

**Empirical Validation of Automated Redistricting Algorithms on the Virginia  
House of Delegates District Map**

Madeleine Goertz

International Community School

AP Research

Randall S. Huberman

May 20th, 2021

## Literature Review

### Detection of Gerrymandering

- Explain existing research that focuses on whether or not existing maps are gerrymandered
- This is good, but given the move to nonpartisan commissions/collaborations within the state legislatures, it's necessary to have fair methods to generate the districts in the first place.
- This is why we need.....

### Automated Redistricting Algorithms

- Explain purpose of these
- Give some of the first examples
- My research uses three of these, go in depth explaining them.

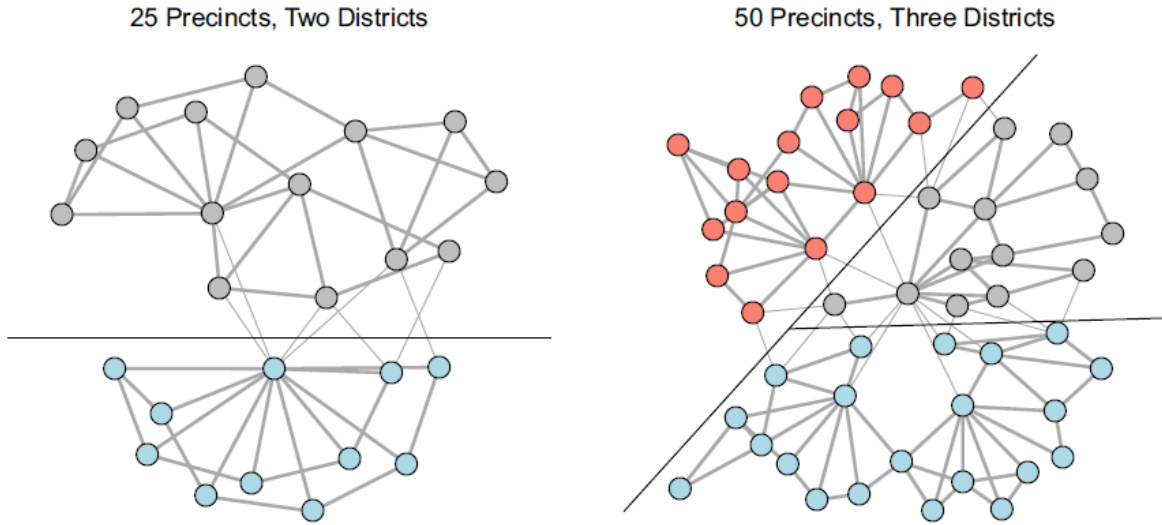
### *Markov Chain Monte Carlo*

The first automated redistricting algorithm that I'm going to discuss in detail is an implementation of a statistical method known as "Markov Chain Monte Carlo," henceforth MCMC (Fifield et al., 2020).<sup>1</sup> The following overview is meant to provide a high-level understand of how the basic algorithm works. Since the purpose of my research is to compare the *outputs*, that is the redistricting plans, generated by these algorithms, *rather than the algorithms themselves*, a rigorous understanding of the algorithms is not a prerequisite.<sup>2</sup>

---

<sup>1</sup> I will refer to the *redistricting algorithm* that uses Markov Chain Monte Carlo as "MCMC," rather than the *statistical method* Markov Monte Carlo.

<sup>2</sup> Please see the section "The Proposed Methodology" in the paper for an in-depth, mathematically-rigorous explanation. (Fifield et al., 2020, p. 2)



**Figure 1**

*Representation of redistricting as graph cutting. Every node is a precinct, and nodes that share an edge are known to be adjacent precincts. MCMC "cuts away" edges between nodes until islands of districts are formed. (Fifield et al., 2020, p. 3)*

MCMC conceptualizes the problem of redistricting precincts as a graph-cutting problem. For the uninitiated, a graph is a network of different interconnected points, where the points are called "nodes" and the lines connecting them are called "edges" (Fifield et al., 2020). MCMC represents every precinct as a node, and it draws an edge between nodes where the corresponding precincts are adjacent. Since the goal of redistricting is to assign every precinct a district, MCMC imagines that edges between nodes are "cut" until "islands" (known as "sub graphs") are formed, which each is disconnected from the rest. The disconnected "sub graphs" then become the districts. Figure 1 provides a nice visualization of this representation with a sample set of 50 precincts. (Fifield et al., 2020)

Now that the representation used by MCMC has been established, we can discuss the specifics of the algorithm. Figure 2 visualizes the steps in this algorithm. The basic MCMC algorithm begins with a valid redistricting plan, such as the one currently in use, and randomly decides to "turn on" some edges in the graph. Then, the nodes (precincts)

that are connected by these "turned on" edges and are located on the boundary of a district are identified. Then, these highlighted graph components are "nominated" for a swap across the district boundary based on some probability, provided that this swap would not break the district into two. Lastly, this new proposed redistricting plan is either accepted or rejected based on some "acceptance probability." This process is repeated as many times as desired. (Fifield et al., 2020).

MCMC in its current form allows for further constraint of this algorithm, particularly with regard to the relative population size and compactness of each district. (Fifield et al., 2020, p. 6) The explanation of this algorithm is beyond the scope of this paper.

### *Sequential Monte Carlo*

### *Random Seed Growth*

### **Empirical Validation**

- Usually small-scale validation within the papers that propose these Algorithms
- Needed to see how well the algorithms scale
- For a commission, it's helpful to be able to compare the results of several different leading algorithms.

### **Evaluation of Redistricting Plans**

Katz et al. (2020) brings mathematical rigor to the various proposed metrics for measuring partisan symmetry.

### *Partisan Symmetry*

A legislative is said to have partisan symmetry if both parties can receive  $m$  proportion of the overall votes and therefore have  $n$  proportion of the seats in the legislative body. An example would be that if Republicans win 60% of the votes but control

65% of the seats, then in a symmetrical system, Democrats should also be able to control 65% of the seats by winning 60% of the votes. Katz et al. (2020)

Partisan symmetry is usually observed by plotting a "seats-votes curve." This plot has  $V$ , the proportion of the overall votes won by the party, on the x-axis, and  $S(V)$ , the proportion of the seats won by the party, on the y-axis. Figure 3 (Katz et al., 2020, p. 175) illustrates several hypothetical seats-votes curves.

Naturally, it's very rare to observe the necessary electoral outcomes under the same electoral system in order to determine partisan symmetry. (ie., it's very rare for two parties to tie one year, have one win 51% of the total votes the next year, and then win 49% of the votes the following year.)

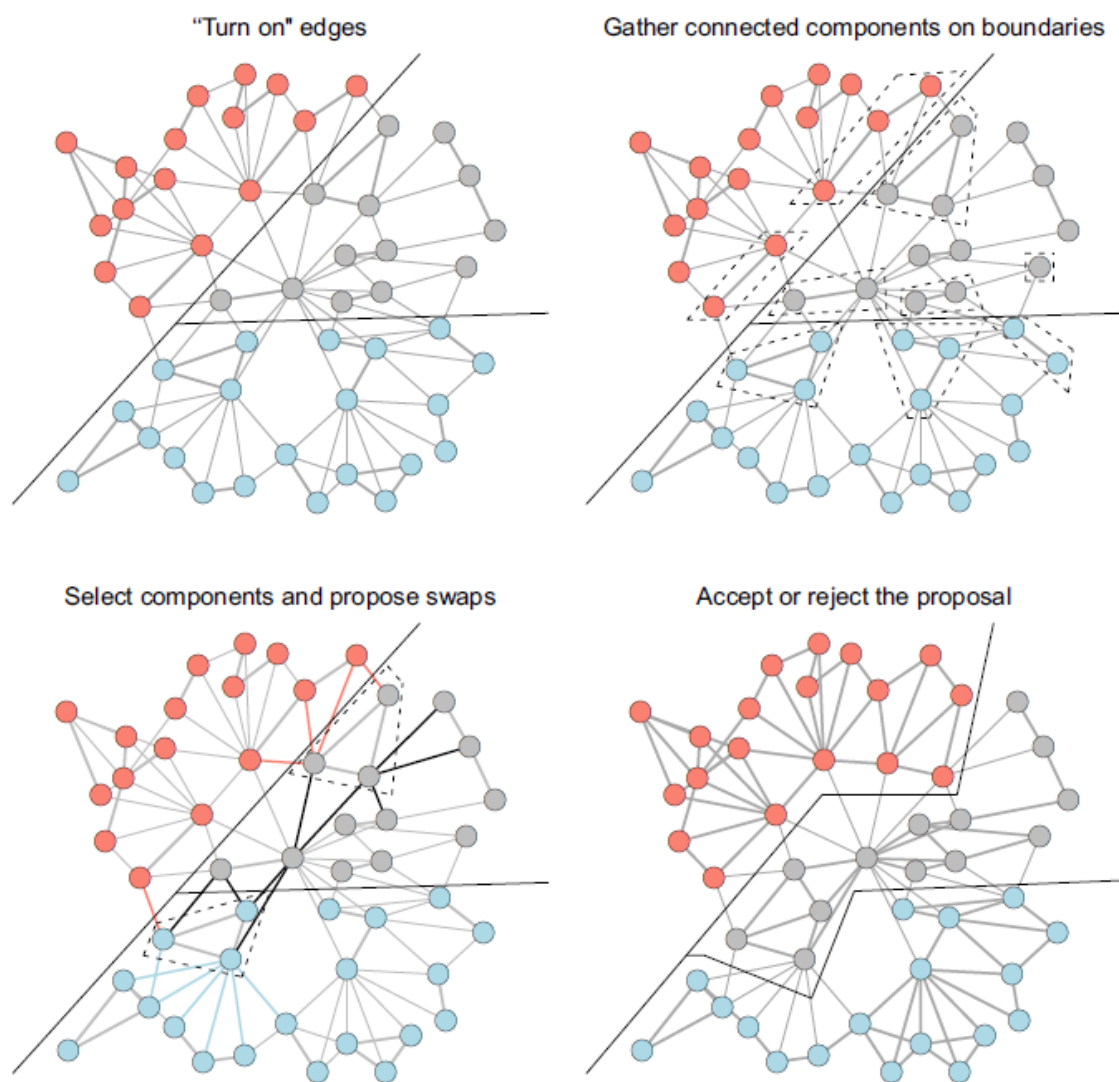
In practice, one can estimate a seats-votes curve using the principle of uniform partisan swing.

### ***Uniform Partisan Swing***

Uniform partisan swing is the principle that when the overall vote share  $V$  changes by some amount  $dV$  between different elections under the same electoral system, the vote share at the district level also usually changes by the same  $dV$ . Katz et al. (2020) Empirically verified this to be true in 646 different elections.

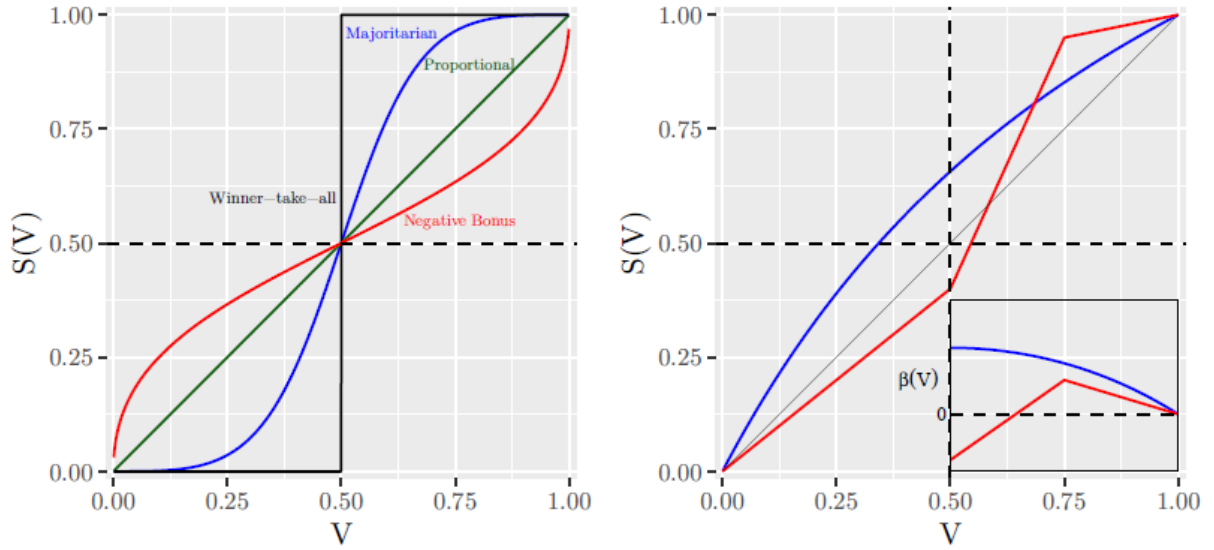
Thus, given a list of vote proportions per district  $v_1, v_2, v_3, \dots$  from one election, one can adjust each vote proportion by an arbitrarily small  $dV$  until the seats share  $S(V)$  is covered from 0 to 1. (Katz et al., 2020).

An example of such a curve is shown in Figure 4



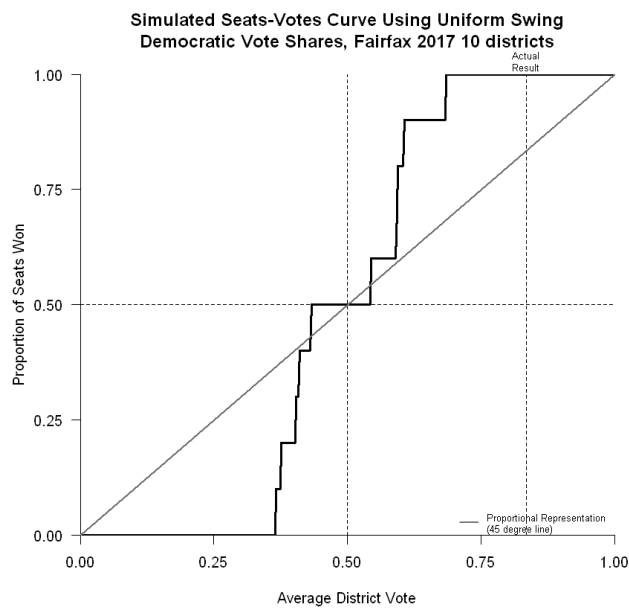
**Figure 2**

*Representation of MCMC algorithm. (Fifield et al., 2020, p. 4)*



**Figure 3**

*Types of Seats-Votes Curves. Left panel: Symmetric (fair) curves with differing levels of electoral responsiveness. Right panel: Asymmetric (biased) curves, including one consistently biased toward the Democrats (blue) and one with biases favoring different parties depending on  $V$  (red); the inset graph is for  $(V)$  for  $V \in [0.5; 1]$  with the vertical axis scaled to be the same as the main plot, and lines color coded to the seats-votes curves. (Katz et al., 2020, p. 175)*



**Figure 4**

*Sample seats-votes curve generated using uniform partisan swing. Ignore title. (Katz et al., 2020, p. 175)*



## References

- Fifield, B., Higgins, M., Imai, K., & Tarr, A. (2020). Automated Redistricting Simulation Using Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics*, *0*(0), 1–14. <https://doi.org/10.1080/10618600.2020.1739532>
- Katz, J. N., King, G., & Rosenblatt, E. (2020). Theoretical Foundations and Empirical Evaluations of Partisan Fairness in District-Based Democracies. *American Political Science Review*, *114*(1), 164–178. <https://doi.org/10.1017/S000305541900056X>