Resource Apportionment by Quasi-Voronoi Graph Partitioning

J. Mac DeLay

Birmingham-Southern College
Birmingham, AL
Advisor: Dr. Jeffrey Barton

1 Impetus

It is the ambition and obligation of any democracy – any good one, that is – to ensure that its peoples' voices are heard. Among demonstrations and market behaviors, the vote is the most typical operation of civic opinion. However, the political system is not perfect because its executers are not perfect; several structural flaws prevent the government from perfectly describing public input on governance. Voting is sampling, and access to polling stations can significantly bias an election. The aim of this project is to investigate how a discrete analog to the Voronoi polygon can be used to improve precinct partitioning with optimal regard to the capacity of a polling location (or ballot drop box) and the convenience of all residents.

The method described in this paper partitions a Census map into geographical assignments to input precinct locations. Modern GIS tools address resource placement using location-allocation algorithms.[1] This powerful tool is not useful for this topic as we assume that the precincts cannot be moved. Instead, we will mimic how Voronoi diagrams are generated, sacrificing the perfect geometry to accommodate logistic and structural obstacles presented by other allocation methods. The objective is to describe a process that can build the perfect, or at least a nearly perfect, partitioning of a map without entertaining the immensely expensive process of checking and evaluating the possibilities. Included at the end of this paper is the code used to implement the algorithm below.

2 Definitions

A lot of this project covers topics across math disciplines and includes some new terminology that can get mixed up depending on the reader's prior exposure to the topic. Let's take a moment to get on the same page with some terms.

Definition 1 (Voronoi Diagram)**.** Given some plane and a collection of points within it, a Voronoi diagram partitions the plane into subregions that have exactly one generator point. These subregions must satisfy the property that every point inside each polygon is closest to that polygon's generator point. Such diagrams have many interesting properties; the most important one for us is how each polygon is convex.

[1] Schietzelt, T. H. and Densham, P. J. "Location-allocation in GIS". 2003.

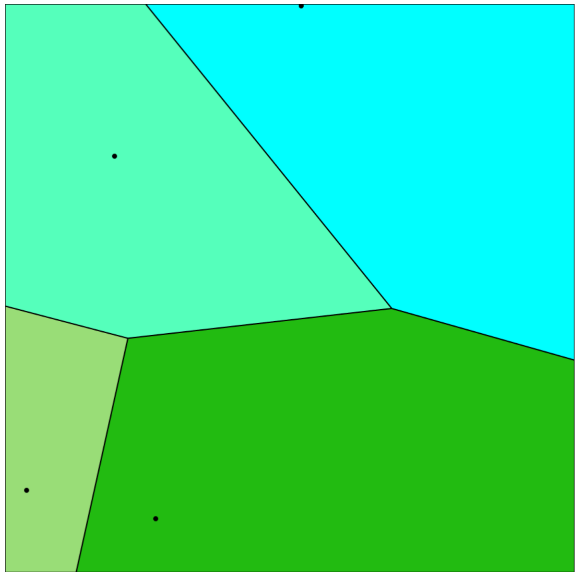

Figure 1: This is an arbitrary 2-D polygon with 4 generator points. If you pick any point in the plane, the shortest line between that point and any generator is the one that shares the same subregion.

Definition 2 (Graph)**.** A graph is a collection of objects, or vertices, and the edges that connect them. Vertex-labeled graphs have values associated with each vertex. Generally, edges represent subsets of relationships among all the points – a social network of people (vertices) could represent who is friends with whom (edges).

In most established political geometry, it is desirable to have "simple" shapes. In the past, quantifying simplicity helped flag maps that may have been drawn with foul intentions. The geometric property we want to borrow from Voronoi theory is convexity.

Definition 3 (Convexity)**.** The interval $I(u, v)$ between vertices $u, v \in G$ is the set of all vertices that lie on the shortest paths between u and v. That is,

$$I(u, v) = \{x \in V_G : d(u, x) + d(x, v) = d(u, v)\}$$

A subgraph H induced on G is convex if H contains the intervals on G between any two vertices in H. The disk $N_r(x)$ of center x and radius $r \geq 0$ is the set of all vertices of distance r from x.[2]

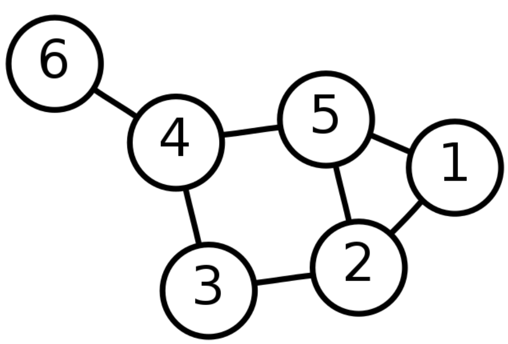

Figure 2: The subgraph [1,2,5] is convex, and the subgraph [2,3,4] is not. We would need to add vertex 5 to include the alternative shortest path.

[2]Bandelt, H and Chepoi, V. "Metric graph theory and geometry: a survey". *Contemporary Mathematics*

Convexity is desirable because it ensures that diagrams or subregions don't skip over blocks that are closer than other chosen areas.

Definition 4 (Weight)**.** Since we're looking at vertex-labeled graphs, the *vertex weight* is the number assigned to the point. The *subgraph weight* is the sum of products of each vertex weight and its minimal distance to the root vertex. The following is the definition of subgraph weight for a subgraph of n vertices:

$$\sum_{i=1}^{n} D(v_i, s)(\omega_i)$$

Definition 5 (Global Impedance)**.** We define global impedance as the sum of subgraph weights generated from the original graph.

Definition 6 (Volume)**.** A subgraph's volume is the sum of the vertex weights in the subgraph.

Definition 7 (Global Disparity)**.** If we assume that polling sites accommodate roughly the same number of people, we need to make sure the volumes aren't outrageously irregular. This is the measure for how irregular the volumes are.

$$\frac{1}{k}\sum_{i=1}^{k}(Vol_i - \frac{Total\ Pop}{k})^2$$

Future work can be spent altering this project's parameters to optimize based on personalized volume limits.

3 Simplest Case: Euclidean Distance Without Capacity

Let $P = \{p_1 \dots p_n\}$ be a set of points, or generators, on a closed plane. We define the Voronoi partitioning of the plane as a set of polygons that satisfy the following:

$$V(p_i) = \{x : |p_i - x| \leq |p_j - x| \ \forall\ j \neq i\}$$

If the goal is to partition geographically, then we're done! Given any point inside the plane, the generator closest to that point corresponds to the polygons to which the point belongs. The exception to this property is that points that lie on the polygon edges have more than one closest generator.[3] Assuming that each location could serve any number of residents and that Euclidean distance is the best measure for impedance, getting precincts as close to a map's Voronoi partition would be perfect. We can summarize all qualities of the partitioning by measuring how closely the real district matches its polygon (Figure 1).

[3] Joseph O'Rourke. *Computational Geometry in C.* Second Edition, 1998, p.155-192, DOI, mathematical review number.

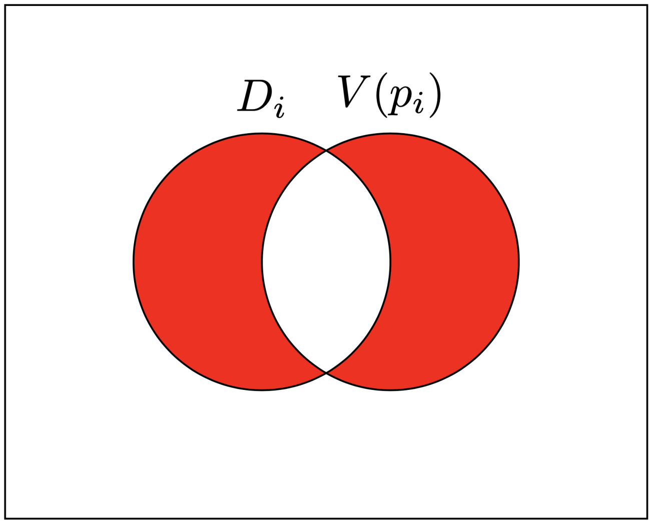

Figure 3: We seek to minimize the nonoptimal area represented as red space above.

However, we shouldn't stop here. What if a precinct that historically accommodates one hundred residents is now assigned one thousand? Suppose a river turns a fifty-foot commute, as a crow flies, into a twelve-minute drive. Where do you go if the polygon edge bisects your property? Does your neighbor with the same issue go to the other location? These realistic hurdles could make election operations *worse*.

4 Variable Rate: Existing Work

Existing work already claims that once population layers and intricate geographies are introduced, simple Voronoi diagraming is insufficient for map partitioning. Inspiration for this project comes from a paper addressing redistricting by way of nuanced Voronoi partitioning.[4] Given generator locations, they try to construct partitions with equitable populations. Without much rigid documentation, they propose a clever, nearly-continuous, geometric solution. One method of Voronoi diagraming draws expanding circles around each point until circles touch, delineating the border edges. This method alters the process by letting less-populated regions expand faster than those with more residents. Computationally it translates as follows, where t is a timestep in map V_i acquisition and $f : \mathbb{R}^2 \to \mathbb{R}$ is the accumulation of residents in the area (x, y):

$$V_i^{(t)} \subset V_i^{(t+1)}$$

and

$$\int_{V_i} f(x,y)dA = \int_{V_j} f(x,y)dA$$

With this project, we take a different approach to solving map partitioning inspired by the ideas proposed by the Washington cohort. They briefly mention that geographical features lend some unexplored challenges with their partitioning. Basing impedance on block adjacency instead may yield a more reliable sense of accessibility rather than the hope that raster cells in images have annotated structural and geographical features properly. Our restructuring of the question will also provide the concrete properties by which we judge quality partitions such as graph weight.

[4]Svec, L.; Burden, S; and Dilley, A. "Applying Voronoi Diagrams to the Redistricting Problem". *The UMAP Journal*, 28.3 (2007)

5 Introducing Discrete Distance

The Washington cohort's model is discrete already, but as picture pixel density increases, it approximates continuous geometry very closely. In Figure 2[5] we see that the Census Bureau already has an atomic way to look at the composition of the country's geography.

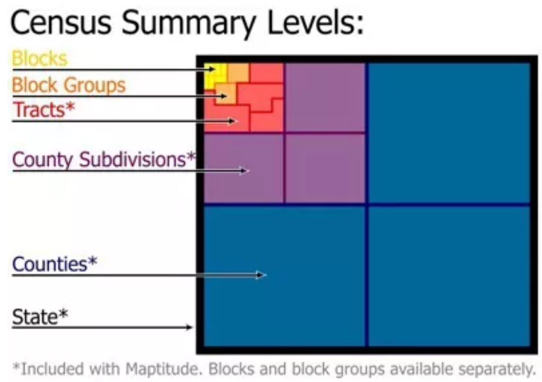

Figure 4: A simplified representation of how the Census Bureau classifies space through the country. There are over 11 million census blocks in the US, and some of them don't have any residents!

Let's construct a graph representing census block adjacency – our blocks, the nodes, will share an edge on the graph if they share a border on the map [6].

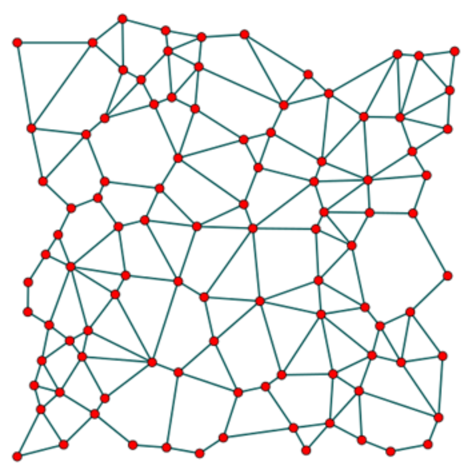

Figure 5: Example of what a city's blocks would look like as a graph. Some areas have few neighbors, and others, many.

It's important to note that with a graph, we no longer consider Euclidean space. Because the graph represents related elements, we define our distance as the smallest number of block traversals needed to get from the block of interest to the precinct.

[5]https://www.caliper.com/maptitude/census2000data/summarylevels.htm
[6]https://plus.maths.org/content/maths-minute-graphs-and-handshaking-lemma

6 Ways to Partition Discrete Precincts

Let us now discuss some processes that could be used to generate partitions of a graph given certain generator nodes. Each of the following methods prioritizes some qualities that are reasonably desirable for a civic apportionment of residents. Going forward, we use the graph depicted in Figure 6 to discuss partitioning processes.

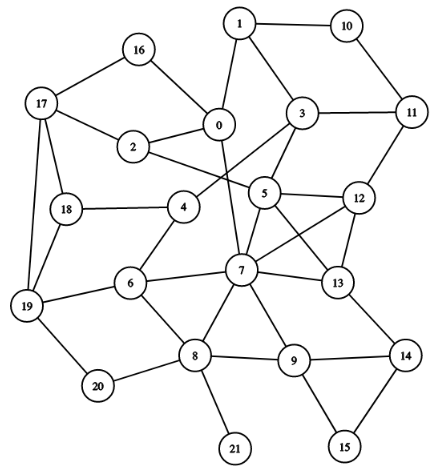

Figure 6: Arbitrary graph with weighted nodes

6.1 Minimal Disparity

There are many different ways one could partition a graph such that all of the volumes are optimally similar. While finding a partition this way is elementary, it is extremely unlikely that this process returns convex – or even contiguous – partitions as there is no way to direct how the subgraphs look.

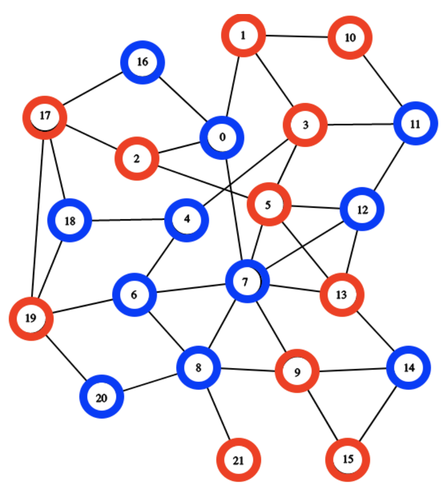

Figure 7: A partition of the graph with the minimum possible difference in volume. This is not contiguous and should not be an example of good zoning.

Global Disparity: 0.75

Global Impedance: 723

6.2 Minimal Impedance

As with disparity, it is just as easy to optimize the impedance of the graphs. In the example below, the subgraphs are generated as follows: for each non-generator vertex, do a breadth-first search until it reaches a generator, merge (if it is equally close to several generators, assign arbitrarily). Now we have a partition with minimal impedance to residents. However, this process has a bias against densely-populated neighborhoods. Unlike the disparity method, the only importance is proximity, not capacity. Arbitrary assignment can yield an unnecessarily high range of disparity.

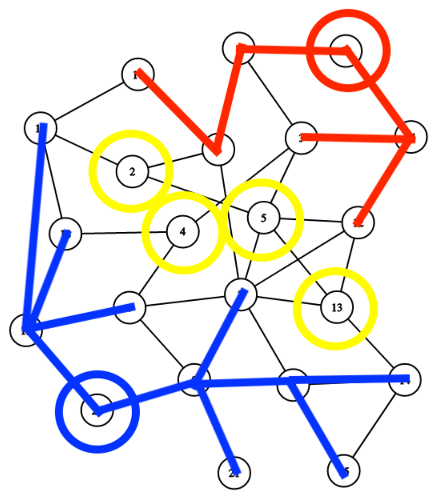

Figure 8: A partition of the graph with the minimum global impedance. Border vertices are yellow. Merging all of the border nodes with blue would significantly increase the disparity of the graphs.
Global Disparity: ??
Global Impedance: 432

6.3 Maximal Convexity

This method is the closest analog to the Voronoiesque method described in the existing work, only here we take blocks in increments instead of map raster pixels. It generates subgraphs using breadth-first searches originating from each generator. A turn to acquire neighbors is granted to the incomplete subgraph with lowest volume. In metric graph theory, a *disk* centered at v with radius r is as follows:

$$disk_r(v) = \{x \mid dist(v, x) \leq r\}$$

Since breadth-first subgraph generation returns disks by definition, it would be nice if we could prove that they always returned convex subgraphs. Unfortunately, they do not. We also know by counterexample that our dueling BFS graphs do not either. However, it is still useful to study because they might return an optimally low number of *excluded* vertices, vertices that lie on a shortest path but are not included in the subgraph. Evidence for this would be immensely useful as it translates to our final algorithm.

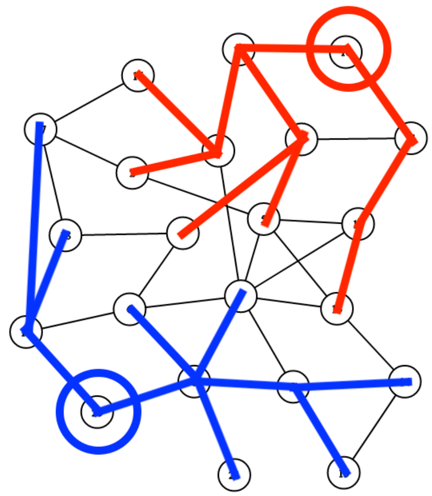

Figure 9: The issue with this method is the erratic a nature of the volumes by each turn. It would be nice to fine-tune the accumulation of vertices by volume.
Global Disparity: 38.5
Global Impedance: 432

7 Composite Partitioning

This algorithm is influenced by the breadth-first approach to building a minimal spanning tree (for our case, an optimal subgraph). We modify it such that there are several roots with disjoint descendants as well as pausing conditions based on accumulating volumes. Given that the map must be partitioned for k generators, let G be connected graph with vertices V_g and edges E_g. We also define $S \subset V_g$ as the set of vertices identified to be the generator locations. These will be the roots of the trees that lend nodes for partitions of G.[7]

$U \leftarrow V_g - S$;
$T_i \leftarrow \{s_i \in S\}$;
$level(T_i) \leftarrow 0$;
while $|U| \neq \emptyset$ **do**
    $M := min_{vol}(T)$;
    $A \leftarrow \{v : (v, v_i) \in E_g; D_{min}(v, root(M)) = level(M) \, \forall \, v_i \in U\}$;
    if $A = \emptyset$ **then**
        $level(M) \leftarrow level(M) + 1$;
        $A \leftarrow \{v : (v, v_i) \in E_g; D_{min}(v, root(T)) = level(H) \, \forall \, v_i \in U\}$;
    end
    $merge(M; max(A); (max(A), max(A)_{-1})$;
    $U \leftarrow U - max(A)$;
end

Algorithm 1: k-Partitioning with Equitable Volumes

[7] The Python script partitions in a slightly different way. In short, it lists the priority ranking of all vertices and incrementally acquires them, ignoring taken vertices.

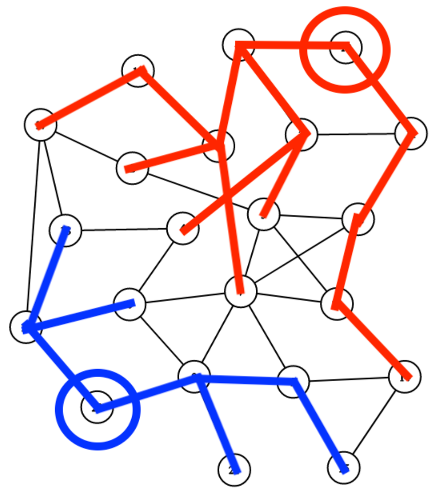

Figure 10: The output of our algorithm.
Global Disparity: 0.75
Global Impedance: 487

At each level, we take heavier vertices before lighter ones. This gives us the ability to "fine-tune" the disparity as mentioned earlier. For large graphs, we will find that because of this ordering, the probability that any additional vertex has a bigger impact than the last is smaller than the complimentary case. This fact also helps to optimize weight as the vertices that will have the greatest impact on weight take precedence.

8 Properties, Evidence, and Limitations

Proposition 1. *Let simple, connected graph G with generator subset S be partitioned as described. Then every non-generator vertex is mapped to one and only one subgraph.*

Proposition 2. *Let simple, connected graph G with generator subset S be partitioned as described. Then each subgraph is connected.*

Proposition 3 (Optimality)**.** *Let simple, connected graph G with generator subset S be partitioned as described with resulting s_0 exclusions, global impedance i_0, and global disparity d_0. Then there is no alternative set of contiguous partitions – one generator per partition – where $s \leq s_0$, $d \leq d_0$, $i \leq i_0$.*

Since this project lacks proof, let's try to synthesize some convincing support for this process. There are two tiers of reasoning: optimality and efficiency. If our propositions are proven true, then this is all we need. Given the parameters for what makes a perfect partition (Prop 3), this is algorithm will assign residents to resources perfectly. Suppose these propositions are not proven. This algorithm belongs to the "greedy" family. That is, choices made as each timestep are, in that moment, the optimal ones. Our regions are not disks, but they should optimize exclusions; and the disparity and impedance may not be the absolute lowest, but they should be among the lower partitions'. If a bright mathematician, far brighter than me, were to support its greediness, they would present a case that contradicts our optimality proposition and prove that it cannot exist.

The other important quality is efficiency. To check each partition by brute force, you would need to build every possible case and run tests on it. If we wanted k partitions on a graph of n vertices, the number of potential partitions is a Sterling number of the k kind:

$$S(n,k) = \frac{1}{k!}\sum_{i=0}^{k}(-1)^i\binom{k}{i}(k-i)^n$$

These numbers are comically large. Even if we chose to narrow the set of potential partitions, calculating the number of excluded vertices among the concave subgraphs would be redundant and recursive beyond reason. An analytical constructor is the only sensible option.

Now, let's examine a limitation. One vulnerability is entrapment. Consider a very simple case where one generator vertex is adjacent only to other generators. Then as the algorithm proceeds, the disparity will only increase with no hope of improvement. As this dilemma scales up, we are left with the concern that when generators are more densely clustered, they may become entrapped. With limited investigation, we assume that the only solutions to entrapment are to either accept suboptimal – by our standards – partitions or to relocate generators.

9 Future Work

One might find a solution to entrapment by revisiting Voronoi theory. Let's first discuss adding generators. We know that the entrapped location will have a smaller volume than desirable; hence, there will be locations that are too large. Similar to the solution to greatest interior circle, perhaps adding a generator in the interior of the greatest-volume triangulation could improve the system. Now, relocation. In the realm of redistricting, some scholars redistrict by starting with generators, computing the polygon centroids, and generating again from the centroids. In our analogous solution, one could investigate employing subgraph centers to prevent entrapment. This could potentially de-cluster problematic areas. This, though, would depends heavily on the success of the work thus far.