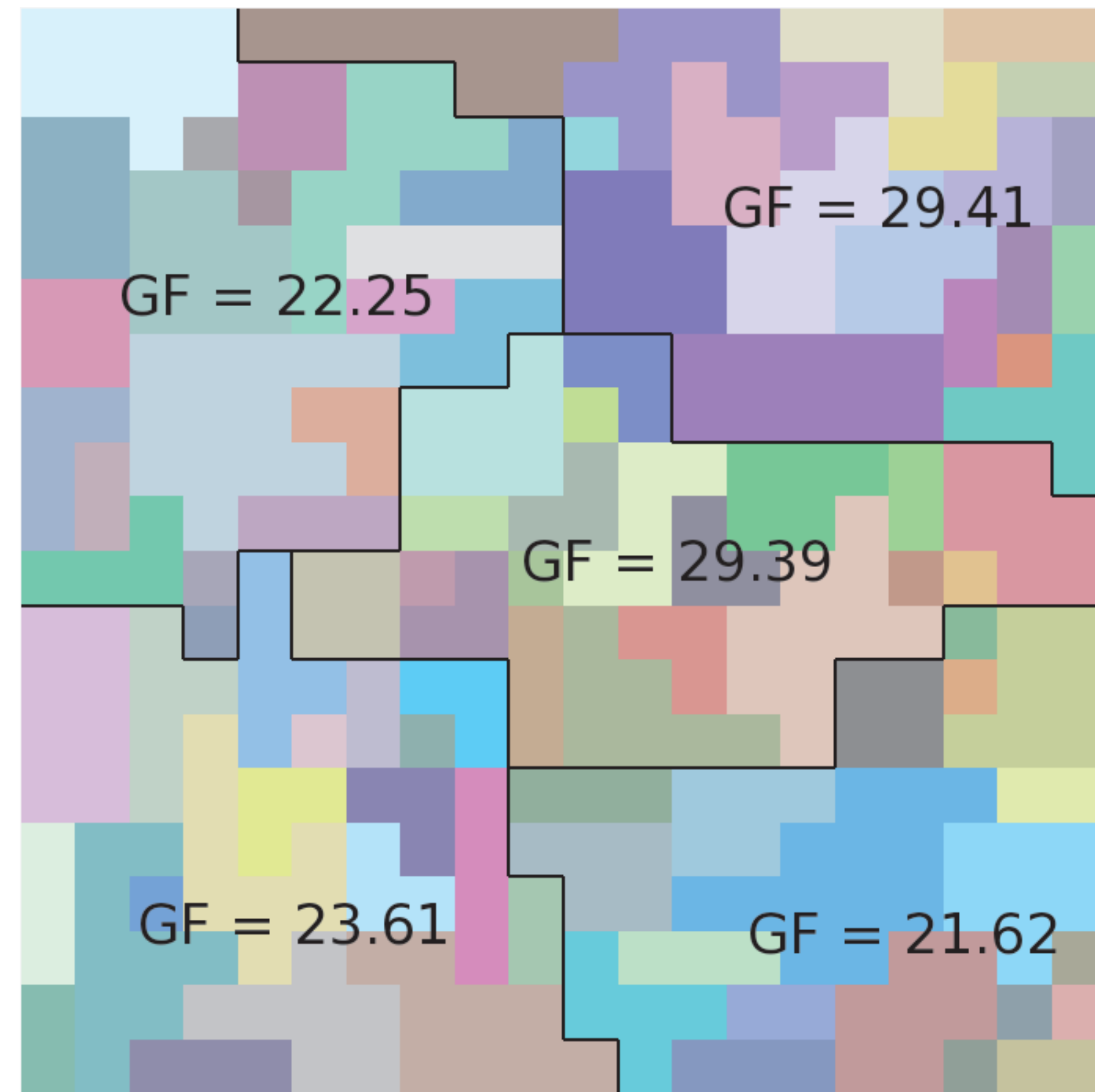


# Redistricting Utopia: Using Computers for Impartial Redistricting Plans

Walker Lindley and Dr. Randy Bentson



Synthetic precincts in color with black outlines defining the 5 districts created by our redistricting algorithm. Each district is labeled with its GF value.

## Apportionment and the Gerrymander Factor (GF)

There are two important measures of fairness for any redistricting plan: apportionment and compactness. Apportionment is the principle of "one person - one vote" and it ensures that the vote of a person in one area carries about the same weight as a person in another area. This is accomplished by making districts with roughly the same number of people in them.

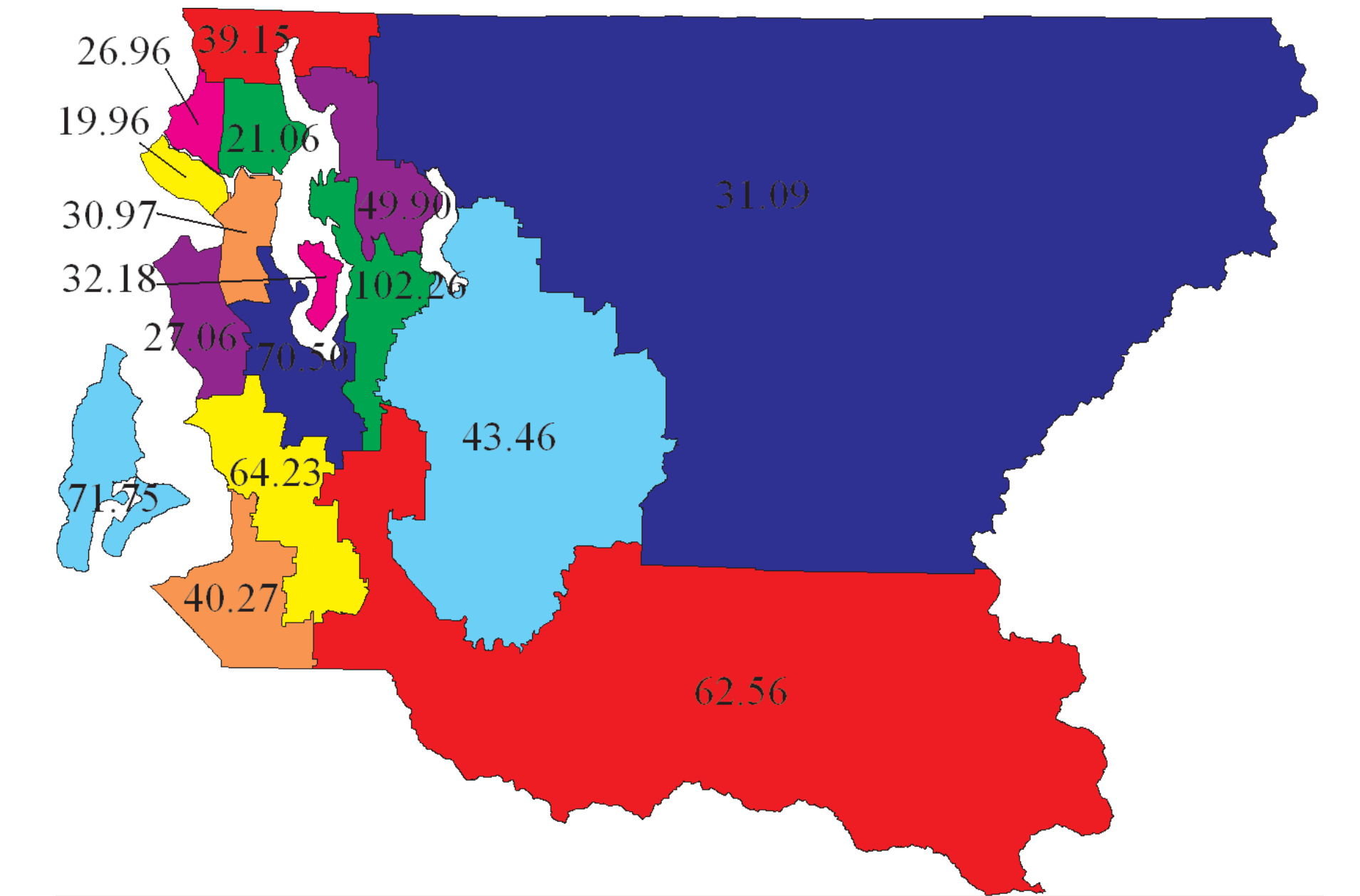
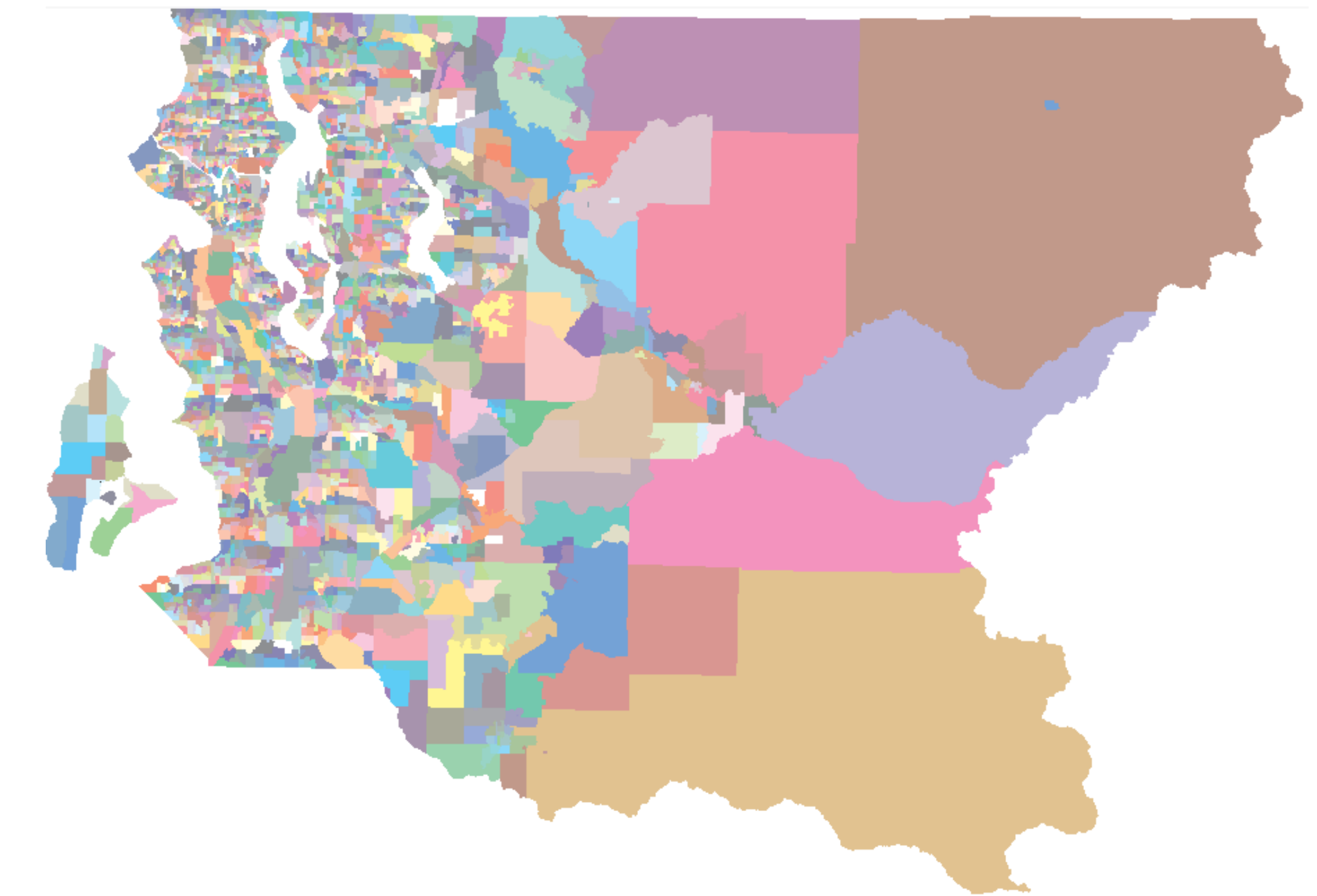
Measuring apportionment is pretty easy, but there must also be a good measure of compactness. First we need some criteria for this compactness metric. It will be applied to many districts, so it should be something that can be calculated quickly and easily. Some districts will certainly be shaped very oddly, so we need to make sure it works for all polygons, especially irregular ones. Finally, we want districts of the same shape to have the same value regardless of size (a square is a square, after all), so our metric should be scale invariant.

We have named our solution to this problem the Gerrymander Factor or GF, for short. The GF of a polygon is defined as the square of the perimeter of the polygon over the area of the polygon. GF meets all three requirements discussed above and is so easy to calculate, it can even be done by hand. Below is a diagram detailing the math behind calculating the GF for a few common shapes, as well as for a few precincts from King County. Higher GF values are worse, so the circle is most compact out of the three shapes presented below. In fact circles have the lowest GF of any shape, so 12.5 is the best GF we can have. However, it is impossible to completely cover any area with circles, so hexagons appear to be a better answer. Hexagons can completely cover an area without leaving empty space and would produce districts with a GF of 13.8; this appears to be the practical best GF for our districts. In practice, it is difficult to create districts that approximate any regular polygon and we saw districts with GF values in the 20's and 30's when we ran our program. When compared to the GF values of the current King County Council districts (to the right), we can see that our algorithm produces districts more compact than those seen in the real world.

Our basic assertion is that districts created with equal apportionment and compact form leave less room for partisan mischief. By optimizing these two values, we believe we can create fair and impartial redistricting plans using computers. This would prevent interference from partisan politicians thereby strengthening our democracy. Extensive work has been done on optimization problems in computer science and with these metrics, redistricting becomes a two-variable optimization problem (albeit a very complex one). While optimization problems are not trivial to solve, there are efficient ways to approximate the answer. We believe that a technique borrowed from the field of artificial intelligence will allow us to do that. Read on below.

## Application to Physical World Data

If an algorithm similar to ours was adopted by governments at all levels, it could effectively eliminate gerrymandering and would dramatically increase the quality of apportionment. Higher quality apportionment directly leads to the weight of everyone's vote being more equal. It would lead to representatives that better represent their constituents. Eliminating gerrymandering would help to prevent one party rule and would likely lead to less divisive politics. The two parties would have to work more closely together because the option of eliminating the other party would no longer be available. Other systems, such as Instant Runoff Voting (IRV), have been suggested to solve this problem, but they have significant drawbacks. Implementing IRV would require a major overhaul of current election law as well as teaching citizens how to vote under the new system. Our approach has the advantage that it can be used with no change to current law.



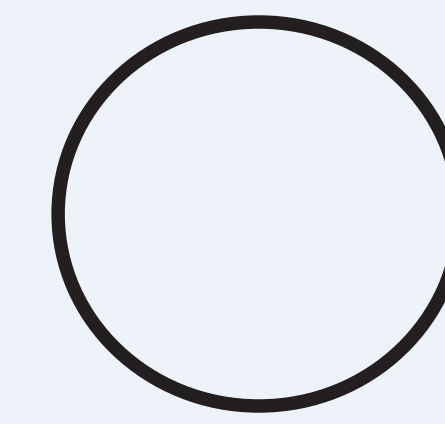
Top: Map of the precincts in King County.  
Bottom: King County Council districts labeled with their GF values.

## The Problem

For a representative democracy to work, the system for electing representatives must be fair and impartial. The two-party system is one answer to this problem and is usually a very effective one. However if one party manages to get a majority, they can drastically alter the fragile balance of power the two-party system relies upon through a process called gerrymandering.

Each county in the country is divided up into voting precincts. These are the smallest political unit other than the voter himself and they usually have a few hundred people in them. It is important that precincts have a similar number of people to ensure the principle of "one person - one vote", so the geographic size of precincts decreases in heavily populated areas, such as cities. Precincts are combined to form districts that elect various positions in the local, state, and federal governments. If politicians know that certain precincts tend to vote for their party, then when it comes time to redraw the districts they can shore up support for themselves by moving like-minded precincts into their district. This is gerrymandering and it has dire implications for the fairness and effectiveness of our political system.

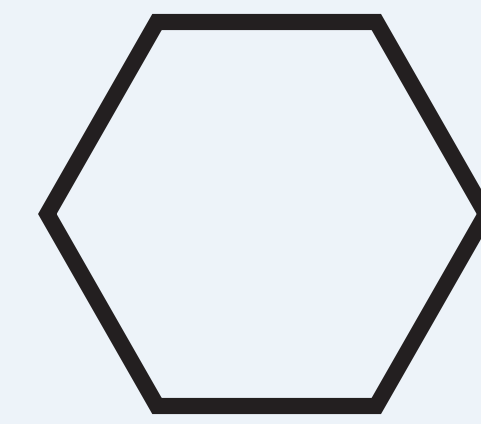
## GF Values for Sample Shapes



$$A = \pi r^2$$

$$P = 2\pi r$$

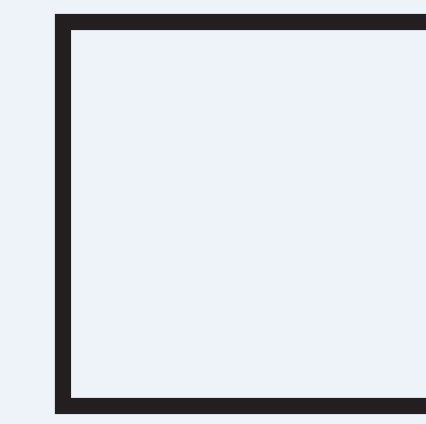
$$GF = \frac{(2\pi r)^2}{\pi r^2} = 4\pi = 12.566\dots$$



$$A = \frac{3\sqrt{3}x^2}{2}$$

$$P = 6x$$

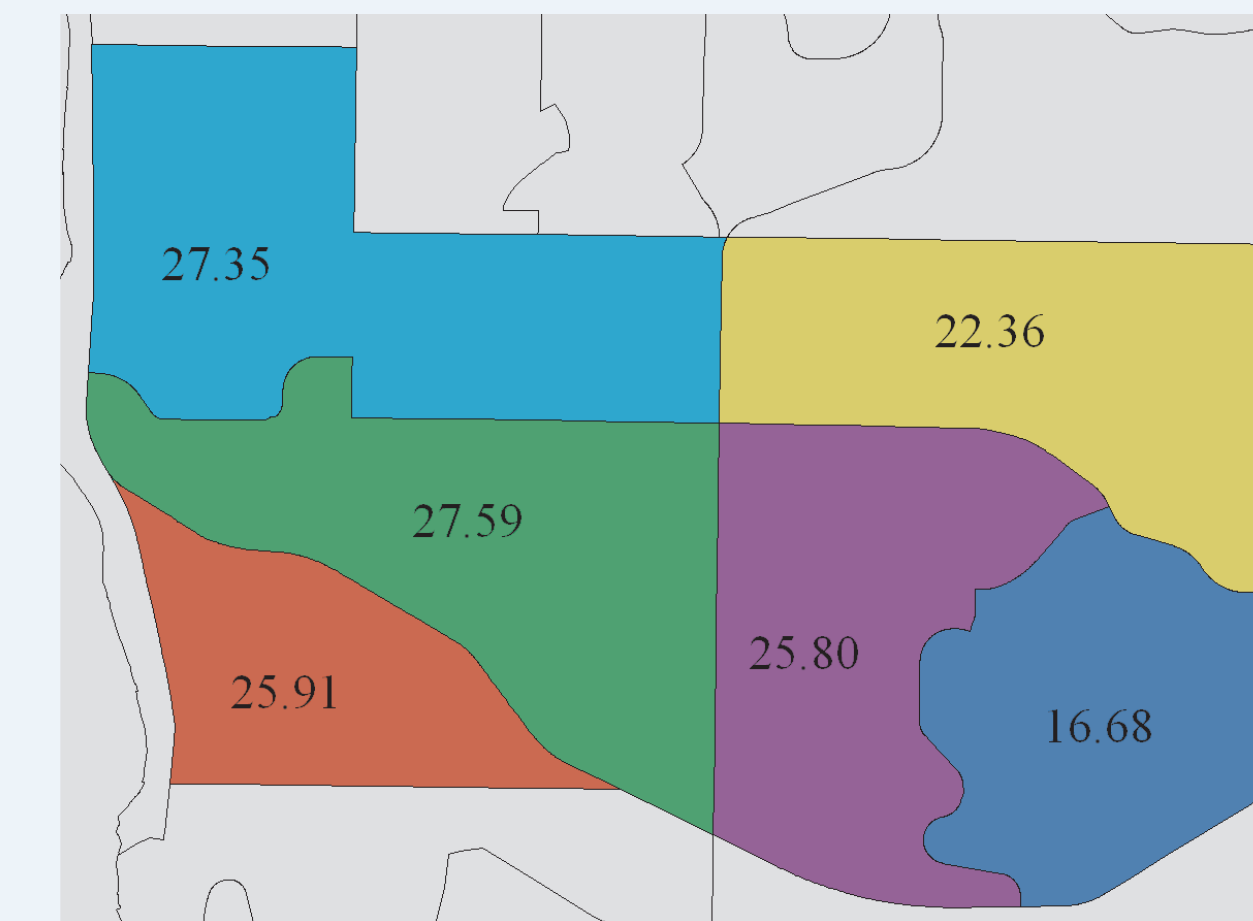
$$GF = \frac{(6x)^2}{\frac{3\sqrt{3}x^2}{2}} = \frac{24}{\sqrt{3}} = 13.856\dots$$



$$A = x^2$$

$$P = 4x$$

$$GF = \frac{(4x)^2}{x^2} = 16$$



A few sample precincts from King County with their GF values labeled.

## Genetic Algorithms

Genetic algorithms are a technique used in the field of artificial intelligence to solve especially difficult problems. For a problem to be solved using genetic algorithms, we must be able to encode the solution as a bit string (i.e. a sequence of ones and zeroes) that will represent a genome or DNA. First, many random bit strings are created to represent a starting population. This is analogous to a population of animals with bit strings instead of DNA. Next a process similar to natural selection is applied to the population repeatedly to "evolve" it and generate better solutions. For natural selection to work, we must have some idea of the relative quality of the solutions; for this we need a fitness function. This function assigns a numerical value to each genome based on the apportionment and GF metrics discussed above. The natural selection process starts by picking two good genomes from the population and "breeding" them using crossover and mutation. We used single-point crossover, so a random point along the bit string was chosen and DNA was taken from both parents to fill out both sides of the child's DNA (the length of the bit string stays the same). Then the child genome is mutated by flipping a few of its bits (1's become 0's, 0's become 1's). Finally, the child genome is added to the population and one of the bad genomes in the population is removed.

The quality of the genomes in the population increases over time as this process of reproduction with natural selection is repeated many times. We usually let the program run for several days at a time to achieve good results and the algorithm will tend to produce a better answer the longer it runs. Unfortunately because of the randomness involved we cannot prove that it will always give the optimal solution or even that it will generate better solutions over time. This means that some starting populations will cause the program to quickly find a decent solution and then not advance at all while other starting populations may start slowly, but eventually come to a better solution.

<u>Crossover</u>		<u>Mutation</u>
Parent 1: 110010101001010111		Child: 11001010100110010101
Parent 2: 01011100010110010101	→	Mutated Child: 11011000101110011101
Child: 11001010100110010101		
Genome added to population: 11011000101110011101		

## References

1. Forman, Sean L. *Congressional Redistricting Using a TSP-based Genetic Algorithm*. Proc. of the Genetic and Evolutionary Computation Conf., 9 July - 13 July, 2002, New York City, New York.
2. Tinsdale, Elkanah. "The Gerry Mander". Boston Weekly Messenger 1812.
3. Whitley, L. Darrell. *Foundations of Genetic Algorithms 2*. San Mateo: M. Kaufmann Publishers, 1993.



The original district described as a "Gerrymander" in this 1812 political cartoon<sup>2</sup>.

## Problems with Physical World Data

To the top left, you can see that our algorithm works well on synthetic data. However, applying our algorithm to real world data is much more difficult. Our algorithm relies on precincts being adjacent to each other for the encoding scheme (bit string) we use in the genetic algorithm. While this is not a problem for the vast majority of precincts in King County or anywhere else, there are a sizeable number of precincts that border a body of water such as a lake or river. Therefore, we must be able to determine which precincts are adjacent to each other across such bodies of water. There are many ways to do this and all of them are arbitrary in their own ways. The option we choose is to fill in the bodies of water with many small, synthetic precincts. All of these synthetic precincts cause the bodies of water to effectively disappear, making the map no different than the synthetic data that we already know the algorithm works on.

Another problem we ran into is that the data itself is not always accurate. We were forced to limit our focus to the King County data because the statewide data was a patchwork of county data that did not precisely match at the edges. This left many small slivers of empty space between counties and, even worse, areas that apparently belonged to two counties. These discrepancies are not ones that can be solved algorithmically; instead problems such as this must be solved in the political arena.