polygon in the shapefile contains data regarding its total population and area, so it was fairly simple to create a new dataset of population density $\left(\frac{totalpopulation}{tractarea}\right)$. Using a program called Thuban [13], population density was rendered into a color-gradiented graphic. By outputting this image and subsequently converting color information into a matrix of integers, we create an environment matrix (see section 3.1) which represents the state of New York.

Upon careful study of the data, it becomes painfully clear that the problem of districting New York is not at all an easy one. Of the more than nineteen million people in New York state, Nearly half (eight million) reside in New York City, which comprises less than 1% of the total land area of the state [14]. Additionally, this huge spike in population is concentrated at a geographical bottleneck, making it difficult for bubbles to grow. For instance, if there were a bubble at the end of Long Island needing to expand, it would have to steal from a nearby New York City bubble, creating an inefficient chain reaction ending upstate. Furthermore, the addition of just a single pixel to a bubble in New York City could drastically alter a bubble's size, wrecking havoc on the algorithm's ability to find a convergent solution.

4.2 Results of the Districting Algorithm on New York State

In agreement with our prediction of the unwieldy data due to New York State, we have found complications near New York City arising routinely in the running of our code. Due to the large amount of stealing that takes place, which cuts into the state with the higher population, our initial goals concerning the compactness and contiguity of the districts are sacrificed for the equality of population. One may ask oneself if this is better or worse than gerrymandering, due to many of the visual similarities between the resulting districts. One might also conclude that our algorithm results in districts that are worse off than the gerrymandered. This is especially evident considering the large distance between discontiguous portions of some districts. Figures ****************** show some of the better and worse results of our algorithm.

5 Conclusion

In this paper we have combined the qualities of several heuristic algorithms into one for the purpose of political redistricting without the influence and interference of legislation. While tradeoffs were made in favor of reasonably fast code, the algorithm itself is still good. The following section details ways this districting method might be improved in terms of computational complexity without having to sacrifice the essence of the algorithm.

5.1 Further Work

It is inevitable and unfortunate that, within the span of a ninety-two hour competition, many ideas are stumbled upon with no time to implement them. The first of two is an improvement upon our current algorithm involving environment matricies of varying size, representing the same state. The algorithm is run on the smallest version, and the result is used as an initial guess for the next smallest version. The chain continues until the problem is solved at the desired high quality resolution. This textitmultiresolution approach would decrease computing time without much, if any, loss of accuracy. Additionally, the algorithm would not lose its probabilistic nature. In fact, if only the best small resolution results become initial guesses for higher resolution problems, then the algorithm takes on a *genetic* quality. That is, it emulates biological evolution as only the good results are kept, leading to more optimum solutions.

Another avenue of further development involves extending the bubble idea to a discrete problem. Our implementation of our expanding bubble algorithm utilizes a discrete approximation of a continuous $\mathbb{R}^2 \longrightarrow \mathbb{R}$ function of population density. It would probably be far more accurate and computationally faster to instead represent each census tract as a single point with a (x, y) location and a value equal to the population at that point. Much of the algorithm would remain the same, in that we still use bubbles, and they are still placed probabilistically. But rather than actual circular bubbles one can see on a map, the growth step would be of a more computational nature: simply search for the point whose coordinates are mathematically closest. This is effectively the same as having a bubble radius and growing it to the

the nearest neighbor location. Probably this variation's biggest advantage is that it literally assigns census tracts to districts, so it has a far better correlation to real life than our current algorithm.

5.2 Final Thoughts

Upon final analysis, we have come to the conclusion that our results are unacceptable for practical use. This is not to say that our algorithm is fundamentally flawed, or wasn't fully thought through. On the contrary, these results show that the algorithm wasn't able to take the form which it's authors envisioned. Implementing a discrete version of the model would most certainly yield the needed speed, and would not have the same ill effects our algorithm produced. This is certainly an area of research that should be pursued.

References

[1] Hatfield, Mark O, with the Senate Historical Office (1997). *Vice Presidents of the United States, 1789- 1993.* U.S. Government Printing Office: Washington, DC. pp. 63-68.

[2] McDonald, Michael P (2004). A Comparative Analysis of Redistricting Institutions in the United States, 2001-2002. *State Politics & Policy Quarterly.* 4(4):371.

[3] Sherstyuk, Katerina (1998). How to gerrymander: A formal analysis. *Public Choice.* 95:27-49.

[4] Morton, Rebecca B (2004). *Analyzing Elections.* forthcoming, W.W. Norton. pp. 135-168.

[5] Silver, E A (2004). An overview of heuristic solution methods. *Journal of the Operational Research Society.* 55:936-956.

[6] Rardin, Ronald L (1998). *Optimization in Operations Research.* Prentice Hall: Upper Saddle River, NJ. pp. 680-705.

[7] Catalano, M S Fiorenzo; Malucelli, F (2005). Randomized Heuristic Schemes For The Set Covering Problem. *ORSA Journal on Computing.* 17(1):10.

[8] Bozkaya, Burcin; Erkut, Erhan; Laporte, Gilbert (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European Journal of Operational Research*. 144:12-26.

[9] Mehrotra, Anuj; Johnson, Ellis L; Nemhauser, George L (1998). An optimization based heuristic for political districting. *Management Science*. 44(8):1100.

[10] New York State Data Center (2000). Content: 2000 Census data and geographic files for New York state at the Census tract level. [online]. Available from: http://205.232.252.81/selectFiles.asp. Accessed 2007 Feb 10.

[11] Environmental Systems Research Institute, Inc. (1998). *ESRI Shapefile Technical Description*. [online]. Available from: http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf. Accessed 2007 Feb 12.

[12] U.S. Census Bureau, Geography Division (2000). *Census Tracts and Block Numbering Areas*. [online]. Available from: http://www.census.gov/geo/www/cen_tract.html. Accessed 2007 Feb 12.

[13] Thuban Project Team (2004). *Thuban 1.0 Documentation*. [online]. Available from: http://thuban.intevation.org/documentation.html. Accessed 2007 Feb 12.

[14] U.S. Census Bureau (2007). *State and County QuickFacts*. [online]. Available from: http://quickfacts.census.gov/qfd/states/36/3651000.html. Accessed 2007 Feb 12.