Kyotaro Hemmi, Peter Mannisto, and Ting-You Wang
University of Washington

GETTING THE VOTES TO COUNT: A GENETIC ALGORITHM FOR NON-PARTISAN LEGISLATIVE DISTRICTING

Abstract.

For decades, district gerrymandering has given incumbents an unfair advantage over their opponents, resulting in artificially high reelection rates. This gerrymandering has become so severe that Iowa [1] has implemented a non-partisan district-drawing method, and other states are sure to follow. Such efforts may have many different objectives in creating districts. In this paper we describe a redistricting algorithm which is flexible enough to accommodate a wide variety of district criteria. We use a genetic algorithm, wherein an arbitrary user-selected number of districts compete for population on a map of the state region. By adjusting the probability formulae for district reassignment events, we can tailor our algorithm to create districts satisfying a wide range of criteria. This algorithm is then tested on the state of New York, by requiring it to produce district shapes that maximize the interior regions of the districts in proportion to their boundaries. The resulting district divisions produced by the algorithm clearly exhibit this property.

After presenting the results of our tests, we will discuss the merits and defects of our chosen algorithm, along with its possibility for extension.

Date: February 11, 2007.

Contents

1. Introduction

Gerrymandering is one of the oldest and most successful methods of disenfranchisement available to Congress. This much-maligned practice is essential to maintaining the 98% re-election rate currently enjoyed by incumbents[1] [5]. Obviously, many people aren't happy with the status quo, as it can render an individual's vote meaningless as his representation is determined not by popular sentiment, but by the incumbents drawing up the districts. What is needed is a new method for determining legislative districts, one out of the hands of incumbents.

What should an ideal district-drawing algorithm accomplish? This is certainly a vexing political question. Should it take the truly unbiased approach and ignore any and all political considerations? We can just randomly create districts, blind to both political ambitions and political fairness. Others may feel that steps should be taken to ensure that elected representatives fairly represent their constituents. Should we set up the districts so that a 60% Republican state will likely have 60% of its districts be majority Republican? Perhaps this just returns us to a different type of gerrymandering, done for the cause of 'fair representation', but gerrymandering nonetheless.

This dilemma can only be solved by political scientists, not mathematicians. However, a mathematician should be able to create a district-drawing algorithm which can accomodate certain 'preferences' in drawing districts, whatever they may be. It is then up to social scientists to ascertain what these preferences should be. Our goal is to produce just such an algorithm. We will then test the algorithm by having it select districts for New York that are 'geographically simple' in a specific sense to be defined later.

[1]Average re-election rate of House incumbents over election cycles 2000, 2002, and 2004.

2. Specifications

We require our district-drawing algorithm to satisfy the following criteria:

2.1. **Population Specifications.**

In the 1962 decision of Baker v. Carr, the US Supreme Court asserted the ability to decide on the constitutionality of state district divisions [2]. Although the Supreme Court has not specified the exact terms of what is acceptable, past rulings have suggested the following: "congressional redistricting plans with overall range percentage variances of up to .73 percent have been approved based upon identifiable state objectives" [1]. Here, the overall range percentage variance is defined by the maximum difference in population between any pair of districts, divided by the 'ideal' district population (which is just the population of the state divided by the number of districts). We would like to note, however, that a lower deviation percentage of 0.69% has been rejected in the past [1]. Pending further legal clarification, we set a range variance of less than or equal to 0.7 percent as our goal for population range variance.

2.2. **Geographic Simplicity.**

We will require our districts to satisfy two criteria. The first requirement is absolute: each district must be connected. We will later describe our method for enforcing this. Our second requirement will be to have each district contain as many interior points as possible. Here, an interior point means a grid point which does not neighbor a grid point of a different region on either the east, west, north, or south. As with any notion of geometric simplicity, this one is motivated by aesthetic reasons: intuitively, discretized versions of squares and circles contain a large number of interior grid points.

The recognition that others may not agree with our particular definition of simplicity is an important reason why we wish our algorithm to accommodate different district division criteria easily.

Let us be more specific about what a district-drawing algorithm should do. We will be given a region (which represents the state map), along with a population density function defined on the state. In our algorithm the region and density function will be discretized, but a priori one thinks of them as approximately continuous. We will discuss the validity of the discretization later.

In addition, the aforementioned social scientists may find other variables to be of interest in drawing up districts. There are many ways in which such data can be naturally presented. For example, we may wish to have our districts follow county lines when possible. In this case, county lines should be included in our state boundary map. On the other hand, information about political leanings is naturally presented as a density function ρ_D on the region (i.e., for each point x on the state, $\rho_D(x)dA$ is the percentage of people living in differential area dA surrounding x who are Democrats, with analogous functions ρ_R, ρ_I for Republicans and Independents). We will later show how these possibilities and others can be

incorporated into our general method.

3. Our Method

In broad overview, our algorithm for dividing up the district is a genetic algorithm, wherein the districts compete for population on the state map. The algorithm takes as input an integer k (the number of districts in which to divide the state) and a representation of the population density as a function of location within the state. This information is given as a rectangular matrix (or grid), with each grid point corresponding to a small square on the map. Each grid point is then assigned the population contained within the corresponding square region of the state. The grid as a whole is rectangular, so it does not match up with the state's borders exactly. Any spot on the grid outside of the state's borders is assigned zero population density, and will be declared 'out of bounds' in the simulation to come. A visual representation of the resulting population density function is seen in the figure on page 8. In our specific example for New York state, this grid is 67 squares wide and 46 squares tall.

3.1. **Source for data.** Our data for population density was obtained from [4], in particular the map seen below. We gridded the map by hand, and then approximated the density of each square based on the scale given. This approach to acquiring numerical density data certainly isn't ideal, but there does not seem to be a readily available digital source for gridded population data for the state of New York.

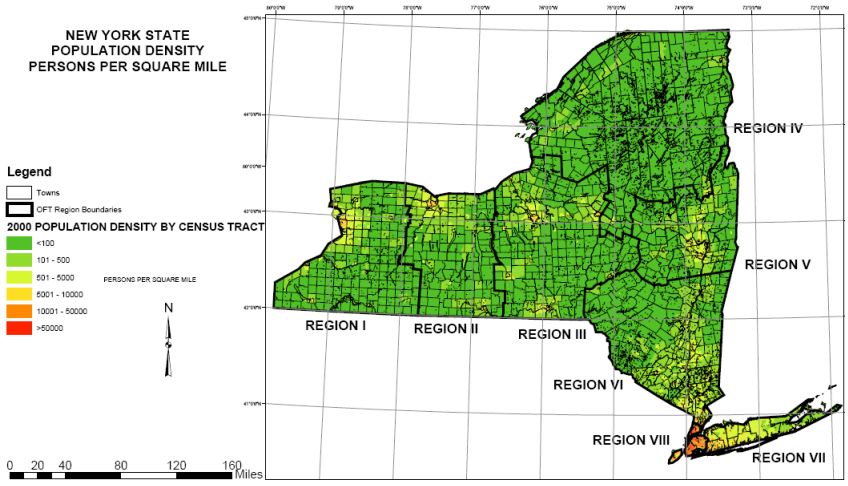

Figure 1. Source for population density figures, New York State

Once the data has been fed into the program, the simulation can begin. We first initialize the regions by randomly picking k grid points within the state boundaries, and assigning each of these k points to a district, so that at the beginning, each

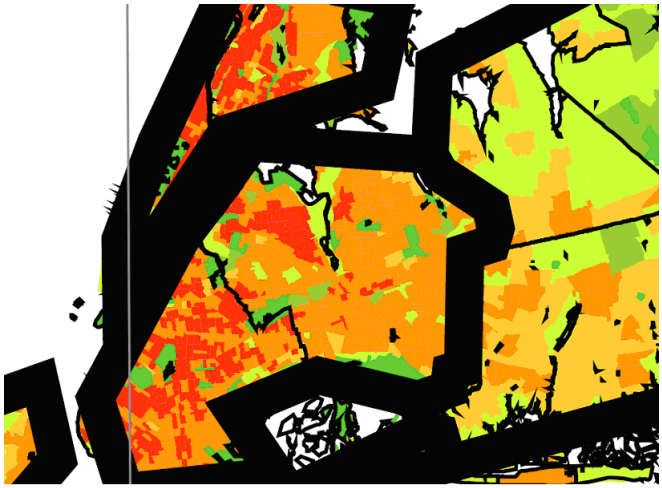

FIGURE 2. Source for population density figures, New York City

district consists of 1 grid point. The rest of the map is not initially assigned a district. Now we loop through all grid points on the map, performing the following tasks for each point:

- If the grid point does not belong to a district, go on to the next grid point.
- If it does belong to a district, say district A, consider each of its neighboring grid points to the west, east, north, and south. If a neighboring grid point does not belong to a district, assign it to district A. This way unused grid points on the map are quickly taken up.
- If a neighboring grid point does belong to a district, say district B, calculate the total population of districts A and B, as well as the potential change in total number of interior grid points if the neighboring grid point were to be re-assigned to district B. Calculate a probability for re-assignment based on these two numbers, and 'flip a coin' based on this probability to determine whether or not the neighboring grid point gets re-assigned to district A. See Appendix 1 for the actual probability formula used.

After the above steps have been performed for each grid point, repeat the whole loop again and again, until (hopefully) the population numbers stabilize. In reality, our discretization prevents the populations of the regions from becoming exactly equal, so we must introduce a cut-off. In our algorithm, this is done as follows: when calculating a potential re-assignment of a grid point (x, y) from district A to district B, look at the relative population difference ΔP_{AB} between regions A and B, given by $\Delta P_{AB} = \frac{P_A - P_B}{(P_A + P_B)}$, where P_A, P_B are the current populations of districts A and B. If this number is below 0.01, our probability formula gives zero probability for re-assignment. This way, theoretically, our simulation should stop when the relative population differences of neighboring districts is smaller than 2%. Recall that our goal for range variance was 0.7%, so we are essentially giving up on this goal. This was a result of recognizing that our discretization was simply not fine

enough to allow for fine-tuning the relative populations to a fraction of a percent. Even worse, note that we only measure differences between neighboring districts, whereas federal guidelines set a cap on deviations between any two districts in a state. We will discuss population differences in more detail after getting a look at some real data provided by our algorithm.

There is one final step to our algorithm: after the districts have stabilized, if any districts are seen to be disconnected, all disconnected pieces of the district except the largest are manually reassigned to the neighboring regions, and the simulation is restarted. The emergence of disconnected regions was a rarity (as intended) when the probabilities were weighted based on interior nodes, so this should only be considered an emergency fix to satisfy our requirement for connected districts.

3.2. **Other possibilities for district-drawing.**

When dicussing this problem, two methods of drawing districts immediately come to mind. One is the genetic algorithm outlined above. The other is best exemplified by the split-line method [3], which can be summarized as follows: suppose for simplicity that we are wishing to divide our state into 2^n districts, for some integer n. We are given a (continuous) population density function on our state, and we determine the shortest straight line cutting the state in half, such that each half of the state contains the same population as the other. We then continue recursively on each divided piece of the state. An example of districts drawn by this method is seen below. The resulting districts do have an obvious geometric simplicity to them.

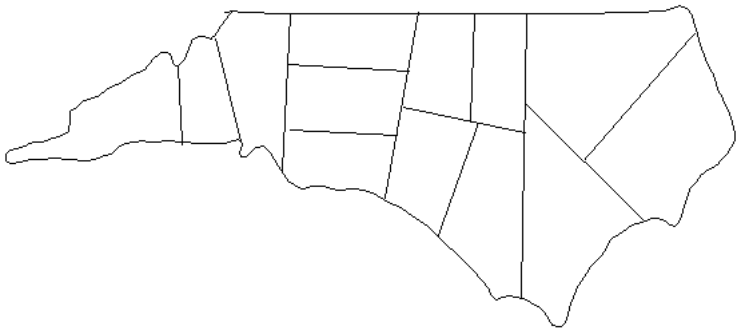

FIGURE 3. North Carolina districts drawn by split-line method

The problem with the split-line method and its relatives is that they only allow for one definition of what makes a 'good' division into districts, and that definition is a rather abstract one, as is any criterion based on geometric simplicity. Is this what voters want? Do they want their districts to be divided in the way which makes the fewest (and shortest) straight-line cuts through the state? Again, our feeling is that it should not be up to mathematicians to make the criteria for a 'good' political subdivision, but that it is the task of social and political scientists. The advantage of our method is that it easily accomodates almost any criterion

for district-drawing, as we can adjust the probability for re-assignment of districts above to depend on almost any relevant variable. We believe this flexibility is essential to any practical district-drawing algorithm.

That is not to say that our method is perfect, of course. Whenever a continuous data set (population density and the geography of the state are best approximated by continuous functions) is discretized one must question the effectiveness and accuracy of the discretization. This question will be taken up after we have seen the algorithm produce results. There is also a philosophical concern: there is no consistent way to determine good probability functions without a bit of trial and error. We try a probability function, look at the districts produced, and adjust accordingly. This introduces a human element into the district-drawing, which is what we were originally trying to replace. There is probably no way out of this defect: in this problem we must choose between an inflexible, determinististic algorithm or a flexible algorithm with some human judgement required. We are committed to the latter option.

4. Data

As mentioned above, the inputted population density data for New York was entered by hand. The resulting density plots are shown below in Figures 3 and 4. As a rough scale, on the zoomed-out map, white corresponds to less than 5 people per square mile, while black corresponds to greater than 10,000 people per square mile, with intermediate steps of gray-scale in between. On the zoomed-in map of New York City, white corresponds to less than 500 people per square mile, and black corresponds to greater than 30,000 people per square mile.

NOTE: Because of the extremely high population density of the New York City area, we must divide our problem into two parts. Based on official 2000 Census data, we determined that 13 legislative districts should be in what we informally define to be the New York City area (in other words, the zoomed-in region shown in Figure 5). The other 16 fill up the rest of New York State. We always run the algorithm twice: once on the greater New York State area, dividing this region into sixteen legislative districts, and once on the New York City area, dividing this region into thirteen legislative districts.

Figure 6 depicts a district division based on probability formula (A.1) (which only involves population differences). Comparison to subsequent figures allows us to gauge the effect of the multiplicative factor involving interior grid points (seen in Equation A.2). Each number corresponds to a different district. As mentioned in the note, the figure contains 16 districts, which is the number of districts to be placed outside the New York City area. Below the figure are shown the populations of each district.

NOTE: The population numbers are not to scale in these figures, and should only be used as relative comparisons between different districts. To convert these populations to actual population numbers, multiply them by 50.

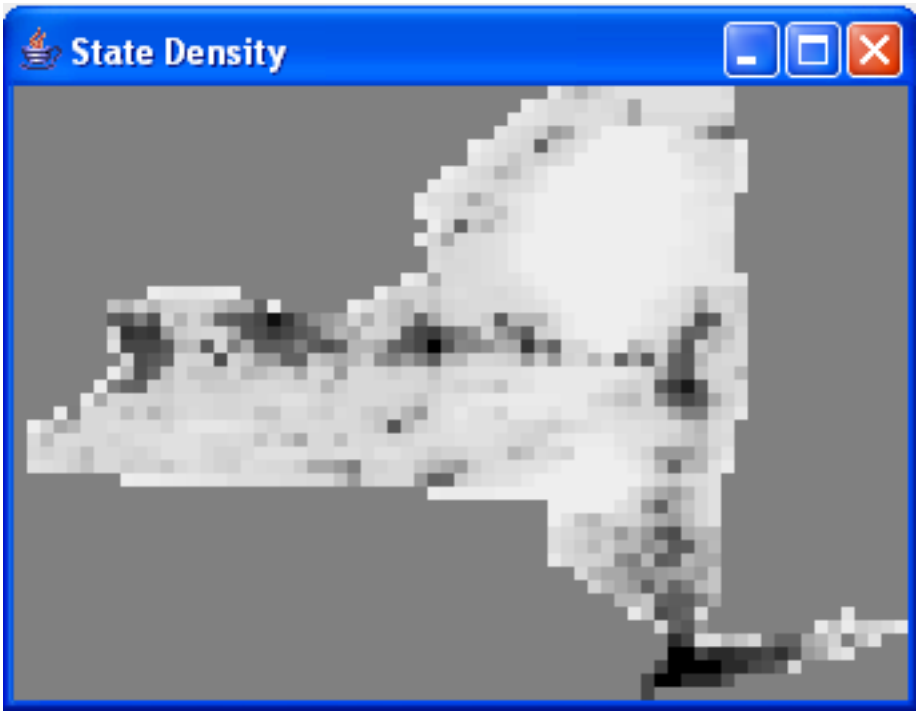

FIGURE 4. New York State Population Density

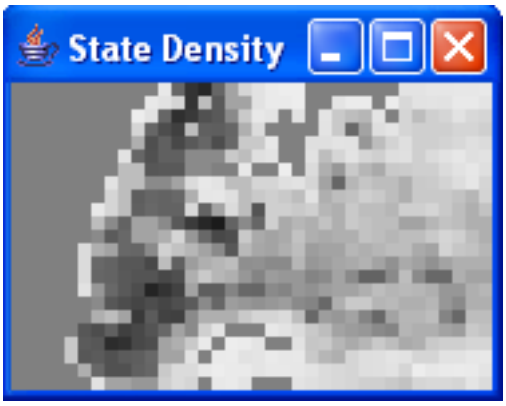

FIGURE 5. New York City Population Density

The maximum range variance in populations is 3.3%, well above our goal of 0.7%. Unfortunately, our discretization simply doesn't appear good enough to meet this criterion.

Figures 7-10 displayed on the succeeding pages show two sets of representative

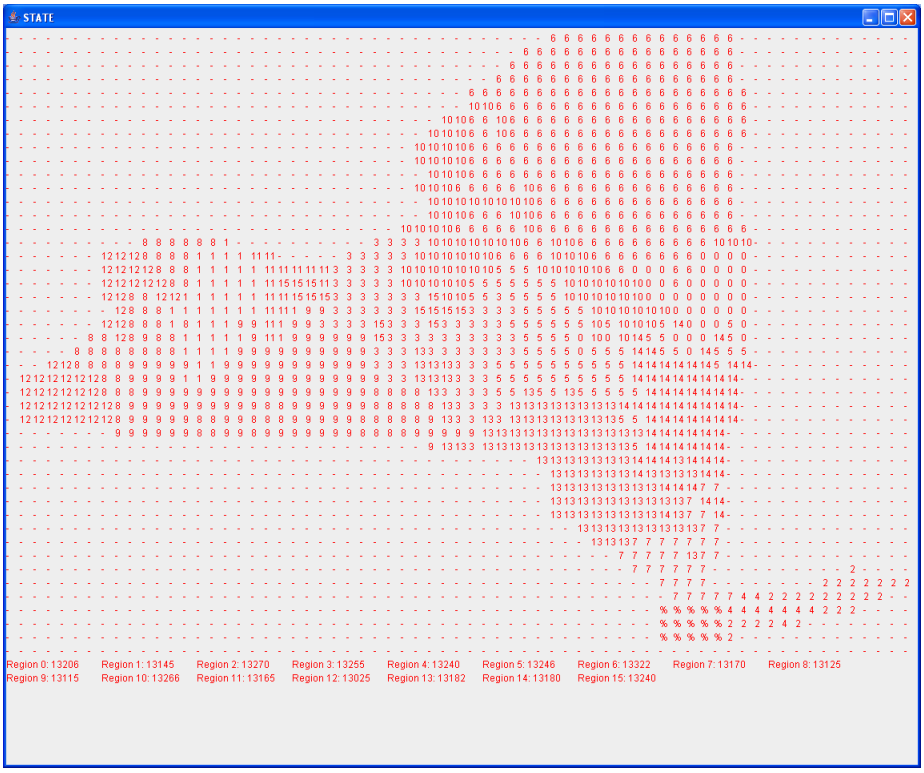

FIGURE 6. New York State Districts, no interior point weighting

districts drawn using the weighting based on interior grid points. It is somewhat coincidental that these two district divisions look very similar; in fact, we have observed several qualitatively different overall patterns to emerge as stable district configurations. However, every stable configuration is built out of relatively few recurring simple geometric shapes, almost all of which are represented on at least one of the figuress. All of these shapes tend to have a large number of interior grid points, and the contrast between these shapes and the ones in the Figure 6 is immediately apparent.

The range variance numbers for these district divisions are similar to the numbers above, approaching an absolute range variance of 4%. Again, this falls well short of our stated goal of 0.7%. We must improve our discretization to improve these numbers.

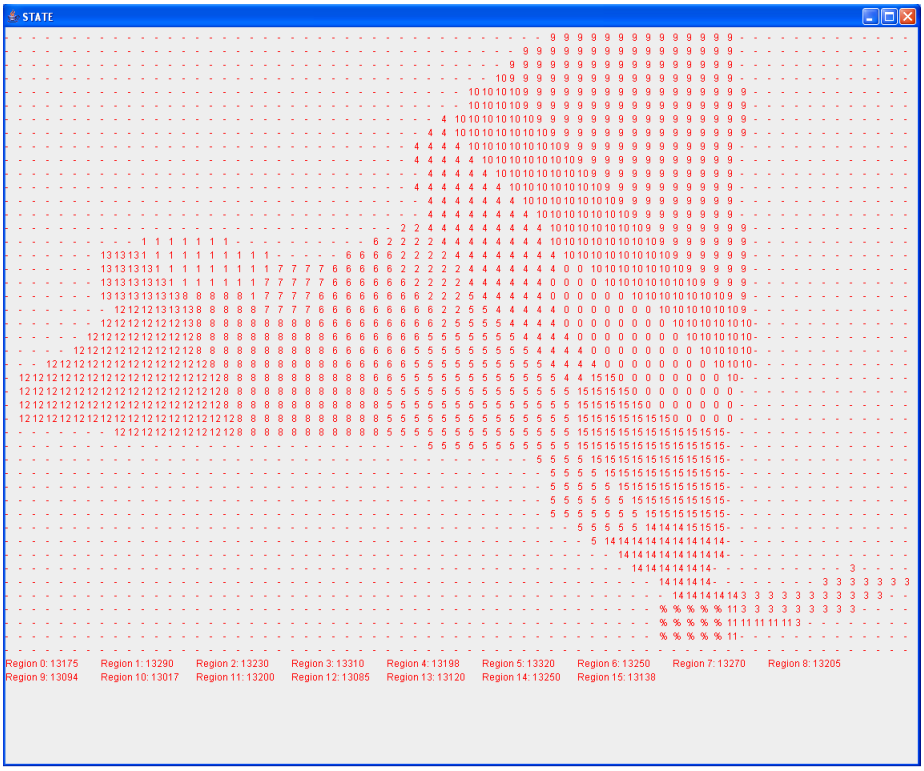

FIGURE 7. New York State Sample 1

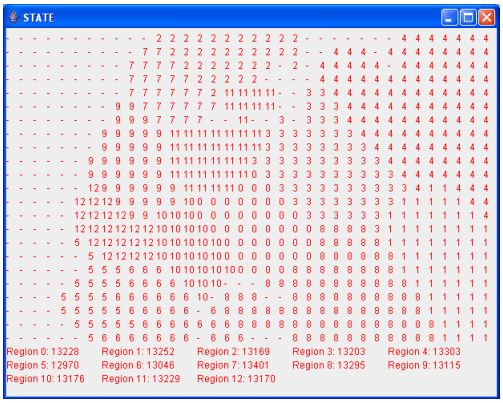

FIGURE 8. New York City Sample 1

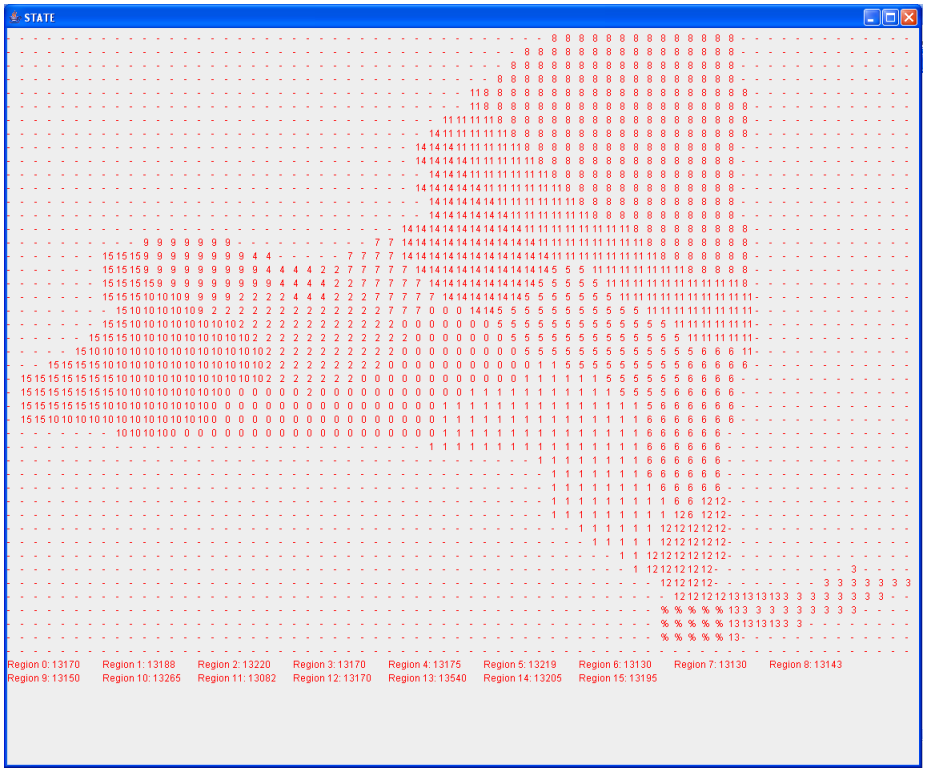

FIGURE 9. New York State Sample 2

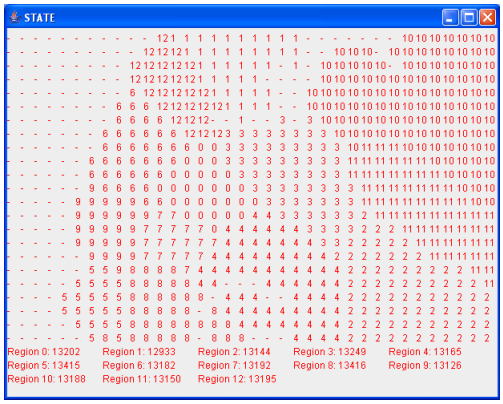

FIGURE 10. New York City Sample 2

5. Evaluation of our Method

5.1. **Defects.**

There are two difficulties inherent to our algorithm, both of which have been alluded to several times in this paper. The first is that we discretized an essentially continuous problem. The range variance of the population districts suffers as a result, with the stated goal of 0.7% range variance being unattainable. The only solution to this problem is to use finer gridlines. We are not limited by computer performance in this regard, as our algorithm typically terminates within 40 seconds. Even this is artificially low, as we graphically update the progress regularly and use a sleep timer slowdown. Without these slowdowns, we estimate a typical termination time of 15 seconds. With our current hardware, we should be able to handle a 200×200 grid (compared to our current 67×46 grid), which should easily be fine enough to bring our population variance numbers below the goal.

A different solution to the problem of inequalities may be to control the growth of each region for each iteration. Currently, the algorithm will simply run through all the points and update their associated regions accordingly. This has the problem that if point P1 updates point P2, which has not yet been visited, P2 could in turn update other points to join to the region, thus cascading through most of the state in one iteration. To prevent this, there could be an imposition on the points that if they were already changed on this iteration, then they should not be able to change other points. With this in place, regions could only grow outwards one "ring" at a time. We have not closely investigated the effects of such a change in the algorithm, but we suspect it will reduce population inequalities by reducing the chaotic expansions and contractions district currently tend to undergo before finally stabilizing.

Related to this issue is the problem of obtaining good population density data. We were forced to estimate population numbers based on a color scale provided a graph (see figures 1,2). Although the resulting density data appears visually acceptable, we have likely introduced some error by this data entry method.

The other inherent difficulty with our method is the ambiguity in consistently assigning probability weights. The sheer complexity of simulating the district alignment process makes it impossible to determine an appropriate probability function by first principles; some trial and error must be used. This introduces a subjective element into the algorithmic process, which may lead us down a slippery slope towards the gerrymandering we are trying to eliminate. Of course, a completely deterministic algorithm is by its very nature inflexible, and flexibility was one of our initial criteria in choosing our algorithm. We believe that any district-drawing algorithm must sacrifice either flexibility or objectivity to some extent; our current algorithm represents a fair balance between the two.

5.2. **Merits.**

An important advantage of our algorithm is its low termination time described earlier. This gives us the possibility of incorporating finer gridlines, more complicated probability functions, and other improvements. In fact, many of the imporvements listed below hinge upon the high computational speed currently attained by our algorithm.

A less objective merit of our algorithm is the geometric simplicity of the resulting districts. It is our opinion that these districts are not only simple in the sense outlined earlier, but are more realistic than the districts produced by the split-line algorithm.

But the essential merit of our algorithm is its potential for extension. To give just a few examples,

- Suppose we wish to have our districts respect county lines or other political/geographical divisions. To do this we may associate with each grid point an additional piece of information, for example the county to which it belongs. We then adjust the probability formula to include a factor which penalizes grid point reassignments which result in a district crossing county lines.
- In addition, we can associate with each grid point demographic information about the population contained within that grid point. Anything from political leanings to economic class to hair color can easily be incorporated. Once again we freely admit our own inability as mathematicians to associate a meaningful probability factor with these data. But given such a factor, there would be no difficulty in incorporating it into our algorithm.

The only limitations on the number of variables we can include are the availability of the necessary data and the cleverness required to produce meaningful probability factors.

6. Appendix 1 - Probability factors

Without weighting for interior grid points, our probability for district reassignment is as follows, where we are considering a grid point to be potentially reassigned from district A to district B:

$$R_{AB} = \sqrt{\frac{P_A - P_B}{P_A + P_B}} - 0.1, \tag{1}$$

where R_{AB} is the probability for district reassignment, P_A is the current population of district A and P_B is the population of district B. The square root is intended to raise the probability of reassignment when the population difference is low. The subtraction of 0.1 imposes the cutoff mentioned in the body of the text. Any value of R_{AB} less than zero is considered to be zero for the purposes of the algorithm.

This is then multiplied by a factor which favors reassignments which increase the number of interior grid points. The final factor is as follows, where Δ_I is the potential change in interior grid points under this reassignment:

$$R_{AB} = \sqrt{\frac{P_A - P_B}{P_A + P_B}}(0.5 + 0.3 \cdot \Delta_I) - 0.1. \tag{2}$$

The factor of 0.3 multiplying Δ_I was obtained by trial and error.

7. Appendix 2 - Other data

In producing a digital representation of New York State's population density, we compared our entered density figures to satellite images [6] to give a different confirmation of our data. The specific figure used is displayed below.

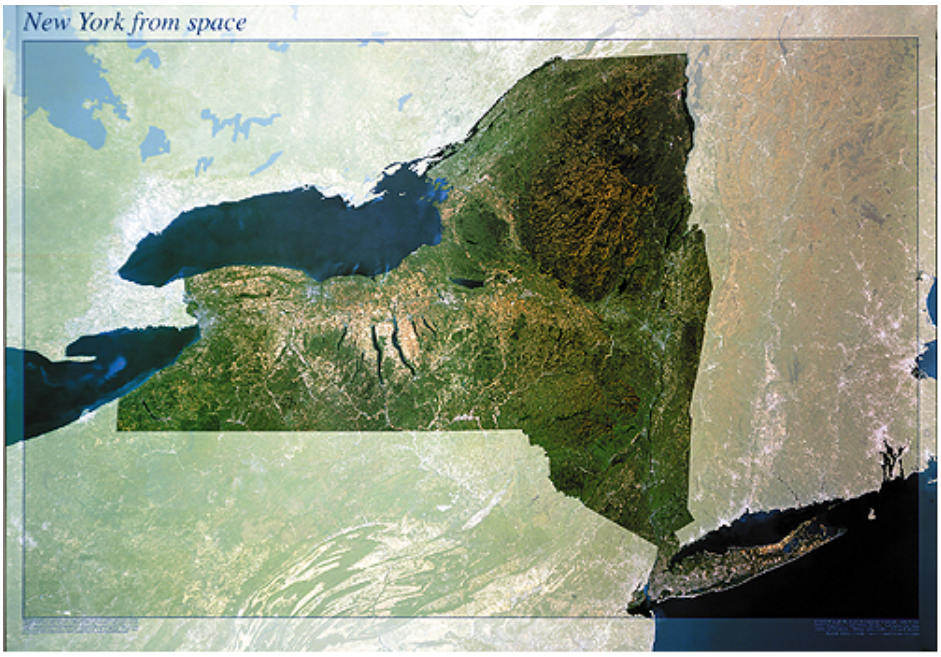

Figure 11. New York State satellite image

References

[1] Legislative Guide to Redistricting. 1995, Iowa General Assembly - Legislative Service Bureau. 2/11/07 <http://www.legis.state.ia.us/Central/LSB/redist.htm#fn20>.

[2] Redistricting Law 2000. January 1999, National Conference of State Legislatures. 2/11/07 <http://www.senate.leg.state.mn.us/departments/scr/redist/red2000/red-tc.htm>.

[3] Examples of Our Unbiased District-Drawing Algorithm in Action. Center for Range Voting. 2/11/07 <http://rangevoting.org/GerryExamples.html>.

[4] Demography - Appendix A: New York State Population Density. January 2004, New York Office for Technology. 2/11/07 <https://www.oft.state.ny.us/SWNdocs/docs/demography.pdf>.

[5] CATO Handbook for Congress. 2003, CATO Institute. 2/11/07 <http://www.cato.org/pubs/handbook/hb108/index.html>.

[6] New York from Space. 1998, M-Sat Corporation. 2/11/07 <http://www.spaceshots.com/images/full/sspn1146.jpg>.