Gerrymandering and Graph Partitioning: Approaches to Population Division Models

Chau Pham
cqpham15@earlham.edu
Department of Computer Science, Earlham College, 801 National Road West, Richmond, Indiana 47374

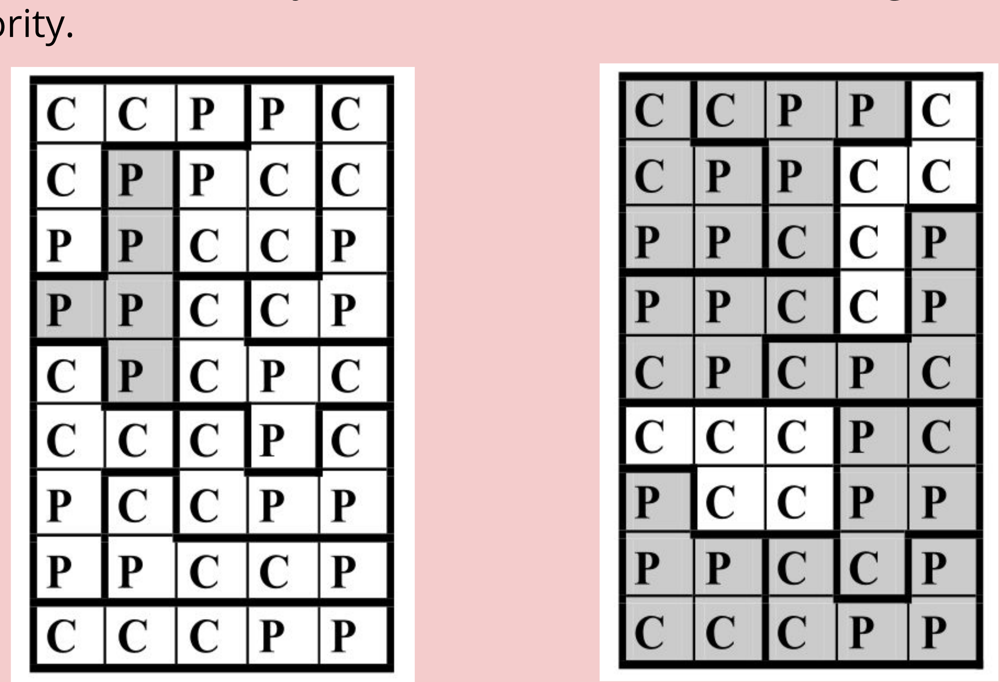
Abstract

Gerrymandering is the practice where voting district lines in a state are drawn to benefit one party over another. To combat this practice, computer-generated districting plans can be utilized. One way to generate districting plans is to treat the geographical area like a graph and model its population attributes, then divide up the area according to these attributes. In this project, a graph partitioning algorithm is created to generate compact electoral districts in several states. The algorithm generates several districting plans, then finds the most compact ones by evaluating several attributes measuring how spread out the population is in each district. The best districting schemes are then visualized using a mapping tool.

Introduction

In the American electoral system, a voting district is a geographical delineation used in the process of electing members of legislative bodies. The process of creating district lines is commonly done by the state legislature. Since the state legislature is controlled by a majority party, drawing district lines is thus a partisan process, which leads to bias, both subconscious and deliberate [1].

The point of gerrymandering is to waste as many of the opponent's votes as possible. Wasted votes are votes which do not contribute to a candidate's win; for instance, any votes above fifty percent in a simple-majority two-candidate race. If one party's votes are concentrated in a few districts and they win those districts by a large margin with lots of wasted votes, their opponent can actually win the rest of the districts and get a majority.



a b

Figure 1: Example by Dixon and Plischke:
(a) Party P wins 1 seat and party C wins 8
(b) Party P wins 7 seats and party C wins 2 [2]

A solution to gerrymandered districting is computer-generated district lines. Drawing district lines is analogous to a graph partitioning problem, where geographical areas are represented in the form of a graph G = (V,E), with V vertices and E edges. The solutions to the problem would be partition schemes which divide G into smaller sections that satisfy specific criteria. For this project, the districting application will focus on creating compact districts.

Software components of the project:

- Web scraping tool which retrieves the 2010 census data files for the specified state from the United States Census Bureau website
- ESRI Shapefile parser which reads the census data into their corresponding geospatial shapes and census attributes [3]
- Graph partitioning algorithm that uses these population attributes to calculate different district shapes and limits to generate several districting plans
- Mapping tool which takes the shapefiles and visualizes them

Methodology & Results

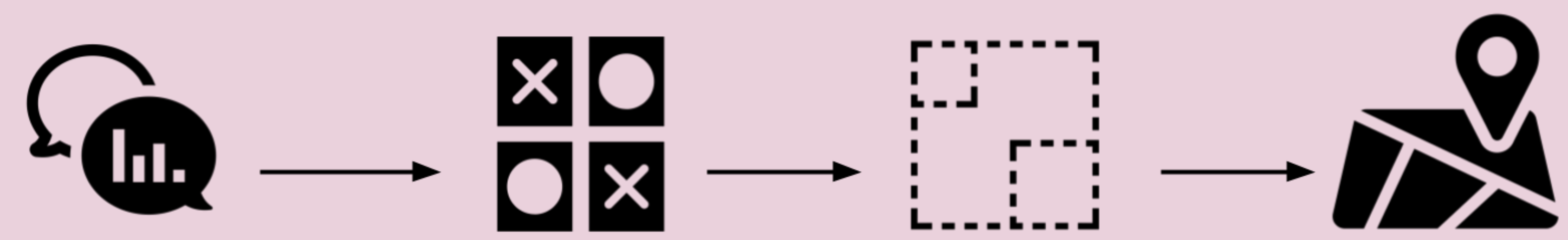


Figure 2: Design framework

To find the most compact districting plans amongst the plans generated, three attributes are calculated:

- the average distance between a voter and the center of their district in kilometers (km/p)
- the population difference between the least populous district and the most populous district (spread)
- the standard deviation of district populations (std)

PA_Congress		km/p	spread	std	gen
Run #1	**Best km/p**	31.80	6957	2107.86	2899
	Best spread	32.78	5166	1456.92	9606
	Best std	32.78	5166	1456.92	9606
Run #2	**Best km/p**	32.59	6159	1647.00	8971
	Best spread	32.61	4947	1609.04	6980
	Best std	32.59	5173	1365.19	9855

Table 1: Generations of Pennsylvania districting plans with the best values for compactness attributes

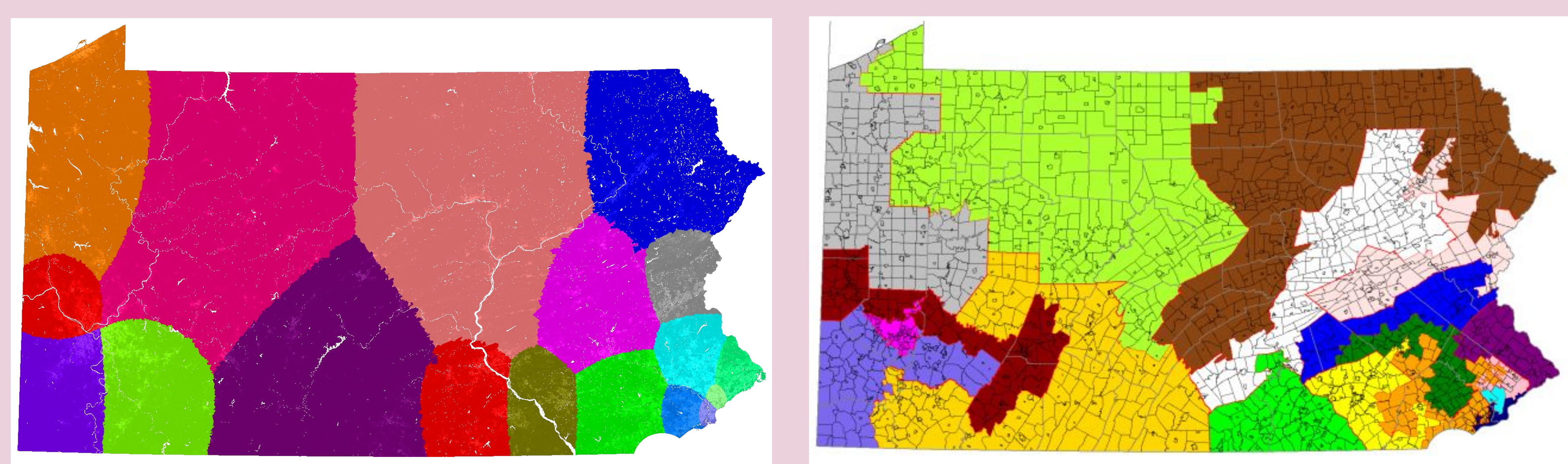


Figure 3: a) Computationally generated congressional district map of Pennsylvania
b) District map of Pennsylvania (before the 2018 court-mandated redistricting) [4]

CT_Congress		km/p	spread	std	gen
Run #1	**Best km/p**	22.26	5887	2320.42	779
	Best spread	22.35	1265	516.78	847
	Best std	22.35	1265	516.78	847
Run #2	**Best km/p**	22.37	6565	3175.03	695
	Best spread	23.62	635	272.47	3661
	Best std	23.62	635	272.47	3661

Table 2: Generations of Connecticut districting plans with the best values for compactness attributes

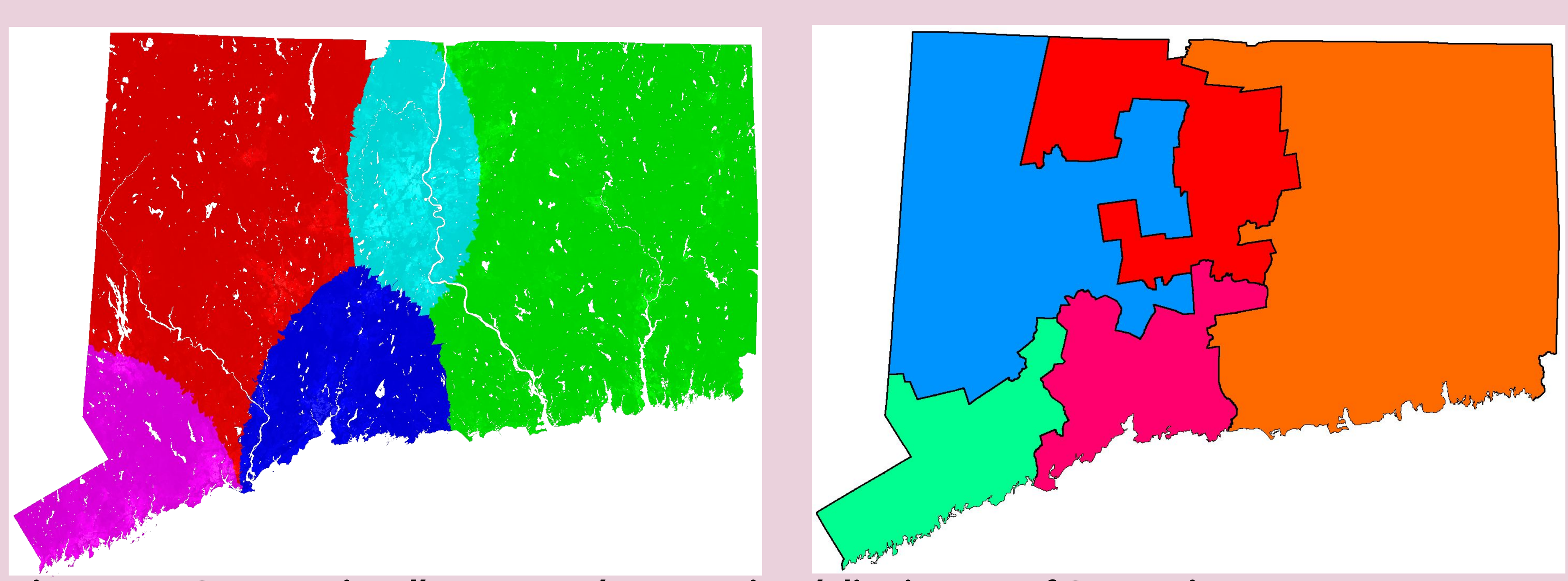


Figure 4: a) Computationally generated congressional district map of Connecticut
b) Current district map of Connecticut [5]

Data

One of the census data files is the **tabular blocks** and their boundaries for each geographical area [6]:

- The tabular block data file has the geographic coordinates of the **start** and **end nodes** for each block.
- This data connects to the census **edges** and **faces** data files.

The **edges** data include the **geometry** and **attributes** of each topological primitive edge:

- Each edge is delineated by a **start node** and an **end node**, and these edges are the boundaries that create the tabular blocks.
- Each edge has a unique **Line Identifier** value.
- Each edge also have identifiers for its **left** and **right faces**, which link to information in the faces data files.

Each **face** encompasses an **address range**, and include **population data** about households in that range:

- Each face and edge also has an identifier for which county and state it belongs in.

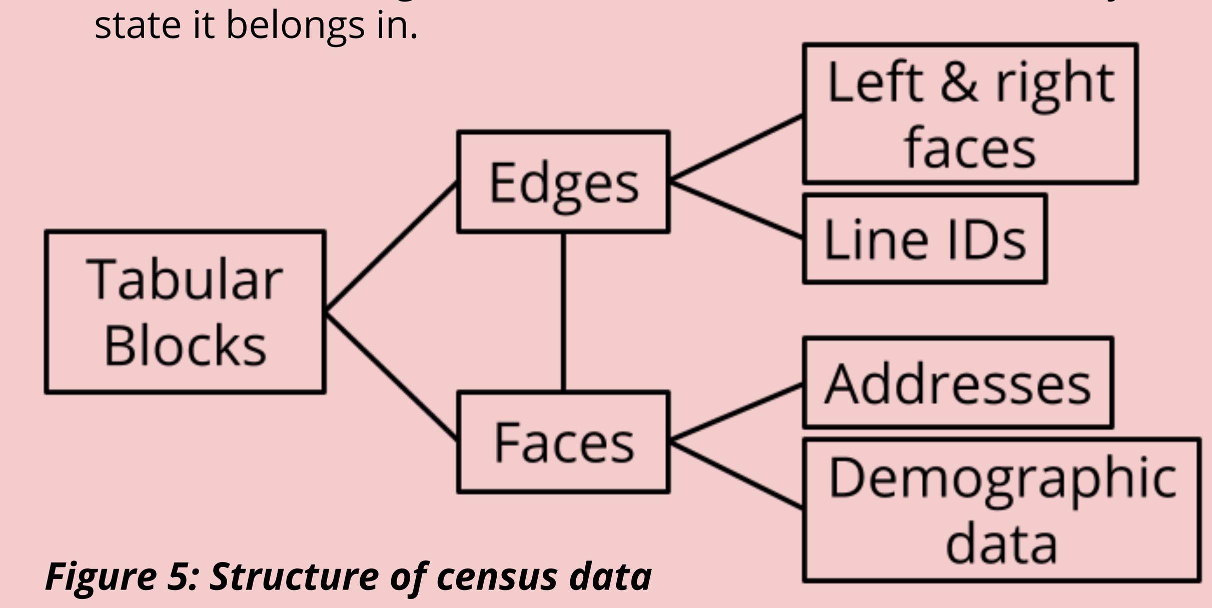


Figure 5: Structure of census data

Conclusion

Since some of the worst gerrymandering schemes achieve their goal by creating unusually-shaped districts, compact districts are less likely to be gerrymandered.

The generations of the districting plans with the best spread and standard deviation are usually the same because the plans with the lowest standard deviation have districts which the most equal population, which also minimizes the difference in population of the most and least populous districts.

The graph partitioning algorithm is very resource intensive for the amount of census data available, and it is not efficient to focus on every attribute of a region when generating districting plans. It is much more useful to focus only on a few attributes at a time.

There have been research questioning whether an unbiased solution will always exist for a districting plan or if gerrymandering even significantly affects elections at all [7]. These are major questions which should be addressed in future work.

References

[1] Yan Liu, Wendy Cho, and Shaowen Wang. 2016. PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation* 30 (2016), 78–92. https://doi.org/10.1016/j.swevo.2016.04.004. [2] N. Apollonio, R.I. Becker, I. Lari, F. Ricca, and B. Simeone. 2009. Bicolored graph partitioning, or: gerrymandering at its worst. *Discrete Applied Mathematics* 157, 17 (2009), 3601 – 3614. https://doi.org/10.1016/j.dam.2009.06.016 Sixth International Conference on Graphs and Optimization 2007. [3] Brian Olson. *Redistricter*. Bitbucket (2017). https://bitbucket.org/bodhisnarkva/redistricter/src/default/ [4] Image source: https://www.jurist.org/news/wp-content/uploads/sites/4/2018/03/PA_2011_map.png [5] Image source: Ashley Smith https://imgur.com/ELYurYm [6] William Macmillan, Todd Pierce, S. Fotheringham, and Peter Rogerson. 2013. *Optimization Modelling in a GIS Framework: The Problem of Political Redistricting*. CRC Press, 221–246. https://books.google.com/books?id=Hirdl1ZFE38C. [7] Clemens Puppe and Attila Tasnádi. 2008. A computational approach to unbiased districting. *Mathematical and Computer Modelling* 48, 9 (2008), 1455 – 1460. https://doi.org/10.1016/j.mcm.2008.05.024 Mathematical Modeling of Voting Systems and Elections: Theory and Applications.

Acknowledgement

The author would like to thank Dr. Xunfei Jiang for her assistance and feedback, and the Computer Science Department at Earlham College in the preparation and execution of this project.