

# ”Is Gerrymandering Bad or Am I Just Bad at Gerrymandering?”

Mirae Sunny Kim mk34442 and Joel Walsh jaw5744

December 9, 2019

## 1 Introduction

The particularly American incarnation of redistricting is often referred to as Gerrymandering. The term takes its name from a portmanteau of Gerry and salamander. In February of 1812 Massachusetts governor Eldbridge Gerry signed a bill allowing legislators to essentially draw their own districts. At an 1812 dinner party put on by the federalist party, one such absurdly shaped district earned the nickname the gerrymander (Dean, 1892). Throughout the last century, advances in technology have allowed politicians to draw these districts with ruthless precision. Laws governing the process vary from state to state, and the Supreme Court appears poised to allow states to devise their own regulations for the process. This hits congressional districts the hardest, especially in the elections directly after a US Census, when information on voters is most accurate.

## 2 Background on use and methods

Gerrymandering is not a clandestine plot, whose objectives are buried deep in the recesses of some hidden lair. The Redistricting Majority Project (REDMAP), started in 2008, expressly states their purpose on their website as being, to create 20-25 new Republican Congressional Districts through the redistricting process over the next five election cycles, solidifying a Republican House majority. (REDMAP, 2019) This project proved to be wildly successful in subsequent congressional elections. A prime example is Michigan, where Democrats represent roughly 50 percent of all voters, and yet 9 of the 14 congressional seats went to Republican candidates. This is not to say this phenomena exists solely to benefit Republicans, as Democrat-leaning states Maryland is generally regarded as one of the most gerrymandered states in the union.

Redistricting generally follows one of two different aims when minimizing majority power, packing and cracking. Packing refers to the process of putting as many of one party as possible into a district, in an effort to minimize the spread of their influence. Cracking refers to breaking up a majority population into several districts, so as to help slight majorities form of the opposing party. Our program can be thought of as

inadvertently achieving both aims; cracking apart majority populations when necessary and packing majority voters into densely populated districts to dilute their influence.

### 3 Conception and tabulation method

The next speed bump was the Population minority rules method. This method caused us trouble for a while. We first sought to understand the canonical dynamic programming example, the fibonacci sequence. We noticed that the fibonacci programs we found online strictly defined a more directly defined base case for

$$n = 0, 1, 2$$

. (This is due to the first  $n=1$  and  $n=2$  numbers being 1 and 1.) After that,  $\text{fib}(n-1) + \text{fib}(n-2)$  could recursively solve for any value of  $n$ . Figure 1 shows a diagram found on the website Geeks for Geeks to show how often a recursive call happens for  $\text{fib}(5)$ . This led us to think about the minority rules method in a similar fashion. Similar to how  $\text{fib}(1)$  and  $\text{fib}(2)$  represented base cases or dead ends, the highlighted portion in Figure 1 revealed our own dead ends for the minority rules class.

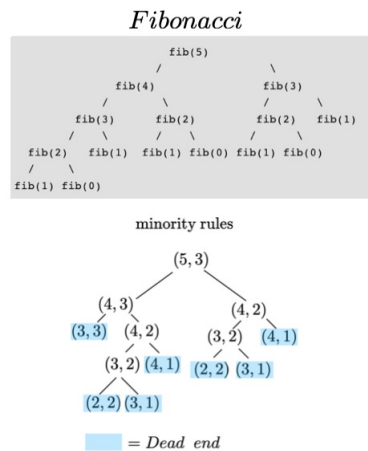


Figure 1: Figure 1: Fibonacci and minority rules

What followed this identification was an organic processes which can best be characterized as tabulation. We started from the bottom and worked our way up through the dead ends, solving the subproblems at the beginning. We created a blank matrix of Districting objects, where we could store each optimal configuration. In terms of rows x columns, our matrix was configured as  $\text{int}$  population (number of voters total) x  $\text{int}$  ndistricts. We started with the case of  $1 \times 1$ . This was easy enough, as theres only one way to partition this. We then realized that every other diagonal element where population equals ndistricts could be divided up in a similar fashion, with each voter representing a district. Our matrix would essentially be empty in the upper right half of

the matrix, where  $\text{ndistricts} \leq \text{population}$ . This makes sense, as its impossible to split four voters into five districts, and so on. The first column of the matrix, the case where ( $\text{population}, \text{ndistricts}=1$ ), was also simple, as the lean of only one group is simple to calculate. This left us with one base case to hard code, (3,2). This was accomplished by comparing the rule of the Districting object with the first two voters in one district as opposed to the last two voters in one district.

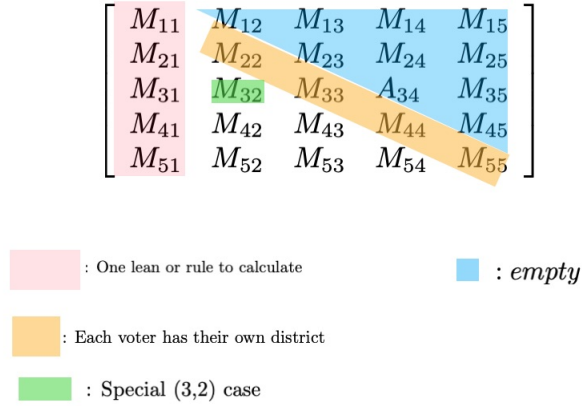


Figure 2: Figure 2: Tabulation matrix, vector of vectors of Districting

## 4 Conception and tabulation method

This tabulation procedure led to some unintended problems, however. This problem differs from the fibonacci problem in one key regard. Fibonacci numbers have definite answers to subproblems; and these sub-problems were slightly more ambiguous. At each stage of tabulation, we were telling our program to make a value judgement. For example, take the case of splitting 11 voters into 5 districts, as given in the textbook. You can either take the case of ten voters and five districts, and add the 11th voter to the last district, or take the case of 10 voters and four districts and add the 11th voter to its own district. Tabulation will work fine, so long as one of the combinations (10,5) then add to last or 10,4 then extend with new, has a more optimal rule. We did not account for this scenario, and when the rule value was equivalent the program defaulted to our else statement for memoization. This led to some very skewed districts.

In order to solve this problem we tried a number of methods. We tried to set a maximum size for groups, adding a condition that favored adding an additional district, and to check district lean within a certain prespecified range. We then attempted a greedy approach (the file labeled greedy.cc”) if both configurations of districts gave the same outcome. We created methods within District called wastedA and wastedB, that calculated the wasted B votes and wasted A votes. The statement would use the amount of wasted votes as a sort of tiebreaker in the event that both configurations gave

the same numerical rule value. The idea is that we want to keep the combination that wastes the most majority votes. This approach was actually quite successful at creating more equivalent districts, but did not work consistently enough. Sometimes it would leave the middle two districts completely barren of voters. The interesting result of this algorithm is that the efficiency gap often went down to a level we had not seen before, to less than 0.1. What we assume is the explanation is that every time an equivalent rule value came up from the two options, the algorithm chose the path that wasted the most A votes roughly the same amount of times as the path that wasted B votes. The groups then are reformulated, folding unbalanced districts into each other. These two paths essentially offset each other, leading to very few wasted votes by the end of the program. We may have unintentionally found a way to minimize the efficiency gap.

## 5 Results

population majority rule	number of times minority rule achieved	number of times minority rule not achieved	average efficiency gap	average wasted votes A	average wasted votes B	maximum minority rule	minimum minority rule
100	10	0	.1078	99.8	8	-3	-1
500	8	2	.5064	498.6	7.8	-3	+1
900	1	9	.906.2	897.7	8.5	-1	+1

The table above shows general trends. We ran each majority rule ten times and calculated averages for each parameter. An interesting development was that the efficiency gap was approximately

$$(\text{majority rules}) \times 10^{-3}$$

As majority rule increased, there were fewer incidences of success at achieving minority rule. This makes sense, as an increasing majority makes it harder to achieve minority rule. However, even with 900 majority rule out of 1000 population, there was one instance of successful minority rule. With given limitations to district size/approximately even district size, this would be more difficult, since the votes that have accumulated to make minority rule happen would need to be separated according to size limitations. Essentially, having size limitations lowers the available "cut-offs" between districts, assuming contiguity, therefore making it more difficult for dividing to achieve certain minority rule with huge overpowering majorities such as 900 out of 1000.

## 6 Results

Dean, J. W. (1892). History of the Gerrymander. Priv. print. The RSLC Redistricting Majority Project REDMAP. (n.d.). Retrieved December 4, 2019, from <http://www.redistrictingmajorityproject.com/>