St. Louis 2022 Ward Redistricting

Austin Tolani

U90 GIS 422

Mollie Webb, Instructor

Heather Navarro, Client

Washington University in St. Louis

29th April, 2020

TABLE OF CONTENTS

EXECUTIVE SUMMARY

In 2022, St. Louis will be reducing its number of wards from 28 to 14, based on the results of the 2020 census. This project explores three potential methods for generating new ward boundaries: manual redistricting, use of Auto-Redistrict (a computer program) and use of the Better Automated Redistricting (BARD) statistical software package. Redistricting criteria, such as equal population, compactness of wards, ward contiguity, preservation of minority voting power, and preservation of communities of interest are identified. Methods used to evaluate redistricting criteria are explained.

The project found that of the redistricting methods considered, the most effective method for generating ward boundaries to meet redistricting criteria was a combination of use of Auto-Redistrict and manual redistricting. This method was used to generate five potential ward boundaries for the City of St. Louis, each prioritizing equal population, compactness, preservation of minority voting power, neighborhood cohesion, and a balance between all criteria, respectively. The variation in performance in redistricting criteria across these ward boundaries illustrated that even with a good method to generate ward boundaries, subjective decisions are still required on the part of those drawing the boundaries.

To share the results of this project, a public facing website was developed. The website addresses an important goal of the project: to inform and engage the public in the redistricting process. This website provides background on redistricting, explanation of redistricting methods and criteria, results in the form of interactive web maps, a section for users to provide feedback and a detailed explanation of the workflow of the project. To create the interactive web maps, two JavaScript libraries, Leaflet.js and Ds.js were used.

INTRODUCTION, CLIENT DESCRIPTION AND BACKGROUND

Since the 1950's, the City of St. Louis has experienced significant population decline, losing over half of its population (Richard L. Forstall, 1995). Despite this population loss, St. Louis is still governed by 28 aldermen and alderwomen who represent 28 wards. In 2012, voters in the City of St. Louis passed proposition R, a measure to reduce the number of wards (and consequently aldermen and alderwomen) from 28 to 14. The measure is now known as St. Louis City Ordinance #69185 (*Ordinance 69185*, 2012). This ordinance stipulates that beginning January 1, 2022, the city will be divided into 14 wards and such ward boundaries will be based on the 2020 Census.

Redistricting of political boundaries is a very important issue in politics. Political lines significantly shape elections and thus have an influence on what party and candidate are chosen to represent constituents. In the 1960's, the U.S Supreme Court reached landmark decisions on redistricting, and stipulated that districts must have equal population, among other criteria (Engstrom, 2013). Following this, various automated redistricting programs were developed and envisioned to remove the politics from redistricting. However, redistricting is an extremely computationally intense problem and no algorithms have yet been developed that output a perfect redistricting plan. As a result, it was found that computational programs were best-suited to assist planners in dealing with the large amount of data associated with redistricting (Altman & McDonald, 2011). In fact, it was due to this use that some of the first Geographic Information Systems (GIS) were born (Altman et al., 2016). While there are various computer programs that aid with redistricting, there is still no method that is considered to be the best for. This project will explore various methods for redistricting, comparing their effectiveness, ease of use, and results.

My client for this project is Heather Navarro, the Alderwoman for Ward 28 in the City of St. Louis. As 2022 approaches, Ms. Navarro identified a need for an independent project to research redistricting methods, propose potential ward boundaries, and create a publicly facing resource with the results of the project. When conversations among politicians and citizens about redistricting occur, Ms. Navarro would like an independent resource that provides information on redistricting methods, potential ward boundaries, and other information on redistricting that can be referred to.

The overall goal of this project is to create a useful resource on the subject on St. Louis's 2022 ward redistricting that will engage and inform the public. Within this resource will be a thorough exploration of various methods and criteria used to generate ward boundaries as well as potential ward boundaries created by these methods.

Data Sources

Various data sources were used in this project. For the redistricting analysis, the smallest unit of analysis used was census block groups. Census block groups for the City of St. Louis were downloaded from the U.S. Census Bureau's website. The American Community Survey's 2018 5-year estimates were used to get the most up to date population and racial data for the census block groups. Additionally, boundaries representing various political and geographic units in St. Louis, such as neighborhoods and parks, were downloaded from the City of St. Louis's Open Data Portal.

DATA PREPARATION

There were several parts of this project that required significant data preparation, including initially preparing the block group data for analysis, preparing the data for use in other software, and preparing the data for use in online web maps.

Preparing Block Group Data for Analysis

To prepare the block group data for analysis, the tabular data from the American Community Survey's 5-year estimates were joined to the shapefile containing the census block group boundaries. Then, a spatial join was performed to join the neighborhoods to the block groups, so that a field was created that identified which neighborhood each block group fell inside. In the case of conflicts, census block groups were assigned a neighborhood based on the location of their centroid.

Preparing Data for use in other Software

For the most part, a shapefile of the block groups containing tabular demographic data was suitable for import into most of the redistricting software that I used. The one exception was the Better Automated Redistricting (BARD) package. In order to import a shapefile, BARD required a file representing the contiguity of features. More specifically, BARD required an .GAL weights file along with the shapefile in order to be imported. A .GAL weights file is a text file that contains, for each observation, the number of neighbors and their identifiers (Luc Anselin, 2018). This file format is not found in many GIS software, including ArcMap, so I used a free and open source software tool called GeoDa to create this file. Once this file was created, it was relatively straightforward to import the shapefile into BARD.

Preparing Data for Online Web Maps

In order to display the created ward boundaries in a web map, I had to host them as web services. To do this, I used ArcGIS Online, which allows users to upload shapefiles which are then hosted as feature layers that can be referenced in a web map. In addition, in order to display how each ward performed on various redistricting criteria (explained later in this paper), the attributes of these shapefiles had to be updated with the relevant calculations, such as their total population and racial breakdown. To calculate and add these attributes, I created a python script that would aggregate various attributes from the census block groups into wards.

Analysis Methodology

Redistricting Criteria

In order to compare the ward boundaries generated in this project, it is important to have a set of criteria that we will evaluate against. Ordinance #69185 stipulates that the new ward boundaries "shall comprise as nearly as practicable, compact and contiguous territory within straight lines, and contain as nearly as may be the same number of inhabitants"(*Ordinance 69185*, 2012). In addition to these criteria, there are other criteria that should be considered, as identified by Supreme Court case law. Such criteria include the preservation of communities of interest, preservation of existing political and geographical boundaries, and respect for minority representation (Justin Levitt, 2010). Listed below are all of the criteria considered in my analysis, how such criteria were measured, and the values needed to satisfy such criteria.

Equal Population

Ward populations should be roughly equal in number. Supreme court law has shown that "roughly equal" generally means a total population deviation less than 10%. In order to measure this, we can perform the following calculations:

$$Ideal\ Population = \frac{Total\ City\ Population}{Number\ of\ Wards\ (14)}$$

$$Deviation\ for\ a\ ward = Ward\ Population - Ideal\ Population$$

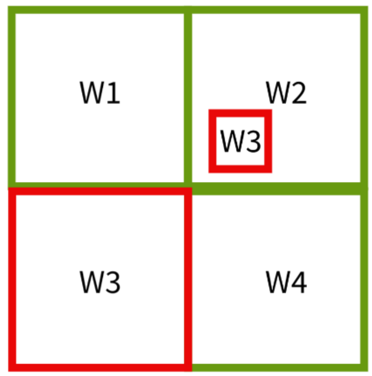
$$Total\ Population\ Deviation$$
$$= |Largest\ Positive\ Deviation| + |Largest\ Negative\ Deviation|$$

So, to meet this criteria, the total population deviation, as calculated above must be less than 10%.

Contiguity

In order to be contiguous, every part of the ward must be reachable from every other part without crossing the ward boundary. Contiguity can be very easily assessed visually. In order to satisfy this criteria, all wards must be contiguous. Figure 1 shows a visual representation of contiguity of wards. In this diagram, W1,W2, and W4 represent wards that are contiguous while W3 represents a ward that is not contiguous.

Figure 1: Contiguity of Wards



Compactness

Every ward should have a regular shape with constituents living relatively close to each other. To measure compactness, we can use the Polsby-Popper test. This is a mathematical measure of compactness. The value of this test will always fall between 0 and 1, where a score of 0 indicates a lack of compactness (a straight line) while a score of 1 indicates maximal compactness (a circle). The formula for this calculation is given below:

$$PP(W) = \frac{4\pi A(W)}{P(W)^2}, where\ W\ is\ the\ ward,$$

$$P(W)\ is\ the\ perimiter\ of\ the\ ward\ and\ A(D) is\ the\ area\ of\ the\ ward.$$

To meet this criteria, the average Polsby-Popper score across all wards should be greater than 0.3. Figure 2 shows a visual representation of a compact (W2) and non-compact ward (W1).

Figure 2: Compactness of Wards

Preservation of Minority Voting Power

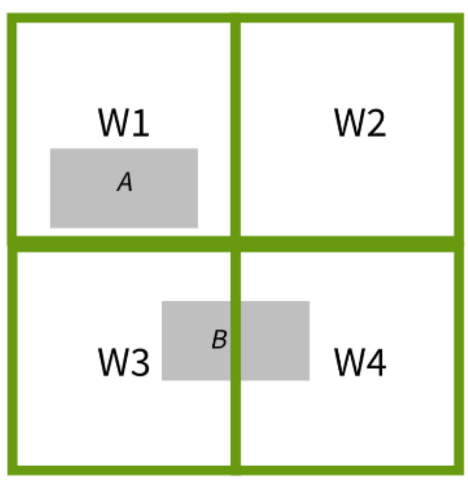
Minority voting power should be preserved such that minority groups should have an effective opportunity to win a certain number of wards based on the region's total population. St. Louis is 46.2% African American, according to the American Community Survey's 2018 5-year estimates. Therefore, 46.2% of the wards (6-7 wards) should allow for African Americans to have an effective opportunity to elect a candidate of their choosing. An effective opportunity means having more than 50% minority population in a given ward. While St. Louis does have other minority populations, none have large enough of a population where they require a ward with an effective opportunity to elect a candidate of their choosing. In summary, to meet this criteria, there must be 6-7 wards with > 50% African American population.

Preservation of Communities of Interest and Political and Geographic Boundaries

Communities of interest and political and geographic boundaries should be preserved where possible. Wards should not split up such boundaries. Neighborhoods in St. Louis represent communities of interest. Political boundaries in St. Louis include Special Business Districts, Tax

Increment Financing Districts, Transit Orientated Development Districts and Community Improvement Districts. Geographic boundaries include parks. For a given community of interest, political or geographic boundary, we can calculate how many times it is split up by the ward boundaries. We can do this for all features and calculate the average number of splits. Ward boundaries should minimize the number of splits that occur for communities of interest and political and geographic boundaries. To meet these criteria, for any given set of boundaries (such as neighborhoods), the average split score across all features should be less than 1. Intuitively, this means that on average, each feature is getting split less than once. Figure 3 shows a representation of this calculation. In this diagram, feature "A" falls completely within ward 1, giving it a split score of 0. Feature "B" is split once between wards 3 and 4, giving it a cohesion score of 1. The average split score for both of these features is 0.5.

Figure 3: Cohesion



REDISTRICTING METHODS

In this project, I experimented with three different methods to generate ward boundaries: manual redistricting (drawing new ward boundaries by hand), use of Auto-Redistrict (a computer program) and use of BARD (a statistical software package). The latter two are the best documented free and open source software packages available for redistricting analysis.

Manual Redistricting

To perform manual redistricting, I made use of Esri's Districting extension for ArcMap/ArcCatalog. This useful extension allows you to draw boundaries while summarizing various statistics relevant to redistricting. To draw boundaries by hand, I carefully examined demographic data and boundaries of interest to iteratively improve the boundaries until I reached a reasonable solution.

Auto-Redistrict

Auto-Redistrict is a computer program that automatically creates fair and compact electoral districts. It uses a heuristic search algorithm to evaluate potential solutions based on criteria like equal population, contiguity, compactness, minimal splitting and minimal racial gerrymandering. Once I imported the data for St. Louis and configured how the program prioritized different criteria, I was able to easily generate and export a solution for St. Louis.

BARD

BARD is a software package for general redistricting and redistricting analysis. To use this package, you must install it as a library in R, the statistical programing language, then use the provided functions to write a script that will generate boundaries. It supports automated generation of redistricting plans by assigning different weights to various criteria. In this software, you first generate an initial plan using a variety of options, including random assignment or using k-means. Then you define a scoring function that combines the various criteria you are interested in optimizing such as contiguity, compactness, and equal population. Finally, you select an optimization algorithm which will yield an improved plan. After experimenting with various scoring functions and optimization algorithms, I was able to create a reasonable solution. Appendix A shows the R code used to create this solution.

Calculating Redistricting Criteria

In order to actually calculate the redistricting criteria for a generated ward boundary, I created a python script and used ArcPy, a Python package, to perform the calculations described by each criteria and output the results. This script allowed me to easily evaluate whether a set of ward boundaries met the criteria described above.

Results

After experimenting with manual redistricting, Auto-Redistrict, and BARD, I found that Auto-Redistrict was the method that produced the best results. Table 1 shows how the boundaries produced by each method compared against each other on the evaluation criteria. Note that with all three methods, I aimed to create boundaries that at minimum, met the defined redistricting criteria. Manual redistricting and Auto-Redistrict both yielded boundaries that met all of the criteria. BARD met all the criteria except for equal population, with its total population deviation equal to 57.8%.

Table 1: Comparison of Outputs of Redistricting Methods

Criteria	*Manual Redistricting*	*Auto-Redistrict*	*BARD*
Total Population Deviation	8.87%	2.92%	57.8%
Contiguity	All wards are contiguous	All wards are contiguous	All ward are contiguous
Compactness (0-1)	0.39	0.51	0.47
Preservation of Neighborhoods (average number of splits)	.26	0.17	.35
Preservation of Minority Voting Power	6 wards with African American population > 50%	6 wards with African American population > 50%	6 wards with African American population > 50%

BARD's poor performance is not necessarily a result of how the software was implemented but rather how I used it. Because BARD leaves so much up to the user, the results created by BARD are contingent on how the user chooses to implement initial plan generation, score functions, and optimization algorithms. The results in Table 1 represent the best ward boundaries I was able to generate with BARD after experimenting with different score functions and optimization algorithms. While I am certain that someone with strong proficiency in R and knowledge of optimization algorithms would be able to create better plans, for the average user, BARD is not able to create plans that out-perform Auto-Redistrict.

While manual redistricting was able to create a plan that met all the criteria, this was an extremely tedious and timely process and not one I would recommend. Iteratively assigning block groups is very tedious and requires a lot of work from the user. With that said, manual redistricting gives the user the most power over what the final ward boundaries look like which could be very useful in certain cases. If I had to recommend a methodology for generating ward boundaries based on my experience and the results, it would be to use Auto-Redistrict to generate an initial plan that met all legal criteria, and then use manual redistricting to tweak this plan to meet any other criteria specified by the user. To create a set of potential ward boundaries for St. Louis, this is precisely the methodology I used.

In my use of Auto-Redistrict, it became clear that the generated ward boundaries are strongly influenced by the criteria that you set as priority in the program. In the set of all ward boundaries that meet the criteria there is a large variation in what the ward boundaries could look like. To demonstrate this and to create different potential ward boundaries for St. Louis, I used Auto-Redistrict to generate 5 different ward boundaries, each prioritizing a different criteria: equal population, compactness, neighborhood cohesion, minority voting power, and balance between all criteria. In some cases, manual redistricting was used to slightly improve the plans once they had been generated in Auto-Redistrict. Table 2 shows how these plans perform on with regard to redistricting criteria. Notably, they all meet all of the criteria but yet have significant variation on how they perform.

Table 2: Performance of Potential St. Louis Ward Boundaries

Criteria	*Equal Population*	*Compactness*	*Neighborhood Cohesion*	*Minority Voting Power*	*Balanced*
Total Population Deviation	2.64%	8.05%	9.03%	9.97%	2.92%
Contiguity	All wards are contiguous	All wards are contiguous	All ward are contiguous	All ward are contiguous	All wards are contiguous
Compactness (0-1)	0.49	0.54	0.45	0.42	0.51
Preservation of Neighborhoods (average number of splits)	0.24	0.25	.13	0.26	0.17
Preservation of Minority Voting Power	6 wards with African American population > 50%	6 wards with African American population > 50%	6 wards with African American population > 50%	7 wards with African American population > 50%	6 wards with African American population > 50%

Discussion and Conclusions

The results in Table 1 indicated that of all the methods considered, Auto-Redistrict consistently produced the best results. This software gives the user a good balance of powerful automated redistricting as well as the ability to prioritize different criteria. While BARD provides a similar functionality, it requires too much specification of score functions and optimization algorithms for the average user. Manual redistricting, while not optimal for generating plans, proved to be very useful for refining plans to meet specific criteria. Based on the software currently available, I recommend the use of Auto-Redistrict to generate initial plan boundaries and manual redistricting to tweak and improve these initial plans.

This analysis also showed that even with a good way to automatically generate plans that meet redistricting criteria, there is significant variation in plans based on which criteria are prioritized. Table 2 shows the extent of this variety. All of the plans in Table 2 meet the redistricting criteria, but perform differently. It is clear from these results that there are often trade offs between prioritizing different criteria. While this may seem intuitive, it suggests that in the redistricting process, there is no such thing as a perfect solution. It will still be up to those tasked with drawing the boundaries to make subjective decisions on which criteria to prioritize.

DELIVERABLES

In addition to analysis of redistricting methods and the outputs they produced, summarizing all of this information into a useful resource was a very important aspect of this project. As mentioned earlier in this paper, the client wanted to use the results of this project as a resource that can be referred to by politicians and citizens as there is more discussion of St. Louis's ward redistricting in 2022. Thus, creating a resource that succinctly and effectively summarized the results of this project was very important.

I chose to create this resource in the form of a website. This ensured that majority of St. Louisan's would be able to access this resource at little cost or effort. Additionally, a website allows for interaction with the data in a way that would not be possible in other formats like a research paper. Finally, a website also allows users to easily leave feedback, which was an aspect very important to my client.

The website I created is split into six sections: background, redistricting criteria, redistricting methods, results, feedback, and workflow. The first four of the sections contain a lot of the same information found in this paper in a more concise and simple manner. The results section of the website contains web maps that show the boundaries of the five ward boundaries in Table Y that were generated using a combination of Auto-Redistrict and manual redistricting. As stated in the discussion and conclusion section, my analysis found that there is a wide variety of solutions that meet redistricting criteria. Again, this implies that it will ultimately be up to those drawing the boundaries to make subjective decisions on what to prioritize. The goal of the results section was to show St. Louis citizens and politicians what prioritizing different criteria looks like, so that they can weigh in on how to make such subjective decisions. The feedback section of the website prompts the user to leave their thoughts on what criteria they would like to see prioritized in the redistricting process. Finally, the workflow section provides information on the various methodologies employed in this project, allowing people in the future to replicate or improve upon my work. This includes this paper, how I used various software, and links to public repositories containing any code I developed for this project.

To create the web maps found in the website, I used Leaflet, an open-source JavaScript library for creating interactive maps. Through leaflet, I was able to customize and optimize the web maps for this specific use case. When a user clicks on a proposed ward, a popup shows various statistics for that ward, including its total population, voting age population, and compactness score. In addition, the popup creates a chart created using D3.js, another JavaScript library, showing the racial breakdown of the population in the proposed ward.

Limitations and Directions for Future Research

One major limitation of this project was the use of census block groups as the smallest geographic unit of analysis as opposed to census blocks, which are the smallest geographic unit that the Census Bureau publishes data on. The reason this decision was made was to improve the running time of the redistricting algorithms used. I found that when census blocks were used (for which there are thousands for the City of St. Louis), Auto-Redistrict and BARD took an extremely long time to run. As this project involved generating many boundaries using these tools, I made the decision to go with a slightly larger geographical unit to improve the running times of the redistricting algorithms. Should the same algorithms be run with census blocks as the geographical unit, more precise and ultimately better ward boundaries would be generated.

Another limitation of the project, which inherently results from the use of census block groups, was that there were many census block groups that crossed more than one neighborhood. This meant that when I labeled each census block group with the neighborhood that they were in, this was not the most accurate labeling as it represented the neighborhood that the census block group had its centroid in, even if the block group was not completely within this neighborhood. This meant that when it came to evaluating split scores for each neighborhood, all of my ward boundaries performed very poorly, even if the redistricting methods effectively reduced splits among the neighborhoods. To account for this, when I calculated the split scores for neighborhoods, I considered the neighborhoods as represented geographically by the attribute data of the census block groups, which was not the most representative of actual neighborhood boundaries. Given the size of census blocks, I am confident that if census blocks, not census block groups, are used as the geographic unit of analysis, such a problem will not arise.

It is clear that the use of census blocks, as opposed to block groups, will improve the generation of ward boundaries to better meet redistricting criteria. One potential direction for future research could be to examine how much this change actually improves the ward boundaries. Another direction for future research could be to evaluate even more redistricting software. While Auto-Redistrict and BARD are two of the best documented free and open source software currently available, there are a variety of other software, including web applications, that advertise automatic redistricting functionality. Future research could certainly analyze the performance of such software. Finally, while BARD was considered to be the least effective of the redistricting methods, this was certainly not a result of the actual software but rather how I used the software. Future research could investigate score functions and optimization algorithms so that one has a better understanding of how to use BARD and thus generate better ward boundaries using this software.

REFERENCES

Altman, M., MacDonald, K., & McDonald, M. (2016). From Crayons to Computers: The Evolution of Computer Use in Redistricting. *Social Science Computer Review*. https://doi.org/10.1177/0894439305275855

Altman, M., & McDonald, M. P. (2011). BARD: Better Automated Redistricting. *Journal of Statistical Software*, *42*(1), 1–28. https://doi.org/10.18637/jss.v042.i04

Engstrom, E. J. (2013). Gerrymandering and the Evolution of American Politics. In *Partisan Gerrymandering and the Construction of American Democracy* (pp. 1–18). University of Michigan Press; JSTOR. https://doi.org/10.2307/j.ctt1gk086k.4

Justin Levitt. (2010). *A Citizen's Guide to Redistricting* (2010 Edition). Brennan Center for Justice. https://www.brennancenter.org/sites/default/files/legacy/CGR%20Reprint%20Single%20Page.pdf

Luc Anselin. (2018, March 16). *Contiguity-Based Spatial Weights*. GeoDa. https://geodacenter.github.io/workbook/4a_contig_weights/lab4a.html#gal-weights-file

Ordinance 69185. (2012). City of St. Louis. https://www.stlouis-mo.gov/internal-apps/legislative/upload/Ordinances/BOAPdf/69185.pdf

Richard L. Forstall. (1995). *MISSOURI Population of Counties by Decennial Census: 1900 to 1990*. US Bureau of the Census. https://www.census.gov/population/cencounts/mo190090.txt

APPENDIX

Appendix A: R Script

The following script was used to generate ward boundaries using the BARD software package for R.

```
#Import libraries and set environmental vairables
library("BARD")
setwd("C:/Users/austi/Desktop/GISClinic/BARD")
library("maptools")

#Import shapefile and set number of districts
stl.plan <- importBardShape("block_groups")
numDists <- 14
#Generate initial plan using weighted k-means
wkPlan <- createWeightedKmeansPlan(stl.plan,numDists, weightVar="TotalPop")

#Define score function
calcPopScore(wkPlan, predvar="TotalPop")
calcContiguityScore(wkPlan)
calcBBCompactScore(wkPlan)
calcRangeScore(wkPlan,predvar="Black_alon",predvar2="TotalPop",targrange=c(.50,.6
5))
myScore<-function(plan,...){
combineDynamicScores(plan,scorefuns=list(function(x,...)calcPopScore(x, predvar="
TotalPop"),calcContiguityScore,calcBBCompactScore,function(x,...)calcRangeScore(x
,predvar="Black_alon",predvar2="TotalPop",targrange=c(.50,.65))))

}
# Apply optimization algorithm to yield improved plan
improvedwkPlan <- refineGreedyPlan(plan=wkPlan, score.fun=myScore, displaycount=N
ULL, historysize=0, dynamicscoring=FALSE,   tracelevel=1, checkpointCount=0, resu
me=FALSE)

# Export plan
exportBardShape(wkPlan, "C:/Users/austi/Desktop/GISClinic/BARD/BARDOutput.shp", i
d = "BARDPlanID")
```