Senior Capstone: Algorithmically Redistricting Maryland

Project Members

**Willie deButts**

Washington University in Saint Louis

(703)-244-0090

[wfdebutts@wustl.edu](mailto:wfdebutts@wustl.edu)

**Noah Lieberman**

Washington University in Saint Louis

(802)-734-3898

[noah.lieberman@wustl.edu](mailto:noah.lieberman@wustl.edu)

Advisor

**Zachary Feinstein**

Washington University in Saint Louis

(314)-935-7536

[zfeinstein@wustl.edu](mailto:zfeinstein@wustl.edu)

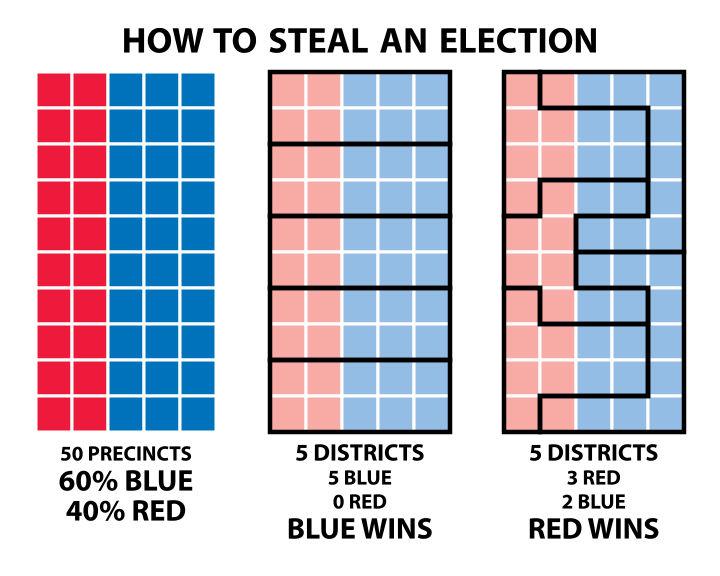
**Abstract:**

United States Representatives and state legislators are elected into office by voting districts within their state. Redistricting is the process of creating a new district boundaries within the state. This process is perceived to be done within the best interests of political parties trying to optimize the number of elected politicians that are a part of their party. The process of redistricting for this gain is known as gerrymandering. Legislators typically hire advisory commissions to make recommendations for the redistricting. The house will vote on the new districts and the governor then has the ability to veto the new boundaries.

No matter how it is done, there is always a political party that loses elected officials because of the redistricting. There have been many attempts to come up with a “fair” way to do it. Our goal was to develop a way to determine district boundaries in a fair and logical way. In this project, we used Python to algorithmically create 8 new districts in Maryland using census data from 2010. We created districts that were contiguous and of equal population. We also created districts that preserved “communities of interest” to protect the interests of the voters. Finally, we investigated the “fairness” of these districts and assessed how our algorithm could be improved.

**Introduction:**

“Gerrymandering is generally defined as the intentional alteration of established political boundaries or the creation of artificial communities by the grouping of political units to form temporary election districts for the purpose of effecting an election outcome. [1]” There are two methods to Gerrymander. “Packing” is the process of grouping all of the minority voters as possible into concentrated districts to dilute their votes in other districts. “Cracking” is the process of splitting the minorities up apart into as many districts as possible to drain their voting power[2]. Below is a graphic of how populations can be manipulated to achieve desired results.

[3]

Maryland has a history of having voting districts that are perceived to be gerrymandered. In 1990, the new census data showed that the population in Congressmen Steny Hoyer’s Fifth District had a large majority of African-American voters. Hoyer was a very important Congressmen and the Democratic Party was nervous that he was going to lose his seat. There is no hard evidence, but it is perceived that Hoyer met with other incumbents to resolve this potential threat.

In 1991, House Bill 10 was adopted to create new voting districts. A district was created with African-American majority so they were free to choose a representative of their choice. Congressmen Hoyer was given a white district so he could retain his position. The political boundaries of Maryland in 1991 were not representative of natural political or geographical boundaries.

**1991 Maryland Voting Districts**

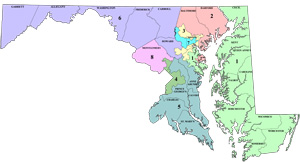


Image from: <http://www.mdp.state.md.us/Redistricting/historical.shtml>

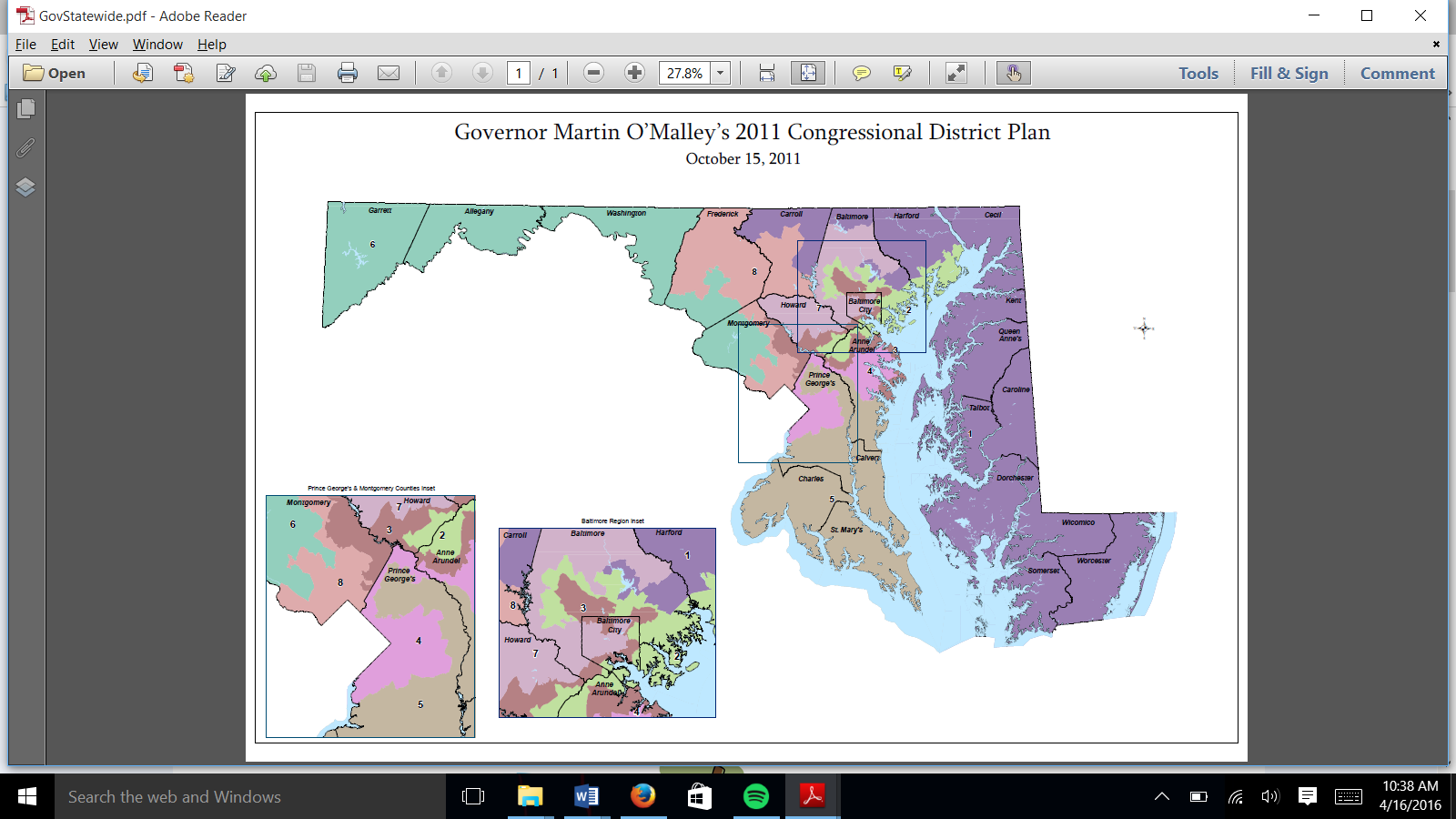
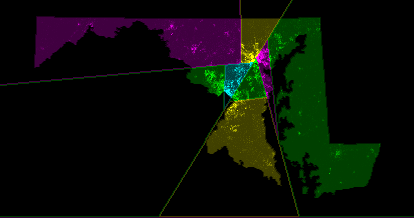
Maryland’s districts have remained this way ever since. In 2011, Maryland created new voting districts after the 2010 census came out (see below). Steve Shapiro, a current law student filed a lawsuit against the voting districts of Maryland for violating the voter’s first amendment. Shapiro alleges that the districts of Maryland were gerrymandered by the Democratic Party [4]. The Supreme Court ruled that a three-judge district court shall hear Shapiro’s case. It is clear that there is a need for a fair way to create voting districts in the state of Maryland.

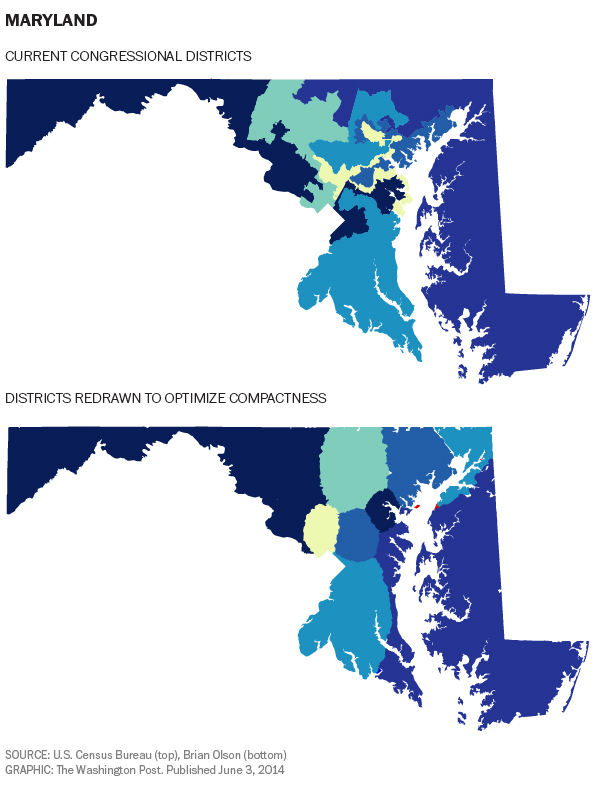
Image from: <http://www.mdp.state.md.us/Redistricting/historical.shtml>

We have encountered two algorithms created to produce unbiased voting districts. The first is the Shortest-Splitline method. The Splitline method recursively divides the state into two boundaries using an optimal line. The optimal line is the shortest geographical line that divides one area into two equally-populated districts. The algorithm is called recursively until N districts are created. The algorithm can add constraints so it doesn’t divide political or geographical boundaries such as a neighborhood, census tract, country, etc.

**Ivan Ryan’s Shortest-Splitline (Maryland 2009)**

 [5]

The second algorithm is the BDistricting tool created by Brian Olson [3]. He optimizes the compactness of the voting districts. This means that he tries to minimize the ratio of the perimeter to the area. These districts are extremely more compact than the voting districts in actuality. They look more natural and are unbiased as well.

[6]

To test the fairness of these two algorithms, Chris Fecor, a programmer in San Francisco, simulated the outcome of the 2008 senate votes using each of the algorithms discussed above. In 2008, North Carolina voted for 8 Democrats and 5 Republicans. Popular vote was 60% for Democrat and 40% Republican. Had the districts been created by the Splitline method, 10 Democrats would have been voted into office. The BDistricting tool would have resulted in 11 Democratic seats. By randomly grouping contiguous voting blocks, Fecor created Districts that elected 9 Democrats. Even though these algorithms were unbiased, they still produced results that didn’t reflect views of the popular vote. Fecor claims to have obtained similar results from the other states he has tested. He came to the following conclusion: “Unbiased redistricting isn’t necessarily fair districting [3].”

**Objective:**

Our goal for this project is to determine voting district boundaries in a fair and logical way. We plan to create an algorithm in python that takes in census data to determine the boundary lines for Maryland.

We want to start out developing the algorithm to group together contiguous voting blocks. We can confirm our results to make sure that we can successfully process the census data and divide up the state in a feasible manner. Once we can confirm our results, we want to improve the algorithm to take in other information to create districts with citizens that share political interests. The algorithm has to potential to take the following information into account: rural/urban divides, ethnicity, cultural background, and socioeconomic status, and geographical boundaries. Doing so will create a “smart” choice when grouping contiguous voting blocks.

Next, we hope to develop an automated visualization tool that creates an image of the state and the new district boundaries. This visualization tool takes in a map of Maryland and color codes the census tracts to create the images of the new districts. This visualization tool makes it possible to analyze our algorithm to see if we achieve logical districts that are made to be fair.

**Methods:**

We first started by creating an algorithm that created eight contiguous districts. In order to do this, we needed to find population data for each census tract. We downloaded data from the 2010 census [7]. Unfortunately, the census data did not include which census tracts were adjacent to each other. Luckily, John Logan’s US2010 project created adjacency lists for each of the census tracts [8]. Using this data, we were able to create a recursive algorithm to create eight contiguous districts.

Our algorithm took in a starting census tract and adds it to the district. The algorithm kept tracks of every adjacent census tract to the district. It selected census tracts to join the district until a target population was reached. The selection of the census tract was the key step to our algorithm. Rather than randomly select contiguous tracts, we wanted to make a decision that resulted in compact districts that look natural. In order to group census tract logically, it selected the adjacent census tract with the fewest district-less neighbors. This selection prevents tracts from being surrounded by one district while the district’s population reaches its limit. Doing this kept the districts from overlapping each other.

The algorithm repeated this process until all eight districts have been created. Once all eight districts have been created, there will still remain a couple districts that still remain unassigned to any districts. The correction phase populated a list of all of the unassigned tracts and assigned them to the district with the most tracts adjacent to them. It did this intelligently by sorting the list by number of assigned adjacent tracts so that the districts retain their natural shape.

When the algorithm terminated, it produced a comma-separated values (CSV) file with each census tract and a corresponding value to indicate the district it belonged to. Using QGIS’s software, “A Free and Open Source Geographic Information System,” and shape files of Maryland, we were able to generate an image of Maryland’s new districts color-coded. The ability to visualize our districts was a very valuable tool in determining the effectiveness of our algorithm.

After we got our algorithm to work, we wanted to preserve the “communities of interest” within our districts. To do this, we had to enhance the selection criteria for the census tract to add to the district. At the time, it selected the tract with the fewest remaining neighbors. When there was a tie, it arbitrarily selected the tract with a higher “GEOID” tag. We wanted to create a tiebreaker that could quantify how similar the tracts were to the district.

Communities of interest can be interpreted in many ways. Kansas has some guidelines for communities of interest: "[s]ocial, cultural, racial, ethnic, and economic interests common to the population of the area, which are probable subjects of legislation." Alabama also has some thoughts on communities of interest, "[i]t is inevitable that some interests will be recognized and others will not, [but] the legislature will attempt to accommodate those felt most strongly by the people in each specific location. [2]"

We created a value function to determine which tract the algorithm would select in the event that multiple tracts had the same amount of remaining neighbors. Using the census data, we were able to find data to quantify similar communities of interest. Based off of Pew Research Center’s findings about demographic and party affiliation we decided to include age, proportion of households that were families, and the population percentage of African-Americans [9]. We came up with the following function:

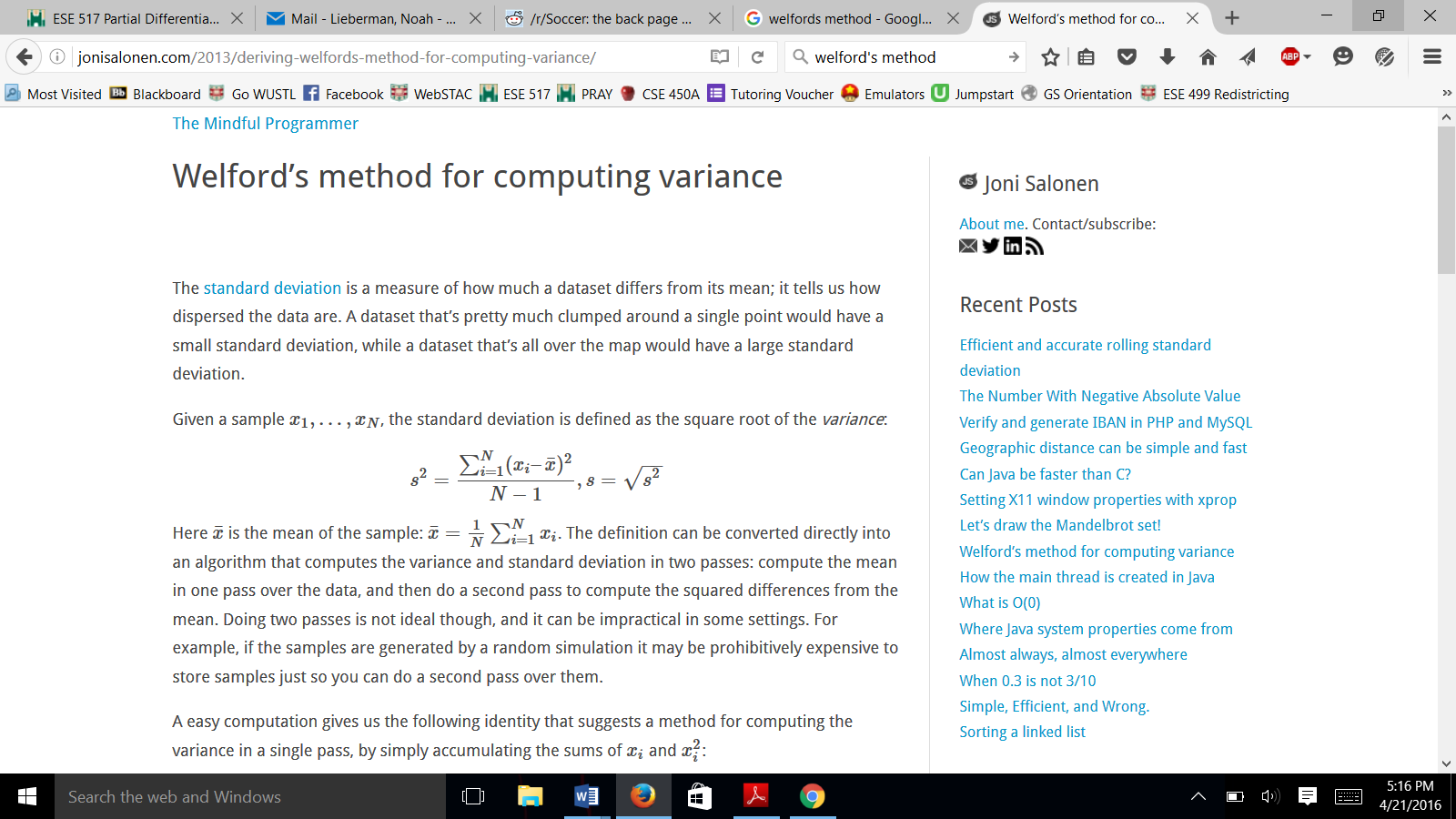
V= A\*(Age) + B\*(Family Percentage)**2** + C\*(African-American Population)**2**

We normalized the values by determining how many standard deviations away from the district’s mean the corresponding values of the tract were. For simplicity, we ignored the population of the tract when calculating the value function. The improved algorithm selected the tract with the lower value.

Age was going to be the driving factor in this value percentage so we made it linear with a high co-efficient. Family percentage wasn’t as important so we gave it a smaller co-efficient, but made it important if it was vastly different by squaring its value. In order to keep race from really affecting the value function, we gave it a small co-efficient. We squared it so it would really only be considered when there was a large difference between the district and the tract. The values for A, B, and C were determined by training the function. We came up with different scenarios and decided how we would rank them by hand. We then adjusting the coefficients accordingly until the function produced desired results for multiple scenarios. The Value function became the following:

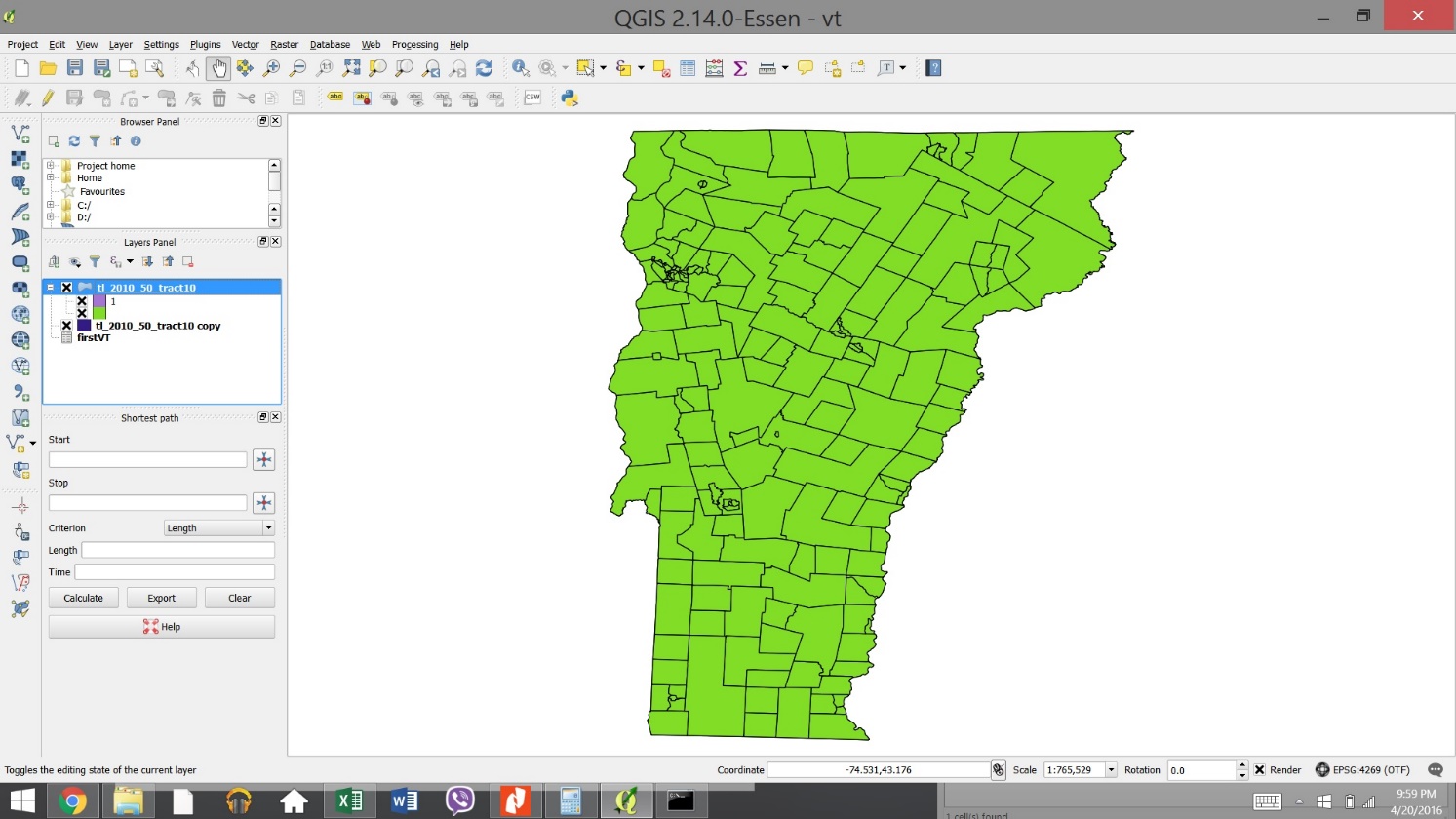
V= **17**\*(Age) + **5**\*(Family Percentage)**2** + **1**\*(African-American Population)**2**

There were some issues when incorporating this function into the algorithm. 16 of the census tracts did not have accompanying census data. We averaged the values of the surrounding census tracts using the assumption that adjacent census tracts are similar in demographic. When a district was selecting its first tract, the standard deviation needed to have a starting value. We initialized it to one and it converged to the true value over time. We used Welford’s method to update the deviation every time a tract was added to the district [10].

 [10]

**Results:**

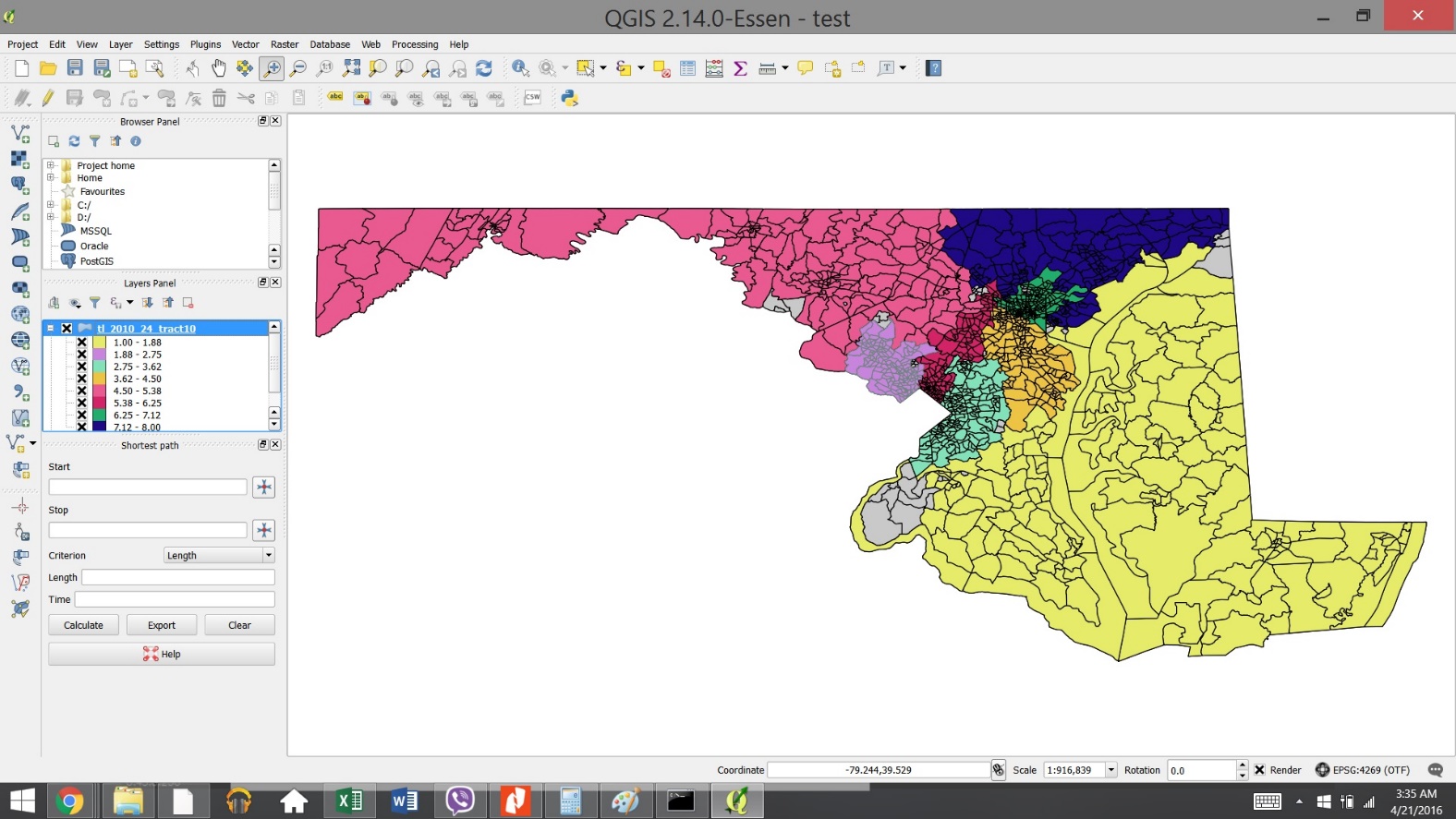
To test our algorithm worked, we used the state of Vermont. Vermont only has one voting district so the algorithm, if successful, would produce a monochromatic map of Vermont. If there were any census tracts left out you would see them displayed in bright pink. Below is a picture of what our algorithm produced.

****

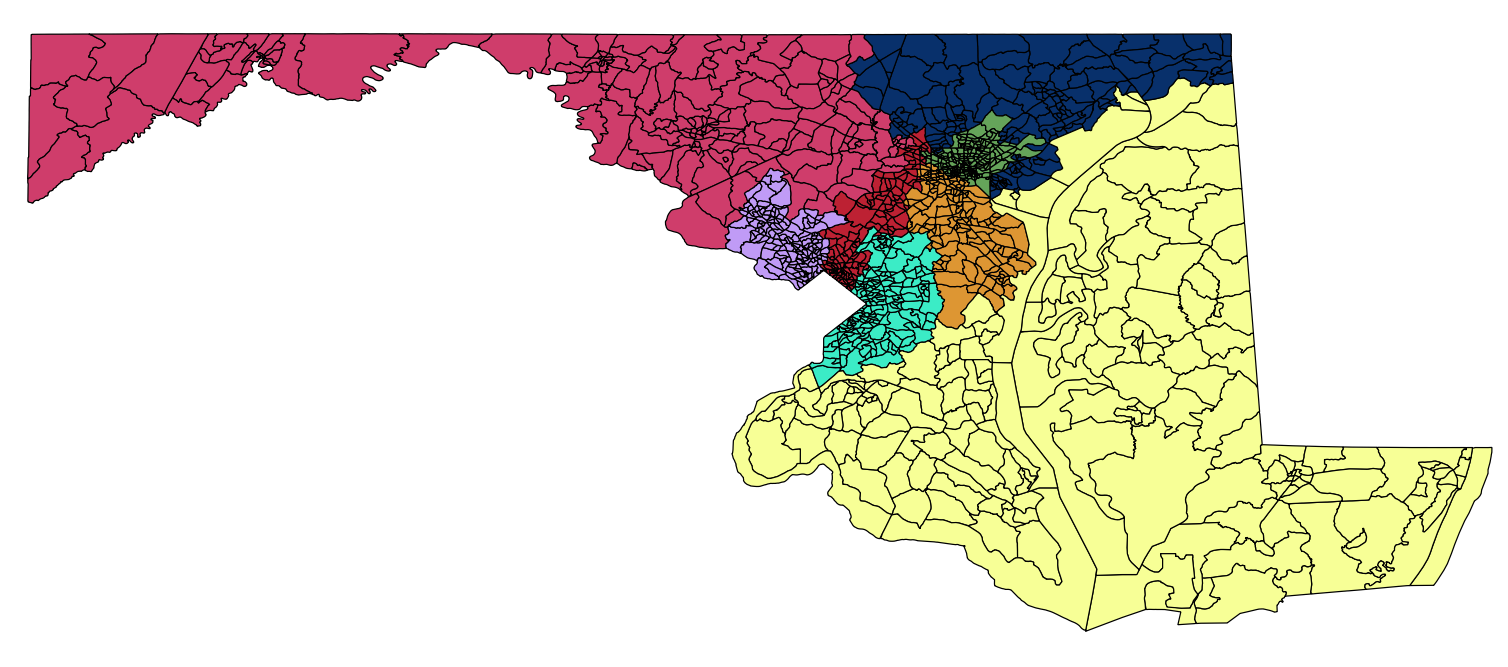
As you can see there is no pink color and there are no residual districts left after the first run of the algorithm. This is exactly what we expected from our algorithm.

Now we can turn our attention to Maryland and the primary result. First displayed below is the map of the raw output of the algorithm. Each unique color represents a district and the grey areas in the graph are tracts not assigned to a district due to issues inherent in the recursive approach. The slack value was 30,000. Here is the legend for the map:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **NA** |



Now we can apply the correction algorithm to force all tracts into a district. It worked except for two tracts which were entirely enclosed by district 5 so we manually assigned them to district 5. The resulting map is displayed below, it follows the same legend as above.

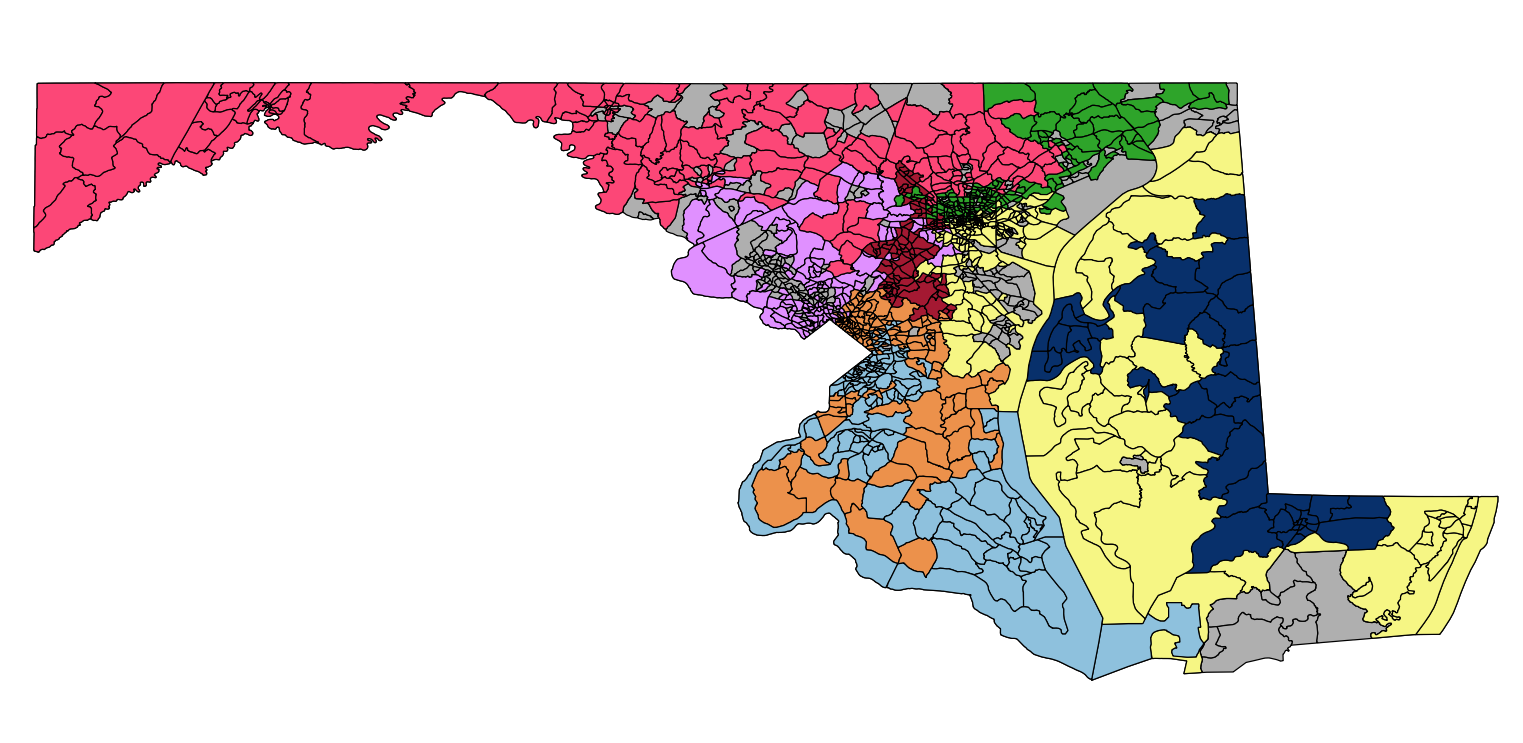


The sample mean and standard deviation of all the variables used in our value function were recorded for each district and are displayed below to use in comparison to districts generated by the value function.

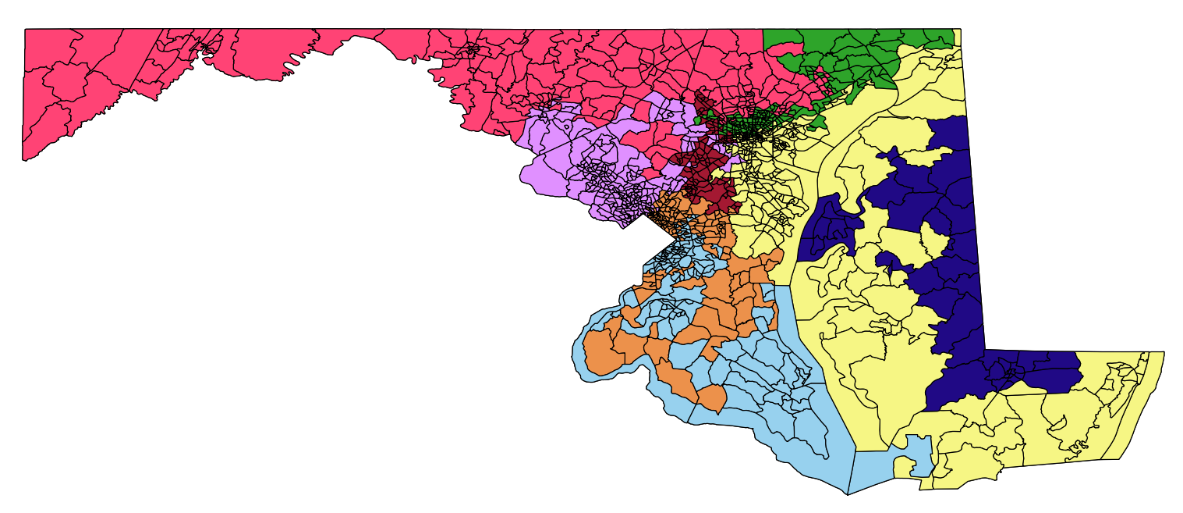
**Algorithm 1 demographic results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| District | Age mean | Age σ | Family % mean | Family % σ | AA pop  % mean | AA pop  % σ |
| 1 | 41.21 | 7.07 | 0.707 | 0.164 | 0.1963 | 0.1645 |
| 2 | 39.6 | 5.768 | 0.708 | 0.137 | 0.119 | 0.137 |
| 3 | 36.71 | 4.87 | 0.688 | 0.1511 | 0.706 | 0.15 |
| 4 | 38.12 | 7.03 | 0.6715 | 0.0268 | 0.2547 | 0.052 |
| 5 | 40.75 | 4.83 | 0.713 | 0.190 | 0.072 | 0.186 |
| 6 | 36.86 | 5.67 | 0.663 | 0.179 | 0.340 | 0.180 |
| 7 | 36.65 | 5.97 | 0.566 | 0.170 | 0.533 | 0.171 |
| 8 | 40.89 | 6.11 | 0.709 | 0.095 | 0.112 | 0.094 |

Then we moved into the more complex value function outlined above. The slack term for this run was 100,000. The output is below with the same legend as all the previous maps.



As you can see there are more holes in this graph which makes sense given the selection function. The cleaned districts are below.



For each district we recorded the mean and standard deviation of each variable of our value function:

**Algorithm 2 demographic results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| District | Age mean | Age σ | Family % mean | Family % σ | AA pop | AA pop |
| % mean | % σ |
| 1 | 40.64 | 7.744 | 0.619 | 0.137 | 0.2507 | 0.137 |
| 2 | 43.58 | 7.08 | 0.718 | 0.197 | 0.105 | 0.196 |
| 3 | 35.98 | 4.56 | 0.705 | 0.098 | 0.633 | 0.105 |
| 4 | 37.5 | 5.77 | 0.681 | 0.181 | 0.4452 | 0.183 |
| 5 | 45.33 | 5.44 | 0.705 | 0.175 | 0.063 | 0.1752 |
| 6 | 35.1 | 3.85 | 0.66 | 0.088 | 0.411 | 0.086 |
| 7 | 39 | 4.88 | 0.623 | 0.221 | 0.426 | 0.226 |
| 8 | 38.59 | 5.23 | 0.678 | 0.131 | 0.197 | 0.165 |

Here is a comparison of the sigma values between the first (simple) and the second (complex) algorithms:

**Sigma values by algorithm**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| District | Age (1) | Age (2) | Family % (1) | Family % (2) | AA pop | AA pop |
| % (1) | % (2) |
| 1 | 7.07 | 7.744 | 0.164 | 0.137 | 0.1645 | 0.137 |
| 2 | 5.768 | 7.08 | 0.137 | 0.197 | 0.137 | 0.196 |
| 3 | 4.87 | 4.56 | 0.1511 | 0.098 | 0.15 | 0.105 |
| 4 | 7.03 | 5.77 | 0.0268 | 0.181 | 0.052 | 0.183 |
| 5 | 4.83 | 5.44 | 0.190 | 0.175 | 0.186 | 0.1752 |
| 6 | 5.67 | 3.85 | 0.179 | 0.088 | 0.180 | 0.086 |
| 7 | 5.97 | 4.88 | 0.170 | 0.221 | 0.171 | 0.226 |
| 8 | 6.11 | 5.23 | 0.095 | 0.131 | 0.094 | 0.165 |
| AVG | 5.915 | **5.569** | **0.139** | 0.154 | **0.142** | 0.159 |

The Standard deviation of age within a district was decreased by selecting the districts using the value function; however, the standard deviation increased for family proportion and African-American population proportion.

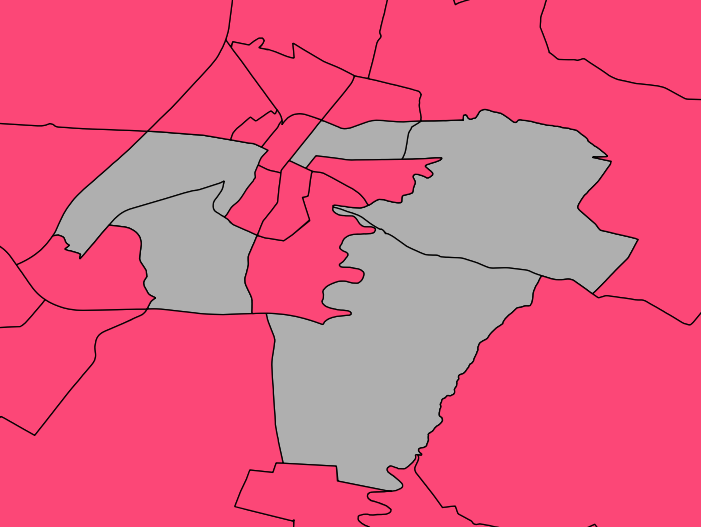
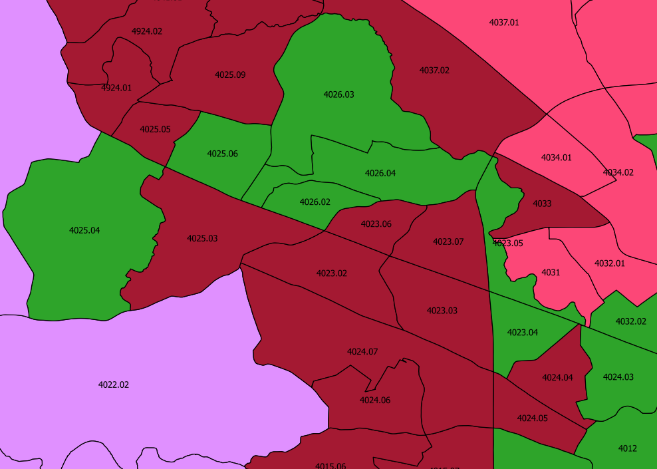
**Discussion:**

The discussion of our results can be broken down into two primary areas, first the recursive method used, and second the resulting districts.

By definition of the recursive method the first districts drawn will be able to roam freely across the state adding the tracts that best suit their value function since comparatively few of the tracts are already placed into districts. The later tracts will have to add tracts with low score values since they are geographically bounded and won’t reach the target population. In our data districts one through six were able to be computed with minimal loss and interference but districts seven and eight suffered the most from this. As a result these districts have some of the highest standard deviations across all three of our variables.

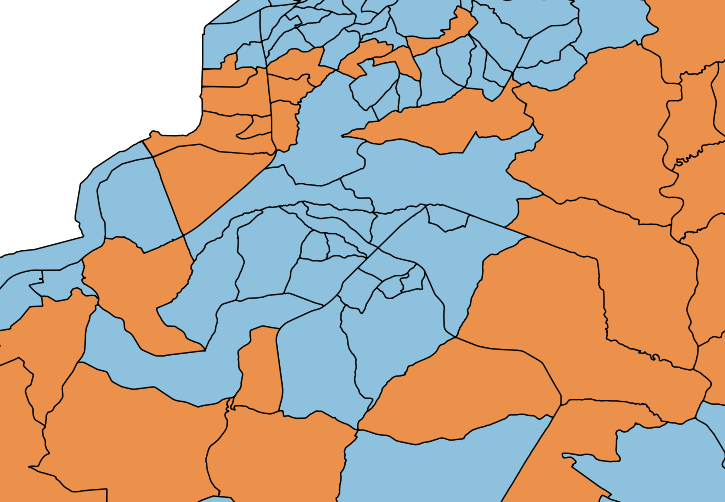
Using the simple compactness condition actually created surprisingly good results. They pass the eye test since they are in fact far more compact than Maryland’s existing districts. Due to the clumping nature inherent in this selection method many of the numerical problems present in our method did not emerge here significantly. We were able to set our slackness term to 1/3 of what we needed in order to get a result for the value function. As a result the correction phase played a relatively minor role here.

Adding our value function instead of emphasizing compactness changes a lot of our results. At first glance these districts are a lot less logically shaped. Our complex value function the recursive approach led to a lot of districts ‘running’, and ‘circling’. A district running is when the algorithm forms a bridge between two areas with similar demographic information. Circling behavior is when a district primarily based in rural or suburban areas does not want to add urban tracts to itself as the value functions for this tract are very low.

Encirclement in District 5 Running in District 7

Additionally using the recursive method meant that we had to choose the starting tract for each district while this was done essentially at random from the pool of remaining tracts it inevitably lead to an increase in subjectivity of our results. Also our value function struggled when all adjacent districts had roughly the same score. This lead to districts becoming entangled in illogical ways, an example of this is the entanglement of districts three and four.



Interestingly the orange district was drawn before the blue one but the values for all of these tracts are similar so the marginal differences lead to a non-optimal shape. Overall while the recursive approach did lead to some odd behavior it still generated a workable outcome that we believe are still superior to Maryland’s current districts.

We were happy that our value function lead to decreases in the standard deviation of age with little increase in the standard deviation of family proportion and African-American population proportion. The results matched logical expectation in most places. Furthermore, many of our districts follow well defined communities that should have their own representative. For example, Appalachian west Maryland is entirely in District 5, the tech savvy I-270 corridor to the northwest of DC makes up almost all of District 2, and the agrarian/waterman communities of the eastern shore are contained in District 1.

In other places our method fails; District 8 is terrible and it exists as an artifact of the recursive method. It is possible that our value function fails in creating districts that reflect the true communities of Maryland. In order to improve our value function, we could take input from the public. Community members can go to their officials and assist them with defining communities of interest. We would adjust our algorithm to combine the census tracts that are in these predefined communities into one. Nonetheless, we still believe our value function created workable districts in its current form.

It is possible that the solution to the redistricting problem is to turn away from it completely. Now that the world is so connected, geographical location isn’t as important anymore. It is possible that the need for contiguous voting districts is outdated. These regulations were created in the 18th century when most Americans were farmers.

An alternative solution is popular vote. If each state were to create one voting district, there wouldn’t have to be a winner-takes-all voting system. This would work perfectly under the assumption that Democratic and Republican candidates were interchangeable with each other. Unfortunately, this is not true. There are issues with popular vote as well. If states implemented a ranked voting system, states could solve the problem with non-interchangeable candidates. This starts to get ugly in bigger states like California. Voters don’t have the time to familiarize themselves with 53 candidates that they would vote for. Similarly, candidates don’t have the time or resources to campaign to the entire state of California. It is currently much easier for voters to focus on only one election. Popular vote may not be any easier to implement than winner-take-all elections in voting districts.

**Conclusion:**

We were able to create an algorithm that produces eight voting districts of equal populations. Our first algorithm produced compact districts that were contiguous. These districts looked very natural. Due to the random nature of the algorithm, the districts are completely unbiased. Our second algorithm produced districts that aren’t completely contiguous, but are still more compact than the existing districts in Maryland. They preserve communities of interest based on our value function. Because we used a value function, there is a chance that we inadvertently created biased districts.

With more time, we would like to find data from previous elections to simulate the results using the districts our algorithm created. Simulating elections would give insight to any bias associated with the districts. This bias could be due to the natural population distributions in Maryland, or the algorithm itself.

Finally, we would like to see how our algorithm would hold up creating districts for other states. Other states could be more naturally biased than Maryland. We would have to simulate the elections of many states to determine if the algorithm can be used for more than just Maryland.

**Bibliography:**

[1] Niemeyer, Paul V. "The Gerrymander: A Journalistic Catch-Word or Constitutional Principle? The Case in Maryland,” *Maryland Law Review,* vol. 54, pp. 242-260, 1995.

[2] J Levitt. (2011). *All About Redistricting* [Online] Available: <http://redistricting.lls.edu/index.php>

[3] R. Cima. (2016, Jan 22). *Can an Algorithm Eliminate the Unfairness of Gerrymandering?* [Online]. Available: <http://priceonomics.com/algorithm-the-unfairness-of-gerrymandering>

[4] R. Barnes. (2015, December 8). *Supreme Court says challenge of Maryland district lines may proceed* [Online]. Available: <https://www.washingtonpost.com/politics/courts_law/2015/12/08/190bd7c0-9dbe-11e5-8728-1af6af208198_story.html>

[5] Range Voting. (2009, September 21). *Splitline districtings of all 50 states + DC + PR* [Online]. Available: http://rangevoting.org/SplitLR.html

[6] C. Ingraham. (2014, June 3). *This computer programmer solved gerrymandering in his spare time* [Online]. Available: <https://www.washingtonpost.com/news/wonk/wp/2014/06/03/this-computer-programmer-solved-gerrymandering-in-his-spare-time/>

[7] Investigative Reporters and Editors. (2010). *Download Bulk Census Data* [Online]. Available: <http://census.ire.org/data/bulkdata.html>

[8] US2010. (2010). *Census geography: Pooling adjacent tracts to improve reliability of estimates* [Online]. Available: <http://www.s4.brown.edu/us2010/Researcher/Pooling.htm>

[9] Pew Research Center. (2015, April 7). *A Deep Dive Into Party Affiliation* [Online]. Available: <http://www.people-press.org/2015/04/07/a-deep-dive-into-party-affiliation>

[10] J. Salonen. (2016). *Welford’s method for computing variance* [Online]. Available: http://jonisalonen.com/2013/deriving-welfords-method-for-computing-variance/

**Appendices:**

