Pomona College

Senior Thesis in Mathematics

Mathematics of Redistricting: Identifying Gerrymandering Through Outlier Analysis

Author:
Olivia Leu

Advisor:
Dr. Jo Hardin

Submitted to Pomona College in Partial Fulfillment
of the Degree of Bachelor of Arts

April 14, 2023

Abstract

Gerrymandering is the act of drawing new political district boundaries in a way that gives electoral advantages to certain political parties, groups, or individuals. It is difficult to prove instances of gerrymandering, but this paper explores the process of ensemble generation to produce a large sample of district maps which is representative of the universe of plausible district maps which legislators should be choosing from. Next, researchers can perform outlier analysis to determine whether a given map is an outlier compared to the ensemble based on a number of different characteristics. If the map is an outlier, it may be a product of gerrymandering. The paper then turns to the `ReCom` algorithm, which is the most sophisticated established method for generating this ensemble of plausible district maps.

Contents

Chapter 1

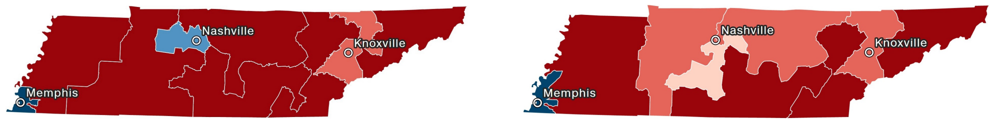
Introduction

Redistricting is the process of drawing new boundaries organizing geographical units into congressional and state legislative districts. Legislators would ideally treat the process of redistricting as a geometrical optimization problem: how can we best create new district boundaries which satisfy a number of constraints, such as equal population, contiguity, compactness, and more?

However, state legislatures are sometimes accused of *gerrymandering*: the process of redistricting in a way that gives electoral advantages to certain political parties, groups, or individuals. The state of Tennessee has been accused of gerrymandering in their creation of new congressional districts following the 2020 U.S. Census, as one of Tennessee's two previous majority Democratic districts has been split and redistributed into majority Republican districts, leaving the state with only one district which is projected to vote for Democratic candidates. Tennessee's new congressional district map motivated the writing of this thesis as an exploration of how to computationally identify instances of gerrymandering.

It is difficult to prove in a court of law that legislators are guilty of gerrymandering, since such an argument entails proving that legislators intentionally drew a district map with partisan bias. Until recently, arguing that legislators purposefully created biased maps entailed an attempt to get inside legislators' heads, identifying the intrinsic motivation (partisan, racially motivated, or otherwise) that led them to choose a particular map.

However, a number of mathematical techniques have been developed in recent years to examine the levels of bias in redistricting. One approach is to treat redistricting as a graph partition problem and use spanning trees for recombination in order to generate an ensemble of plausible district maps.



(a) Post-2010 Census Map (b) Post-2020 Census Map

Figure 1.1: Maps of Tennessee's nine U.S. congressional districts created following the 2010 and 2020 censuses. Maps are colored on a scale of blue to red based on the electoral results of the 2020 presidential election, with dark blue representing strong Democratic majority and dark red representing strong Republican majority (Boschma et al., 2022).

The `GerryChain` Python package uses the family of `ReCom` Markov chains to generate this ensemble and perform outlier analysis comparing a distribution of maps to the enacted map, examining a number of different potential biases. This mathematical technique provides an objective basis for comparison of district maps, enabling researchers to identify when, in the context of an ensemble of plausible district maps, it appears evident that an enacted district map is an extreme outlier and might therefore be a product of gerrymandering.

It is important to start by defining a few key terms which I will be referencing throughout my thesis.

Definition 1 (Voting Tabulation District (VTD)). **Voting tabulation district** (VTD) is a general term which refers to the geographical units of land which determine where citizens vote in local and state elections. Other familiar terms such as *precinct, ward,* and *census block* are all considered to be VTDs (Mumpower, 2021; U.S. Bureau of the Census, 1994).

VTDs serve as basic geographical units, and together they form *districts*.

Definition 2 (District). A **district** is a cluster of VTDs which is assigned a representative in local, state, or federal government. Individuals may vote in the election of a representative if and only if they are a resident of that representative's district (Mumpower, 2021; U.S. Bureau of the Census, 1994).

Having defined VTDs and districts, we can now formally define redistricting.

Definition 3 (Redistricting). **Redistricting** is the act of assigning all VTDs in a state to k districts, making sure that the resulting districts are equal in population (or only deviate up to a certain established threshold). The process of redistricting occurs once every ten years, following the U.S. Census.

Gerrymandering is thus, in short, intentionally biased redistricting. This thesis will examine the methods established by mathematicians to identify bias in redistricting.

Chapter 2 will establish the mathematical background which is the foundation of the analysis techniques in this paper. Chapter 3 will explain methods of ensemble generation, as ensembles serve as a basis of comparison for the potentially biased maps in question. Chapter 4 describes `ReCom`, a sophisticated technique for producing plans which will make up ensembles. Finally, chapter 5 enumerates final reflections and potential future directions for computational redistricting analysis.

Chapter 2

Mathematical Background

It is important to start by defining some of the main mathematical concepts behind algorithms for redistricting analysis.

Definition 4 (Graph). A **graph** $G = (V, E)$ consists of a set of *vertices*, V, and a set E of unordered pairs of elements in V. The elements in E, called *edges*, connect pairs of vertices. The set V must be non-empty, while E can be empty (Shahriari, 2021).

Visually, we can represent a graph as a collection of dots, representing our vertices, with lines connecting various pairs of dots, representing our edges.

Definition 5 (Subgraph). A **subgraph** $H = (V', E')$ of a graph $G = (V, E)$ is a graph such that $V' \subseteq V$ and $E' \subseteq E$ (Shahriari, 2021).

Definition 6 (Planar graph). A **planar graph** is a graph which can be drawn in a two-dimensional plane without any edges crossing (see figure 2.1) (Shahriari, 2021; Weisstein, 2023b).

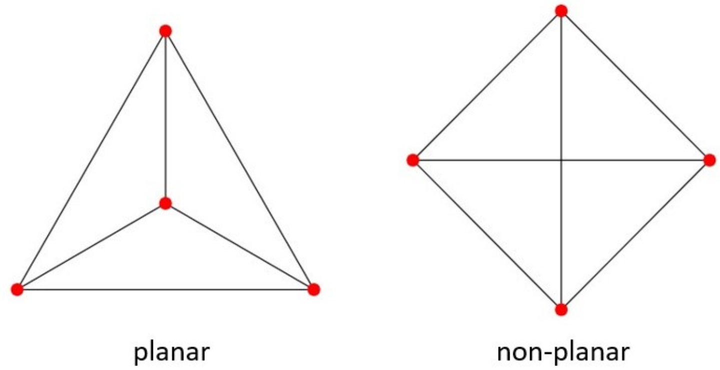


Figure 2.1: Planar graph (left) and non-planar graph (right) (Weisstein, 2023b).

Definition 7 (Dual graph)*.* Let G be a planar graph. We can construct the **dual graph** of G, $G*$, by following these two steps (see figure 2.2 for a visual representation of these steps):

1. Place a vertex in each region (area enclosed by edges) of G, including the exterior region.

2. If two regions of G share a boundary edge E, place an edge $E*$ between the vertices corresponding to these bordering regions, ensuring that $E*$ only crosses E.

The resulting graph, $G*$, will always be planar (Weisstein, 2023a).

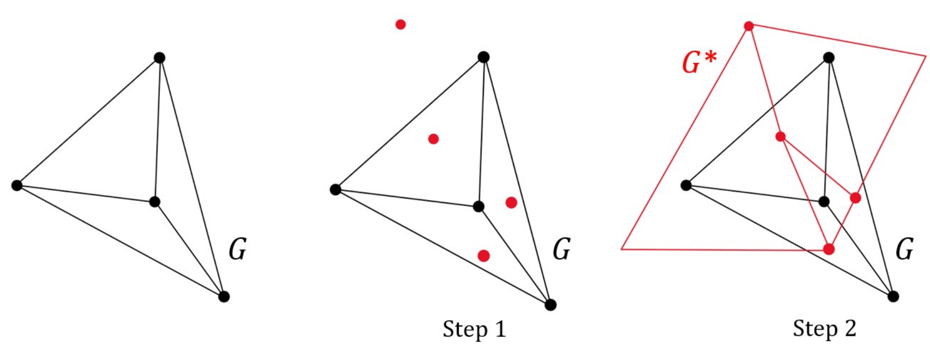


Figure 2.2: Steps to construct a dual graph, $G*$, from a planar graph, G.

Figure 2.3 demonstrates the establishment of a dual graph for the state of Iowa, assigning a vertex of the graph to each VTD, and connecting the vertices of VTDs which border one another with an edge. This establishment of a dual graph will serve as a first step in our ensemble generation efforts. Note that the dual graph of a state does not include a vertex in the region exterior to the state, and in this way differs from the traditionally defined dual graph.

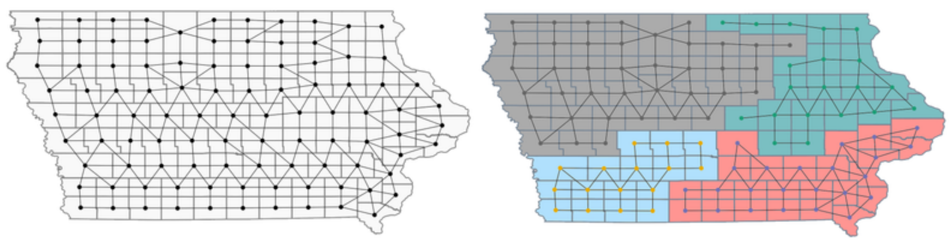


Figure 2.3: Dual graph of Iowa's VTDs (in this case, counties) on the left, with a district map assignment on the right (Deford et al., 2021).

Definition 8 (Walk). Let $G = (V, E)$ be a graph with some subset of the set of vertices V defined as $x_0, x_1, x_2, ..., x_m$. A *walk* of length m is a sequence of m edges in G of the form

$$\{x_0, x_1\}, \{x_1, x_2\}, ..., \{x_{m-1}, x_m\}.$$

We say that this walk *joins* the vertices x_0 and x_m (Shahriari, 2021).

Definition 9 (Cycle). Let $G = (V, E)$ be a graph. Assume this graph has a walk of length m of the form

$$\{x_0, x_1\}, \{x_1, x_2\}, ..., \{x_{m-1}, x_m\},$$

where $x_0 = x_m$ and all other vertices in the walk are distinct. We call this walk a **cycle** (Shahriari, 2021).

Definition 10 (Connected graph). A graph $G = (V, E)$ is *connected* if there is a walk joining every distinct pair of vertices in V (Shahriari, 2021).

Definition 11 (Tree). Let $G = (V, E)$ be a graph. If G is connected and contains no cycles, then we call G a **tree** (Shahriari, 2021).

Definition 12 (Spanning tree). Let $G = (V, E)$ be a graph. Let $T = (V, E')$ be a tree where $E' \subseteq E$. We call T a **spanning tree** for G (Shahriari, 2021). In other words, a spanning tree of a graph G is a subgraph of G containing all of the graph's n vertices, which are connected by exactly $n-1$ edges (see figure 2.4). Note that, by construction, a spanning tree cannot contain any loops.

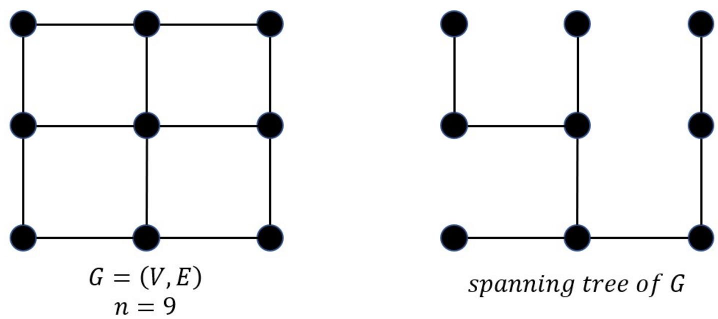


Figure 2.4: Connected graph G with nine vertices and a possible spanning tree of G. Note that the figure displays just one of the potential spanning trees of G.

Definition 13 (Markov chain). A **markov chain** is a sequence of random variables $X_1, X_2, ...$ such that X_k depends only on X_{k+1}. Thus, we can write $\mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n, X_{n-1} = x_{n-1}, ..., X_1 = x_1) = \mathbb{P}(X_{n+1} = x_{n+1} | X_n = x_n)$.

The fact that we can establish a dual graph on a state, with one vertex for every geographical unit and edges that connect the vertices corresponding to geographical units that border one another, informs our approach to ensemble generation. A partition of this dual graph into k districts represents a districting plan. In order to generate an ensemble, we start with a partitioned dual graph of a state and implement a Markov chain which uses the partitioned dual graph as its initial state and modifies the current state at each step of the Markov chain. A particular Markov chain algorithm, known as `ReCom`, uses spanning trees to modify district plans by recombination. Using these iterative Markov chains, we can generate a wide array of plans which together form an ensemble.

Chapter 3

Ensemble Generation

In order to analytically identify instances of gerrymandering, researchers aim to generate an ensemble of district maps which is representative of the set of plausible district maps.

Definition 14 (Plausible district maps)*.* The set of *plausible district maps* is the set of district maps which, in theory, legislators should be choosing from. This set is defined by a number of constraints which exclude maps that would be unreasonable for legislators to choose. The set of plausible district maps should thus represent the sample space from which legislators select a redistricting plan.

The ensemble of maps must satisfy a number of constraints, which are determined by government regulations (and may vary state to state) as well as informal standard practices of redistricting. Although the ensembles will not be representative of the set of all *possible* districting plans (see section 3.2), researchers must take steps to make sure that the ensemble is representative of the set of districting plans which a legislator might *plausibly* propose in a state.

3.1 Constraints

Definition 15 (Constraint)*.* A **constraint** is a restriction implemented in an algorithm which establishes criteria that alter the output of that algorithm.

Constraints are vital in ensemble generation and redistricting practices in general. We may implement any of the following constraints into a redistricting algorithm: population balance, contiguity, compactness, splitting rules, compliance with the Voting Rights Act, and neutrality.

The implementation of constraints in ensemble generation can vary. There are two main types of constraints: *preferences* and *requirements* (also commonly referred to as *rejection filters*).

Definition 16 (Preference). A **preference** is a constraint which assigns higher weights to districting plans which satisfy the constraint, while assigning lower weights to plans which do not satisfy the constraint (Duchin & Spencer, 2021). The resulting ensemble will contain a higher proportion of plans which meet the restrictions established in the preference than those that do not.

Definition 17 (Requirement (rejection filter)). A **requirement**, also known as a **rejection filter**, is a constraint which forces the algorithm to exclude districting plans that do not satisfy the constraint from the ensemble (Duchin & Spencer, 2021). The resulting ensemble will contain only plans which meet the restrictions established in the requirement.

Duchin and Spencer (2021) advise researchers to take care when implementing rejection filters into their ensemble-generation algorithms. They warn that the inclusion of too many requirements, or even a singular overly strict requirement, may cause the ensemble-generation algorithm to "get stuck," preventing the algorithm from taking additional steps and impacting the ability of the ensemble as a whole to be representative of the set of possible districting plans. They cite the Chen-Stephanopoulos protocol as an example of a method which implements a high number of strict requirements, resulting in small sample sizes which are at high risk of misrepresenting the set of possible districting plans (see Chen and Stephanopoulos (2021) for an in-depth description of this protocol).

Researchers can establish rejection filters which exclude districting plans that fail to meet certain constraint thresholds from the ensemble, or they can preferentially weight plans which better satisfy the constraints. How researchers implement constraints varies, and most will likely include both preferences and requirements in their redistricting analysis.

3.1.1 Population balance

The most universally implemented constraint in ensemble generation is *population balance*.

Definition 18 (Population balance). Let P_s be the population of state s, which is subdivided into k districts. Let p_i be the population of the ith district in state s, where $i = 1, 2, \ldots, k$. **Population balance** occurs when, for $i = 1, 2, \ldots, k$:

$$p_i \approx \frac{P_s}{k}$$

In words, **population balance** occurs in a redistricting plan when the population of each district in a state is roughly equal.

It is unlikely that all districts in a state will have the exact same number of inhabitants, and thus most researchers provide a small margin of error for population balance. The exact range of this margin of error can vary based on state-specific redistricting practices or preferences of researchers. Although there is no consistent practice in place for setting a maximum population deviation value, legislators should strive for population balance in their redistricting efforts, and thus it is important to incorporate this constraint into an ensemble generation algorithm.

In their analysis of potential districting plans for the Virginia House of Delegates in 2018, the Metric Geometry and Gerrymandering Group (2018) established a requirement that their computer-generated maps could deviate no more than 1% from the ideal population value. In contrast, in their analysis of nested districts in Alaska, Caldera, DeFord, Duchin, Gutekunst, and Nix (2020) allowed district populations to deviate up to a maximum of 5% from the ideal population value, in accordance with the maximum population deviation established by federal law. Duchin and Spencer (2021) conducted redistricting analysis on a total of twenty states, allowing district populations to deviate up to 2% of the ideal population size. Tennessee's guide to redistricting references a "ten-percent standard," allowing the population to deviate up to 10% of the ideal population size (Mumpower, 2021). However, this standard is not enforced by law in Tennessee. The U.S. Census Bureau simply writes regarding population balance, "Each congressional district is to be as equal in population to all other congressional districts in a state as practicable" (U.S. Bureau of the Census, 2023). Throughout all implementations of population balance constraints researched for this thesis (both in state legislation and in computational redistricting practices), 10% is an upper bound for population deviation between districts. However, legislators practically tend to choose plans with much smaller population deviation between districts, so researchers in computational redistricting often choose an upper limit for this deviation which is smaller than 10% (Deford et al., 2021). A redistricting plan which is lenient regarding population balance may raise questions as to why legislators are allowing for population deviation to that extent, when it is likely not necessary. Thus, setting strict population balance constraints as a *requirement* in ensemble generation allows researchers to identify plans which do not respect this degree of population balance and question whether legislators were motivated by a particular bias in selecting those plans in particular.

3.1.2 Contiguity

Definition 19 (Contiguity). A district is *contiguous* if all of its geographic components are connected to each other. A districting plan which respects *contiguity* is composed entirely of contiguous districts.

With a few exceptions, contiguity implies that, regardless of starting point within a district, one can reach any other point in the district without crossing into another district. Exceptions may occur around bodies of water and in other particular cases. In other words, every geographical unit in a district must neighbor at least one other geographical unit within the same district in order for contiguity to be respected in a districting plan. Constructing dual graphs (see Definition 7) for each state whose edges connect only those geographic components which share a border helps ensure contiguity (Deford et al., 2021).

There are three main types of contiguity: rook contiguity, bishop's contiguity, and queen's contiguity. We are largely concerned with rook contiguity, so we proceed by defining it.

Definition 20 (Rook contiguity). In *rook contiguity*, two regions are considered contiguous if they share a border of non-zero length. This is in contrast to *bishop's contiguity,* under which two regions are considered contiguous if they meet at a single point, and *queen's contiguity,* under which two regions are considered contiguous if they share a border of non-zero length *or* meet at a single point (Anselin, 2020).

In the field of computational redistricting, we want to enforce *rook contiguity*, meaning that in order for a district to be considered contiguous, its constitutive geographical units must each share a border of non-zero length with at least one other geographical unit in the district (Metric Geometry

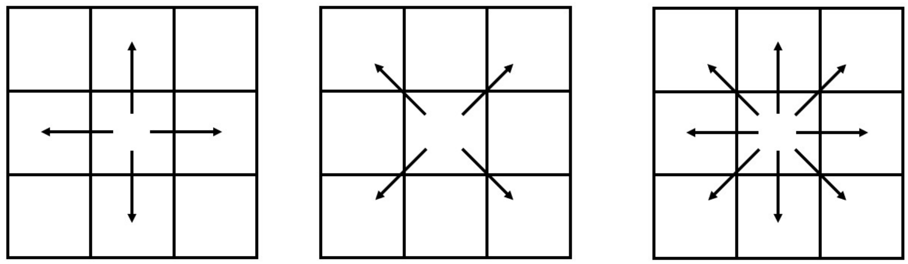


Figure 3.1: Rook contiguity, bishop's contiguity, and queen's contiguity on a $3x3$ grid.

and Gerrymandering Group, 2018). In other words, a district is not contiguous if one of its constitutive geographical units is connected to the rest of the district by only a single point. Like in the case of population balance, we implement rook contiguity as a requirement (as opposed to a preference) in ensemble generation.

3.1.3 Compactness

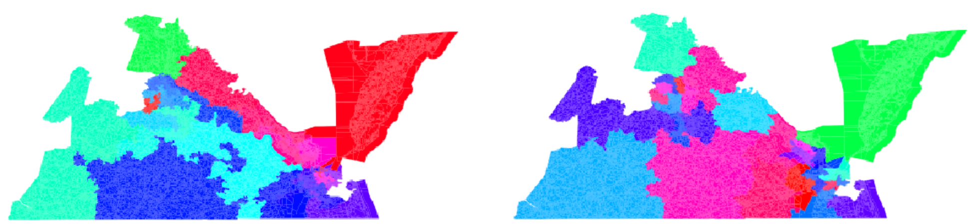
Definition 21 (Compactness)*. Compactness* concerns a preference for "regular" or condensed district shapes.

While population balance and contiguity are relatively simple to measure, compactness is more difficult to define. It is less obvious which districts satisfy compactness constraints and which do not. Deford et al. (2021) write, "in practice, compactness is almost everywhere ruled by the proverbial eyeball test." An eyeball test is difficult for mathematicians to accept, and even more difficult to model, so although legislators simply depend on the eyeball test, researchers in the field of the mathematics of redistricting have developed alternative methods of quantifying compactness. Deford et al. (2021) mention the Polsby-Popper score, which correlates compactness with equal perimeters across districts, as well as the Reock score, which assesses compactness by comparing a district to a circumscribed circle. However, they note that these scores are unreliable and highly variable. Instead, Deford et al. (2021) suggest the use of *cut edges* to measure compactness.

Definition 22 (Cut edge). A *cut edge* is an edge in the dual graph (see definition 7) of a state whose vertices are members of different districts.

The number of cut edges in the dual graph of a given districting plan provides a measure for the relative compactness of that plan.

The total number of cut edges in a districting plan corresponds roughly to the total length of the perimeter shared between districts, and thus represents important information about compactness. Researchers can strive to generate an ensemble of compact districts by minimizing the number of cut edges in the ensemble plans. Deford et al. (2021) also note that the number of cut edges in the dual graph of a districting plan is inversely correlated with the total number of spanning trees of all individual districts in the plan. A compact districting plan will consist of mostly plump, rounded districts, resulting in a small number of cut edges in the state dual graph and a large number of potential spanning trees for each district in the districting plan. A districting plan which is not compact may have skinny, winding districts, resulting in a large number of cut edges in the state dual graph and a small number of potential spanning trees for each district in the districting plan (see Figure 3.2).



(a) Plan with 11,996 cut edges　　(b) Plan with 5702 cut edges

Figure 3.2: These sample districting plans of Virginia illustrate the relationship between compactness and cut edges; the map on the left is non-compact with a high number of cut edges, while the map on the right is much more compact and has fewer cut edges (Metric Geometry and Gerrymandering Group, 2018).

3.1.4 Splitting rules

States often designate a preference for districting plans which preserve areas that are larger than the basic geographical units which compose districts.

These larger areas may be counties, municipalities, or "communities of interest" (Deford et al., 2021). For example, the Tennessee Guide to Redistricting recommends that legislators should "consider other variables when drawing new plans," such as "city or incorporated boundaries" (Mumpower, 2021). However, this constraint is not implemented as a requirement, but rather as a preference.

Deford et al. (2021) note multiple potential methods for implementing this constraint as a preference in computational redistricting. One technique is to score plans based on the total number of larger areas of interest which are split. Researchers would count the total number of areas of interest which are divided by district boundaries and establish a preference for plans which split the lowest number of areas of interest. Duchin and Tenner (2021) discuss "county-conscious sampling," discussing a number of different methods to account for county preservation interests in computational redistricting. One such method requires that the districts merged in stage 2 of `ReCom` (see section 4.2.2) must be two districts that split a county, in the hopes that the two new districts generated from the merged super-district will no longer split that county. Another method counts the total number of counties split in the currently enacted plan, and requires that plans generated by the modification proposal algorithm may not split more counties than the enacted plan. An additional method is specific to the `ReCom` algorithm, giving slightly higher random weights to edges of the spanning tree of the super-district which connect geographical units which are both within the same county, thus increasing the likelihood that the cut edge in the spanning tree will be on a county border (see section 4.2.3 for more information on the specifics of stage three of the `ReCom` algorithm). This method qualifies as a preference. Further, Duchin and Tenner (2021) also propose a rejection filter which sets a maximum number of county *pieces* which the boundaries in a districting plan can create. Instead of simply counting the number of counties which are divided, this method accounts for situations in which a single county may be split into three or more districts. Thus, Duchin and Tenner (2021) propose both preferences and rejection filters as methods for implementing constraints regarding county preservation; they also note the validity of refraining from establishing any county preservation constraint at all.

Ultimately, redistricting splitting rules vary greatly from state to state, with some states requiring specific redistricting practices by law with regards to the splitting of areas of interests, and other states refraining from mentioning county preservation (or the preservation of cities, municipalities, or other communities of interest) as a priority when it comes to redistricting. The constraints adopted in computational redistricting practices thus vary based on the established splitting rules in the state the researcher is analyzing, as well as the priorities or preferences of the researcher. A researcher might choose to generate multiple ensembles with a variety of splitting constraints and compare the characteristics of the resulting ensembles.

3.1.5 Voting Rights Act Compliance

The Voting Rights Act of 1965 is defined as "an act to enforce the fifteenth amendment to the Constitution," which extended voting rights to Black men in the United States in 1870 (*Voting Rights Act*, 1965). Section 2 of the Voting Rights Act has notable implications for redistricting practices, as it prohibits any voting legislation which would result in the "denial or abridgement of the right of any citizen of the United States to vote on account of race or color" (Democracy Docket, 2021b). Specifically, Section 2 of the Voting Rights Act (VRA) outlaws what is known as *"minority vote dilution"* (Hebert, Smith, Vandenberg, & DeSanctis, 2010).

Definition 23 (Minority vote dilution). *Minority vote dilution* occurs when districting plans reduce or eliminate the potential political influence (also known as voting power) of racial or ethnic minority groups.

Section 2 of the Voting Rights Act indicates that the denial or abridgement of the right to vote occurs if the members of any racial or ethnic minority group "have less opportunity than other members of the electorate to participate in the political processes and to elect representatives of their choice" (Hebert et al., 2010). In order to prevent the denial or abridgement of the right to vote for members of racial or ethnic minority groups through minority vote dilution, Section 2 sometimes calls for the creation of *"majority-minority districts"* (Hebert et al., 2010).

Definition 24 (Majority-minority district). A *majority-minority district* is a district in which a racial or ethnic minority group "constitutes an effective voting majority" (Hebert et al., 2010).

In the 1986 Supreme Court decision in *Thornburg v. Gingles*, the Supreme Court outlined a framework indicating when a majority-minority district is required under Section 2 of the Voting Rights Act. The following four statements must all be true in order for the Voting Rights Act to mandate the creation of a majority-minority district:

1. The minority group must be “sufficiently large and geographically compact to constitute a majority” in a single-member district (Hebert et al., 2010).

2. The minority group must be “politically cohesive” (Hebert et al., 2010).

3. The white majority must vote “sufficiently as a bloc to enable it... usually to defeat the minority’s preferred candidate” (Hebert et al., 2010).

4. “Under the totality of circumstances,” the minority group must have “less opportunity than whites to participate in the political process and to elect representatives of its choice” (Hebert et al., 2010). In other words, there must exist a “lack of proportionality” in which the minority group lacks a level of representation which is proportional to their share of the population (Hebert et al., 2010).

When all of the above requirements are true, Section 2 of the Voting Rights Act mandates the creation of a majority-minority district. Hebert et al. (2010) expands in-depth on the criteria for determining the conditions under which each of these requirements is true, noting that different court cases have provided standards for the truth of these statements that are sometimes in conflict with one another. However, the existence of legislation which can mandate the creation of a majority-minority district is meaningful in determining proper redistricting practices.

Deford et al. (2021) note the difficulty of incorporating the requirements of the Section 2 of the Voting Rights Act into ensemble generation as a quantitative constraint, given that the requirements for determining whether a majority-minority district is mandated involve the analysis of “local histories of discrimination and patterns of racially polarized voting.” Becker, Duchin, Gold, and Hirsch (2021) suggest the use of *effectiveness scores* in order to preferentially weight district plans containing districts in which members of racial and ethnic minority groups have “realistic opportunities to nominate and elect their preferred candidates.” By incorporating these effectiveness scores as a constraint in ensemble generation, we can create what Becker et al. (2021) refer to as “VRA-conscious ensembles” which are representative of the “universe of VRA-compliant plans.”

It is important to note that, often, there is a trade-off between VRA-compliance and compactness. In order to build a majority-minority district, legislators sometimes have to create districts which appear wildly non-compact to the eye. Non-compact districts are considered to be warning signs for gerrymandering, but there are cases in which districts are intentionally non-compact in order to give a racial or ethnic minority group the opportunity to gain a majority and elect a representative of their choosing within the district. For example, Illinois' 4th Congressional District has been given the nickname of the "earmuffs" district due to its unusual shape, and many have accused state legislators of racial or partisan gerrymandering because the district is so blatantly non-compact (see figure 3.3). However, Illinois' 4th Congressional District is actually a majority-minority district created in compliance with the Voting Rights Act which provides the Latinx population of Chicago the opportunity to elect a representative of their choosing (Scales, 2020). Thus, although non-compact districts are usually good indicators for racial or partisan gerrymandering, the Voting Rights Act may mandate the creation of unusually non-compact districts in order to provide electoral opportunity to racial or ethnic minority groups.

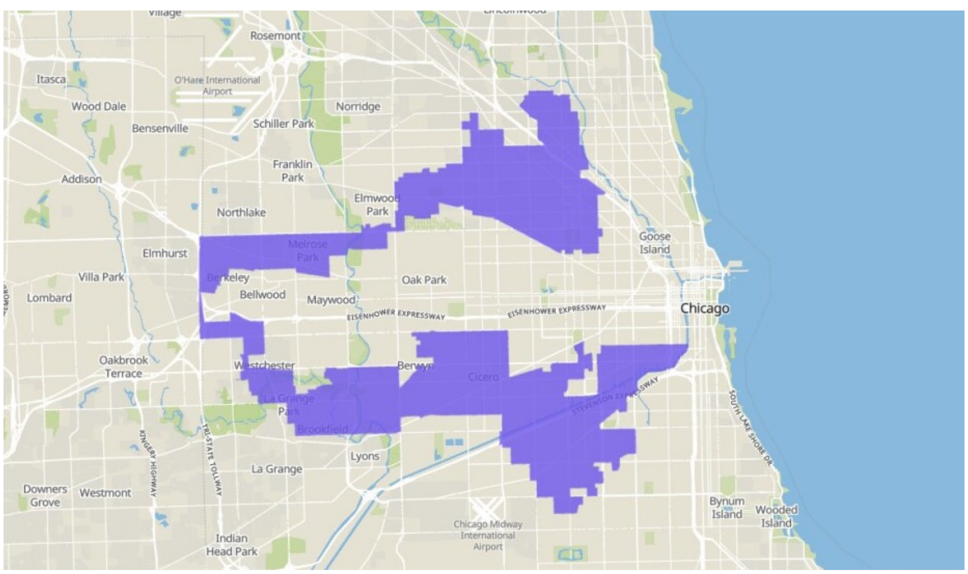


Figure 3.3: Illinois' 4th Congressional District. A VRA-compliant majority-minority district created to give Chicago's Latinx population an effective voting majority (Scales, 2020)

At the time of writing (April 2023), the Supreme Court is considering the case *Merrill v. Milligan*, in which voters and non-profits in Alabama challenged the new congressional district map of Alabama drawn following the 2020 U.S. Census based on the claim that it violated Section 2 of the Voting Rights Act. The map includes only one majority-Black district, when the Black population proportionally makes up enough of Alabama's total population that the state should have two majority-Black districts to ensure compliance with the Voting Rights Act (Democracy Docket, 2021a). If the Supreme Court rules against those challenging the enacted map, they may invalidate Section 2 of the Voting Rights Act, fundamentally changing the way legislators are held accountable for providing racial and ethnic minorities with the opportunity for proportional representation. The Supreme Court previously invalidated another section of the Voting Rights Act which provided even stricter regulations on state redistricting practices, and thus lawmakers are keenly aware of the possibility that the decision in *Milligan* will strip the Voting Rights Act of yet another set of requirements which protect the voting rights of racial and ethnic minority groups.

Given that Section 2 of the Voting Rights Act is still valid at the time of writing, researchers must consider ways to establish constraints which promote the creation of VRA-compliant ensembles in computational redistricting. The most sophisticated way of establishing such a constraint is the use of effectiveness scores, referenced above.

3.1.6 Neutrality

Deford et al. (2021) note that some states establish rules to promote neutral redistricting practices by stating that information such as partisan data or incumbency status should not influence redistricting practices. Notably, the Tennessee Guide to Redistricting contradicts Deford et al. (2021) by explicitly permitting legislators to consider the "protection of incumbents" when drawing new district maps (Mumpower, 2021).

The method for ensuring that computer-generated ensembles are composed of neutral plans is to avoid importing partisan data or incumbent addresses associated with the geographical units which will compose the districts. While a computer-generated ensemble can easily ignore partisan data, it is more difficult for a legislator to do so. Equally, it is difficult to prove that legislators purposefully took partisan data into account when drawing new district maps, or whether the partisan leanings of districts are simply a coincidence. Partisan gerrymandering (purposefully drawing districts in a way that benefits a particular political party) is a major concern in the

United States today– while we can ensure that a computer does not generate plans with intentional partisan bias by withholding from the computer the partisan data associated with the geographical units in a state, it is much more difficult to make sure that legislators lack any partisan bias in their redistricting practices.

3.2 Sampling

As previously stated, it is key to note that when generating ensembles of districting plans, researchers do not aim to generate ensembles which are representative of all possible districting plans (i.e., plans which respect population balance and contiguity). The two universally implemented constraints in redistricting are population balance between districts and contiguity, which requires all districts in a district plan to be composed only of geographical components which neighbor other geographical units within the district. However, many districting plans which satisfy population balance and contiguity and are thus considered to be *possible* districting plans will not belong to the sample space of *plausible* districting plans, largely due to their lack of compactness (see Figure 3.2).

Deford et al. (2021) note two main reasons why *uniform sampling*, randomly sampling from the entire population of possible district maps, is both impractical and unhelpful in the context of redistricting. First, they write, "the uniform distribution would be regarded as prohibitively non-compact in the application domain" (Deford et al., 2021). In fact, they remark that the majority of plans in any random sample from the entire population of possible districting plans would involve bizarrely shaped districts which will have substantially lower compactness scores, on average, than enacted or plausible plans. Randomly sampled plans tend to be so non-compact that they would have no likelihood of real-world implementation, and thus they are not useful to include in our analysis.

While some may suggest uniform sampling from all possible plans with a compactness threshold (e.g., setting an upper limit of the number of cut edges for inclusion in the ensemble), researchers will still face a dilemma similar to when they attempted to uniformly sample without a compactness threshold. Deford et al. (2021) demonstrate in their paper that the majority of plans in an ensemble restricted by a compactness threshold will have compactness scores which are very close to the chosen threshold (i.e., most plans in the ensemble will be as non-compact as possible). The second reason which Deford et al. (2021) provide to explain the necessity of nonuniform sampling is the fact that "there are in any case obstructions to uniform sampling at the practical scale of redistricting problems." They note that even in a $7x7$ grid, there are 158,753,814 possible plans that divide the grid into 7 districts of 7 units each (Deford et al., 2021). When researchers attempt to shift to a problem on the scale of an entire state, the number of possible plans increases exponentially. For example, the state of Tennessee has 240,116 census blocks from which they construct nine districts; it would be computationally prohibitive to generate the population of all possible plans and randomly sample from it (U.S. Bureau of the Census, 2013).

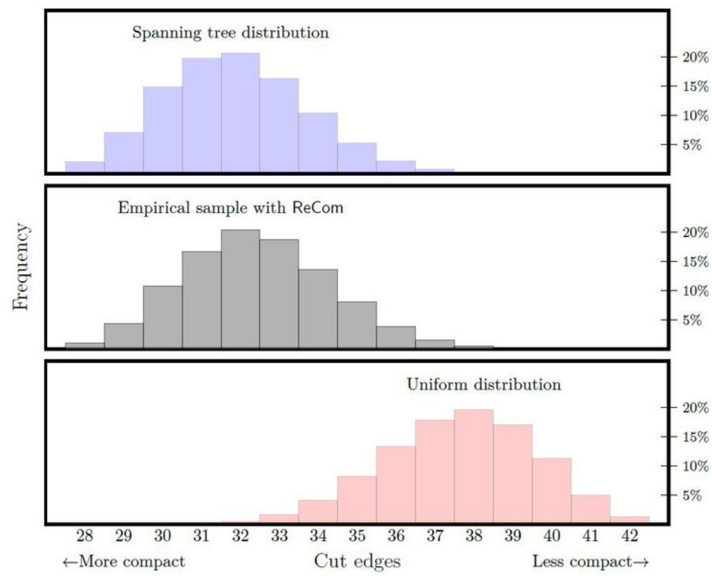


Figure 3.4: Ensemble compactness statistics based on the number of cut edges in district plans partitioning a $7x7$ grid into 7 districts of 7 units each. The red distribution represents the uniform distribution of all 158,753,814 possible plans. The blue distribution represents the more compact spanning tree distribution, which is our target distribution in this case. The gray distribution represents the distribution of 100,000 possible plans generated using the `ReCom` algorithm. It is evident that `ReCom` targets the more compact spanning distribution, demonstrating the ability of `ReCom` to generate compact ensembles (Deford et al., 2021).

Given the difficulty in generating an entire population of possible districting plans to randomly sample from, as well as the lack of compactness in uniformly drawn samples which would restrict practical applications and the potential of meaningful comparison of a uniformly sampled ensemble and an enacted redistricting plan, researchers must identify a different approach to ensemble generation. Deford et al. (2021) note that `ReCom` naturally runs in such a way that the probability of selecting a particular partition (e.g., a particular districting plan) is roughly proportional to the number of spanning trees of the districts that make up that partition. We know that plumper districts have more possible spanning trees, so `ReCom` makes plans which are more compact more likely. This ensures that the resulting ensemble will better represent the pool of plausible plans a legislator should be sampling from than a uniformly sampled ensemble.

More formally, we can define $sp(G)$ as the total number of possible spanning trees of a graph G. Note that for any graph G, $sp(G)$ is greater when the graph is plumper, or more compact. Given a districting plan P which is composed of k districts $P_1, P_2, ..., P_k$, we let $sp(P) = \prod_{i=1}^{k} sp(P_i)$ DeFord and Duchin (2022). A partition P with a high $sp(P)$ value indicates that it is likely composed of compact districts, signaling that the plan P is a compact districting plan.

Deford et al. (2021) show that when using the `ReCom` algorithm to generate an ensemble, the probability of generating a districting plan P is roughly proportional to $sp(P)$, meaning `ReCom` naturally favors districting plans which are more compact. This is desirable, since it means that a `ReCom`-generated ensemble is able to better reflect the pool of plausible districting plans which legislators might be choosing from. Figure 3.4 demonstrates the tendency of `ReCom` to generate ensembles which are more compact than uniformly-sampled ensembles and which approximate the spanning tree distribution (the distribution of the value $sp(P)$), which is our target distribution. Given that the universe of *plausible* district plans is, on average, more compact than the universe of *possible* district plans, the fact that ensembles generated with `ReCom` are, on average, more compact than uniformly sampled ensembles inspires confidence that `ReCom` provides a practical technique for generating a sample which is representative of the sample space of plausible district plans.

3.3 Phases of Ensemble Generation

DeFord (2019) lays out five main phases to ensemble generation:

Phase 1. Establish constraints to define the set of possible partitions that we can sample from.

Phase 2. Select an initial districting plan, known as the seed plan.

Phase 3. Propose a modification to the plan.

Phase 4. Ensure that the modification satisfies the constraints established in phase 1.

Phase 5. Apply the Metropolis-Hastings Algorithm to preferentially accept more compact plans into the ensemble.

Phase 6. Repeat phases 2-5.

3.3.1 Phase 1: Establish constraints

The first phase of ensemble generation consists of establishing constraints to define the set of *plausible district maps*.

Given definition 14 of *plausible* district maps, the goal of generating an ensemble of district maps which represent the sample space of plausible district maps is intuitive, given that our aim in ensemble generation is to generate a set of maps which can serve as a base of comparison for a given redistricting plan in outlier analysis. Establishing constraints to specify the range of characteristics necessary for inclusion in the ensemble is a key part of ensuring that the ensemble is representative of the set of plausible district maps.

Section 3.1 defines the constraints implemented in the field of computational redistricting, and this section will proceed by discussing the implementation of these constraints.

The first (and most universally implemented) constraint that we implement is population balance. Population balance is unique in that, unlike most other constraints listed in section 3.1, states often legally establish upper bounds on the degree to which population can vary between districts in a state. Legislators mostly select plans with districts whose populations deviate much less than the upper limit allows them to, so researchers will often implement a population constraint which is stricter than the law or established precedent in a given state. Researchers universally implement population balance as a *requirement* (see section 3.1), so that the ensemble does not contain any plans whose districts deviate in population more than the established upper limit. It is also a generally accepted practice to implement rook contiguity as a requirement (see definition 20).

Beyond population balance and contiguity, the implementation of constraints varies, and Deford et al. (2021) encourage individuals to experiment with different constraints and see how the resulting ensemble changes.

There are a number of different ways of implementing compactness constraints when generating an ensemble. First, the researchers note that particularly in the case of the `ReCom` algorithm, the algorithm naturally produces plans which are compact by construction (see section 3.2), without any tuning Deford et al. (2021). Thus, with certain modification techniques which tend to create compact plans, it may not be necessary to implement an additional compactness constraint. Other researchers may implement a maximum number of *cut edges* (see definition 22) as either a requirement or a preference, given the established inverse relationship between total number of cut edges in a district plan and compactness. Other researchers have implemented various compactness scores to give preference to district plans which are more compact. The Polsby-Popper score assigns district plans a score based on the lengths of the perimeters of the districts which compose the plan, and the Reock score compares districts in a district plan to a circle in order to assess their compactness (Deford et al., 2021). Deford et al. (2021) advocate for the use of cut edges over these scores as a measure of compactness, but it is clear that there is not a standardized way of handling compactness in the field of computational redistricting.

Researchers enumerate many potential methods of implementing constraints regarding splitting rules, noting that the implementation of splitting rule requirements or preferences depends on the regulations of the specific state being analyzed, as well as the goals and interests of the individuals conducting the research. Becker et al. (2021) note the potential use of effectiveness scores to ensure compliance with the Voting Rights Act, establishing a preference for plans which provide more opportunities for racial and ethnic minorities to elect representatives of their choice. Finally, researchers can ensure neutrality in ensemble generation by refraining from importing any partisan voting data associated with the geographical units in a state.

Thus, while researchers in the field of computational redistricting have accepted the standardized practices of establishing an upper bound for population deviation between districts and rook contiguity as requirements, there is room for researchers to be creative with other constraints. Researchers may not choose to implement any constraints beyond population balance and rook contiguity, or they may implement additional constraints to emphasize the importance of certain characteristics in district plans, such as compactness or Voting Rights Act compliance. However, Deford et al. (2021) warn against implementing too many constraints or rejection filters which are overly strict, as they have the potential to drastically slow down the process of ensemble generation if researchers reject a large percentage of the district plans generated in phase three of ensemble generation: propose a modification.

3.3.2 Phase 2: Select a seed plan

Once we have established constraints, we must select an initial state, or starting point, for our ensemble generation. We will feed this initial state into our modification algorithm in phase three (see section 3.3.3).

Definition 25 (Initial plan). An *initial plan*, also known as a *seed plan*, is the district plan that we use as the initial state in our modification algorithm.

Deford et al. (2021) note the importance of generating an ensemble which is based on multiple seed plans to ensure that the composition of the ensemble is not dependent on a single starting point. In this way, researchers can better ensure that ensembles are representative of the sample space of plausible district maps (see definition 14). Thus, we can use the enacted plan as a seed plan, but we need the ability to generate additional seed plans. These plans must be contiguous and sufficiently population-balanced, and researchers may choose to make them subject to additional constraints, such as compactness (Deford et al., 2021). Deford et al. (2021) describe two potential methods for seed plan generation: agglomerative methods and spanning tree methods. Deford et al. (2021) write that they utilize both of these techniques when generating ensembles.

Deford et al. (2021) lay out the following steps for the *spanning tree method* for generating initial plans of a state with k total districts and total population P_s:

1. Draw a spanning tree τ for the dual graph G of the entire state.

2. Select an edge to cut in τ that will result in one district d_i which has a population within an acceptable range of $\frac{P_s}{k}$, also known as the ideal district population size. The established population balance constraint will determine the acceptable range above or below the ideal district population size. The remainder of the graph G will form the spanning tree τ', which consists of all vertices of G which have not yet been assigned to a district.

3. Repeat step 2 until G has been partitioned into k total districts with populations within an acceptable range of the ideal district population size. This partition of G is a seed plan generated using the spanning tree method.

In addition to the spanning tree method, Deford et al. (2021) describe the *agglomerative method* for generating seed plans with k total districts. The agglomerative method consists of the following steps:

1. Randomly select k geographical units (i.e. vertices of the dual graph of the state) to be the seeds of the k districts in the plan.

2. Merge each of the seed districts with neighboring geographical units until every geographical unit has been assigned to one of the k total districts.

3. Check whether the resulting plan respects the established population balance constraint. If not, re-balance the plan by re-assigning geographical units to new districts until population balance is achieved. To prevent breaking contiguity, make sure to only re-assign a geographical unit into a new district that the geographical unit shares a border with.

The agglomerative method is also known as the “Petri dish” method (Duchin & Tenner, 2021). Deford et al. (2021) note that this seed plan generation method often reaches a “dead-end configuration,” meaning it is unable to complete a valid plan. If this occurs, the algorithm will start over and attempt again to generate a seed plan.

Although Deford et al. (2021) notes the necessity of multiple seed plans, they do not specify how many plans in total are required to ensure that the resulting ensemble is independent of starting point. In their analysis of the state of Virginia, the Metric Geometry and Gerrymandering Group (2018) uses 101 total initial plans: the enacted plan, as well as 100 mathematically generated seed plans. Autry, Carter, Herschlag, Hunter, and Mattingly (2020) define the Multi-Scale Merge-Split Markov Chain Monte Carlo algorithm for redistricting analysis, which they describe as an extension of the `ReCom` algorithm. Autry et al. (2020) analyze the state of North Carolina, generating an ensemble using ten total initial plans: the 2016 enacted district plan, the 2020 enacted district plan, and eight seed plans produced using the spanning tree method. It is clear that there is no standard practice for choosing the total number of seed plans implemented in ensemble generation, and this choice may depend on the goals of the researcher or the computational power available to them.

3.3.3 Phase 3: Propose a modification

The next phase of ensemble generation is to propose a modification on the selected initial plan. Chapter 4 provide an in-depth explanation of two main techniques for proposing a modification, the `Flip` algorithm and the `ReCom` algorithm.

Techniques for proposing a modification take an initial plan as input, modify that initial plan (subject to the established constraints from phase one of ensemble generation), and produce a new plan which will be added to the ensemble if it meets the established constraints and is accepted by the Metropolis-Hastings algorithm. The modification algorithm will proceed by modifying the new plan, and proposing the resulting plan to be contributed to the ensemble. The repetition of this process many times will create an ensemble. The amount of repetitions necessary to generate an ensemble which is representative of the sample space of plausible district maps depends on the selected modification proposal algorithm, which will be expanded on in chapter 4.

3.3.4 Phase 4: Ensure modification satisfies constraints

Although establishing the constraints in phase one of ensemble generation should ensure that all plans proposed by the modification algorithm satisfy those constraints, DeFord (2019) suggests incorporating an additional safety check to exclude any maps from our ensemble that violate the established constraints and therefore are not elements of the sample space of plausible district maps. Given that the goal of ensemble generation is to construct an ensemble of district plans which is representative of the sample space of plausible district maps, it is key to implement this phase of ensemble generation as a sort of security checkpoint for potential member plans of the ensemble, making sure that all plans which compose the ensemble meet the criteria established in the constraints in phase one of ensemble generation.

3.3.5 Phase 5: Metropolis-Hastings Algorithm

The goal of the Metropolis-Hastings Algorithm is to sample from a desired distribution about which we don't know the exact form, known as the *target distribution* (DeFord & Duchin, 2022). In the context of computational redistricting, the target distribution is the distribution of all plausible district maps which satisfy the established constraints from step 1 of ensemble generation. Researchers in computational redistricting tend to choose the distribution of compactness scores of all plausible district maps as the target distribution, given the ability of ensemble generation algorithms to produce plans which are comparably compact to those typically selected by legislators is an area of concern for researchers (Deford et al., 2021). Given the previous discussion regarding the necessity of nonuniform sampling due to the fact that it would be computationally prohibitive to generate the population of all plausible plans and randomly sample from it (see section 3.2), it is also computationally prohibitive to generate the compactness scores of all plausible plans for the same reason. Since we are interested in the distribution of compactness scores of plausible plans but cannot feasibly generate the distribution in its entirety and sample from it, the distribution of compactness scores of plausible plans is a perfect candidate to be a target distribution for the Metropolis-Hastings Algorithm.

The Metropolis-Hastings Algorithm requires an initial target distribution which then initiates a new Markov chain which aims to reflect the target distribution without knowing the characteristics of the target distribution in its entirety. DeFord and Duchin (2022) note that the Metropolis-Hastings algorithm is particularly useful when there exists a score which ranks some states in the state space (not to be confused with the states of the United States!) as preferable compared to other states, and we want to generate an ensemble which targets the desired distribution by preferentially accepting plans with "better" scores. DeFord and Duchin (2022) write, "We essentially use the score to start with one Markov chain, then design a cleverly weighted coin and use a coin flip to accept or reject each proposed move" (354). By designing a process which accepts or rejects proposed states based on a score function, we are able to generate an ensemble of district maps whose characteristics resemble that of the target distribution. In the case of redistricting, we implement a score function which assesses the compactness of districting plans, assigning "better" scores to plans which are more compact. By designating a preference for plans with "better" compactness scores, we are able to use the Metropolis-Hastings algorithm to preferentially accept more compact plans in exactly the way we hope a legislator would prioritize compactness.

Start with a score function s on the state space Ω, such that $s : \Omega \to \mathbb{R}$. To ensure that states are weighted according to their scores and that we are sampling from the target distribution, we assign probabilities to each state in the state space based on their scores. Thus, for any state $y \in \Omega$, let $\mathbb{P}(y) = \frac{s(y)}{\sum_{x\in\Omega} s(x)}$. Note that in most situations it is impossible to calculate $\sum_{x\in\Omega} s(x)$ since we are unable to represent the whole state space Ω given its large size.

In cases where we cannot construct the state space in its entirety, we note that although we cannot calculate $\mathbb{P}(y)$, we can instead calculate a ratio of two probabilities since the denominators of both probabilities cancel:

$$\frac{\mathbb{P}(z)}{\mathbb{P}(y)} = \frac{s(z)}{s(y)}.$$

Being able to calculate this ratio is sufficient for the preferential weighting in the Metropolis-Hastings Algorithm, allowing us to sample from our target distribution without constructing the target distribution in full.

To perform the Metropolis-Hastings procedure, we start with a Markov chain which will propose steps from one state (the initial state) to another (the proposed state). We then calculate the ratio of the score functions (which is also the ratio of the probabilities of the two states) for the two states (initial and proposed) which is used in the decision of whether or not to accept the proposed state with a certain probability DeFord and Duchin (2022) note, "It is this possibility of remaining in place that transforms the stationary distribution to our desired values" (355). In other words, the potential of rejecting a proposed state based on the ratio of its score and the initial state's score allows us to better approximate the target distribution.

We can represent the Metropolis-Hastings Algorithm as a series of steps (DeFord & Duchin, 2022):

1. Starting with an initial state y (i.e., a district plan), use the Markov chain to generate a proposed state z from the initial state y. In other words, propose z as the next step of the Markov chain.

2. Let $\alpha = \frac{s(z)}{s(y)} \frac{q(z|y)}{q(y|z)}$, where $q(z|y)$ is the probability of transitioning from the proposed state z to the initial state y, and $q(y|z)$ is the probability of transitioning from the initial state y to the proposed state z.

   If $\alpha \geq 1$, accept z as the next state (i.e., district map in the ensemble).

   If $0 < \alpha < 1$, accept z with probability α and reject z with probability $1 - \alpha$.

3. If z was accepted, it becomes the next state. If z was rejected, the next state is y.

4. Repeat this process each time the Markov chain generates a new proposed state (i.e., after each step of the `ReCom` algorithm).

Note, in the Metropolis-Hastings Algorithm, the probability α of accepting the proposed state z is calculated using the probability of transitioning from the current state y to the proposed state z, as well as the probability of transitioning from state z to state y. The calculation of these transition probabilities depends on the method of proposing a modification selected in Phase 3 of ensemble generation (see section 3.3). Chapter 4 focuses on the `ReCom` algorithm, a sophisticated method of proposing modifications for ensemble generation. Section 4.2.5 will outline the process of calculating the necessary transition probabilities in order to find the value α when running the Metropolis-Hastings algorithm on a state proposed by the `ReCom` algorithm in particular. For now, it suffices to note that because the current state y and the proposed sate z are linked through the generation of spanning trees, the probability of transitioning from y to z (and back) depends only on the number of total possible spanning trees of each plan and the number of total possible cut edges which could have resulted in two population-balanced districts.

3.3.6 Phase 6: Repeat phases 2-5

As stated in section 3.3.2, it is necessary to generate an ensemble which is based on multiple seed plans in order to ensure that the resulting ensemble is (1) *independent* from starting point and (2) sufficiently *representative* of the sample space of plausible district maps. Thus, researchers must complete phases three through five of ensemble generation for each selected seed plan. As noted in section 3.3.2, there is no generally accepted number of seed plans required to generate an independent and representative ensemble, so this decision is largely based on individual researchers' goals and available computational power.

In addition to the required repetition of phases three through five for each selected seed plan, the ensemble generation process contains more internal repetition which is worth noting. As stated in section 3.3.3, various modification proposal algorithms require different amounts of repetition in order to generate an ensemble which is independent from starting point and sufficiently representative of the sample space of plausible district maps. The total amount of repetition needed may range from 10,000 steps (the generally accepted lower bound for repetition of the `ReCom` modification proposal algorithm) to upwards of 1,000,000 steps when using the `Flip` modification proposal algorithm (Deford et al., 2021). Phases four and five of ensemble generation repeat with each step of the modification proposal algorithm. This means that each time a modification proposal algorithm generates a new potential district plan to add to the ensemble, phase four must ensure that the proposed plan does not modify any of the constraints established in phase one of ensemble generation, and the Metropolis-Hastings algorithm (phase five of ensemble generation) must probabilistically determine whether to accept the proposed plan into the ensemble.

A simplified example will help illuminate the layers of repetition involved in ensemble generation. Assume we have determined that 2 seed plans are necessary to generate an independent and representative ensemble. Assume also that for the selected modification proposal algorithm, 10 total plans are required to ensure that the resulting ensemble is sufficiently representative of the sample space of plausible district maps. Thus, for each of the 2 seed plans, we must generate 10 plans using the chosen modification proposal algorithm. Our ensemble will thus have 20 plans in total. For each of these 20 plans, we must implement phase four to ensure that they meet the constraints established in phase one of ensemble generation, and we must use the Metropolis-Hastings algorithm (phase five of ensemble generation) to preferentially accept more compact plans into the ensemble. If either phase four or five rejects one of the twenty proposed plans from the ensemble, we must return to the modification proposal algorithm and generate new proposed plans (which we will then test with phases four and five of ensemble generation) until we have a total of 20 plans accepted into the ensemble. Thus, the layers of repetition are very important for the ensemble generation process.

The amount of repetition necessary depends heavily on the selected modification proposal algorithm (it was noted earlier that the potential repetition of the modification proposal algorithm ranges from 10,000 steps to over 1,000,000 steps, depending on the chosen modification proposal algorithm). The following chapters explores two of the main modification proposal algorithms that exist in the field of computational redistricting, the `Flip` algorithm and the `ReCom` algorithm, and explains why the nature of these different algorithms calls for such drastically different amounts of repetition.

Chapter 4

The ReCom Algorithm

Once the initial districting plan has been selected, there are a number of proposed methods of modifying this initial plan in order to create plausible district maps which, together, can compose a diverse ensemble which is representative of the set of all plausible district maps. As our goal is to perform outlier analysis utilizing an ensemble to determine whether a given redistricting plan shows evidence of bias, it is necessary to implement a modification technique (i.e., phase three of ensemble generation) which is sophisticated enough to accurately represent the sample space of plausible district maps, while avoiding being overly computationally expensive to allow us to generate a large enough set of district maps. Becker and Solomon (2020) address the issue of attempting to find a modification technique which successfully represents the sample space of plausible district maps without being overly complicated– they explain that, especially in large-scale redistricting problems, we want to select an ensemble-generation algorithm which maximizes the *quality* of the generated plausible district plans while also maximizing the *efficiency* of the algorithm by keeping the runtime required to generate a district map relatively low (Becker & Solomon, 2020). Here, the quality of an ensemble refers to the ability of that ensemble to accurately capture the universe of plausible district maps which, in theory, legislators should be choosing from. The following sections describe several attempts by researchers to find an algorithm which is both efficient and successful at generating high-quality ensembles.

4.1 Other modification proposal techniques

While there are numerous possible techniques for proposing a modification (see Becker and Solomon (2020) for more details), the two most well-known algorithms are `Flip` and `ReCom`. Although the `Flip` algorithm has been widely used in the field of redistricting mathematics, Deford et al. (2021) make a strong argument for the use of the newer `ReCom` algorithm instead. The `Flip` algorithm changes the district assignment of one census block at a time, while `ReCom` works on a much larger scale. The runtime for a single step of the `Flip` algorithm is substantially shorter than the runtime for one step of the `ReCom` algorithm; however, while `ReCom` requires at least 10,000 steps to produce a diverse ensemble of plausible district maps, `Flip` requires at least 1,000,000 steps (and there is some evidence which indicates that 1,000,000 steps may not even be reliably sufficient) (Deford et al., 2021). Thus, this thesis will largely focus on the details of the `ReCom` algorithm, as its ability to generate an ensemble of compact plans with relatively few steps indicates its superiority to the `Flip` algorithm in this case (Deford et al., 2021).

4.1.1 The `Flip` Algorithm

The `Flip` algorithm precedes the `ReCom` algorithm, and is a more well-known technique in the field of mathematical analysis of redistricting practices. This algorithm follows the following basic steps:

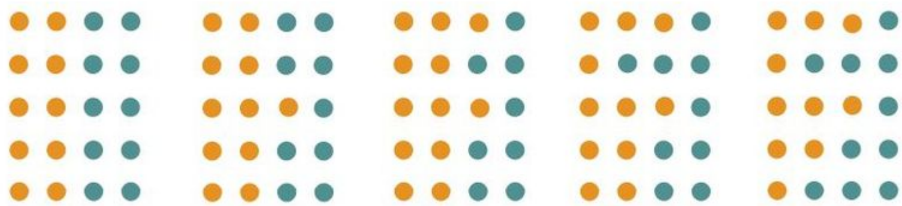
1. Randomly select one geographical unit in some district d_i which borders some district d_j where $i \neq j$ (we must select a geographical unit on the border of two districts in order to preserve contiguity).

2. Reassign, or "flip" the selected geographical from d_i to d_j, resulting in a new district map which differs from the previous map by only one node.

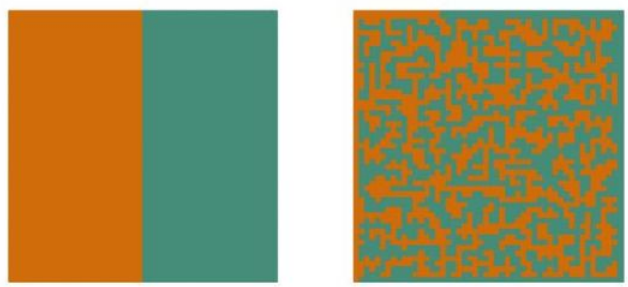
3. Add the new map to the ensemble.

4. Repeat.

While `Flip` ensures that all districts will be contiguous, it frequently results in plans which are notably non-compact. Figure 4.1 demonstrates a major problem with the `Flip` algorithm. It is obvious that a legislator



(A) Sequence of four flip steps



(B) Before and after 500,000 flip steps

Figure 4.1: (A) shows a simplified instance of the `Flip` algorithm on a $5x4$ grid. (B) demonstrates 500,000 steps of `Flip` on a $50x50$ grid which is split into two population-balanced districts (Deford et al., 2021).

would *never* choose the map which results from 500,000 `Flip` steps in figure 4.1(B). Recall that the goal of ensemble generation is to produce an ensemble which is representative of all plausible district maps. `Flip` produces maps that are not plausible, and this impacts the overall quality of our ensemble. It is thus difficult to conduct outlier analysis when an ensemble contains so many maps which are realistically irrelevant.

Deford et al. (2021) also note that while a singular `Flip` step requires very little computational power, it takes an extreme number of steps to generate an ensemble which is independent of the selected seed plan, and even more steps to create an ensemble which is representative of the sample space of plausible district maps. Deford et al. (2021) write, "after 1 million steps the structure of the initial state is still clearly visible, and we will present evidence that one billion steps is enough to improve matters significantly, but not to thee point of approximate convergence of the ensemble." Thus, the apparent computational ease of conducting a singular `Flip` step is counterbalanced by the total number of steps needed to generate a usable ensemble.

Deford et al. (2021) instead propose an algorithm called `ReCom`. A singular step of `ReCom` requires more computational power than a singular step of `Flip`, but substantially fewer steps of `ReCom` are required to generate a representative ensemble which is independent of its seed plan(s). Figure 4.2 compares `Flip` and `ReCom`, demonstrating the vast difference in required steps to generate district maps which are compact and independent from the seed plan.

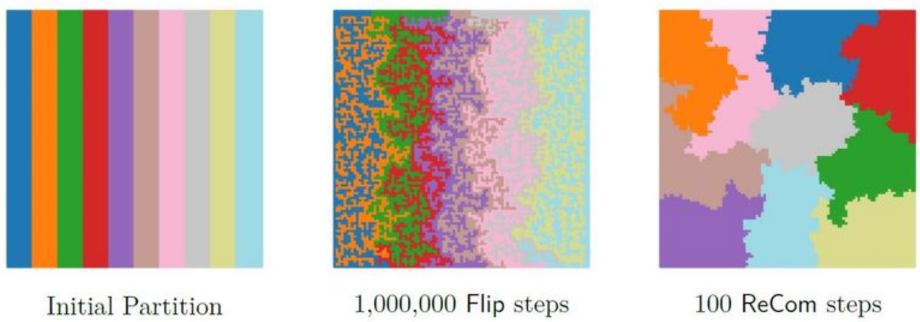


Figure 4.2: Deford et al. (2021) ran both the `Flip` and `ReCom` algorithms on an initial partition of a $100x100$ grid into 10 population-balanced "districts." The resulting `Flip` partition is notably non-compact and exhibits notable similarities to the initial partition. Only 100 `ReCom` steps generate a resulting partition consisting of compact districts which do not resemble the initial partition.

It seems that `ReCom` provides solutions to the issues of non-compactness and high number of necessary steps which plague the `Flip` algorithm. I will proceed by detailing the process of the `ReCom` algorithm.

4.2 The stages of the `ReCom` algorithm

Having established the rationale for implementing `ReCom` in redistricting analyses, it is important to examine the mathematics behind this algorithm.

Note that the `ReCom` algorithm is one possible way to "propose a modification to the plan," or phase 3 of ensemble generation (see section 3.3).

Equally, we could use the `Flip` algorithm as an alternative way to propose a modification to the seed plan. Thus, regardless of the modification technique selected for phase 3 of ensemble generation, phases 1 and 2 must precede the implementation of any modification.

We can represent the `ReCom` algorithm with five major stages (DeFord, 2019):

Stage 1. Select two bordering districts (see figure 4.3).

Stage 2. Combine the geographical units of the two districts to create a super-district (see figure 4.4).

Stage 3. Draw a spanning tree for the super-district (see figure 4.6).

Stage 4. Remove an edge from the spanning tree, leaving two districts with approximately equal populations (see figure 4.7).

Stage 5. Repeat stages 1-4.

It is important to note the reference to the preceding five actions as "stages," rather than "steps." Stages 1 through 4, as a whole, represent one "step" of the `ReCom` algorithm. Thus, stage 5, which asks us to repeat stages 1 through 4, prompts us to run another step of `ReCom`. Hence, each step of `ReCom` implements stages 1-4 and results in a modified districting plan. This modified districting plan gets added to the ensemble, as long as it passes phases 4 and 5 of ensemble generation. Thus, 10,000 "steps" of `ReCom` would involve 10,000 runs of stages 1 through 4 of the algorithm, resulting in an ensemble with 10,000 plausible district maps (with the exception of some maps which may not make it into the ensemble due to their elimination in phase 4 or 5 of ensemble generation).

I will proceed by examining each of these stages in detail.

4.2.1 Stage 1: District selection

The first stage of `ReCom` is to select two bordering districts which the algorithm will manipulate in future stages. Deford et al. (2021) write, "we select our pair of adjacent districts to be merged proportionally to the length of the boundary between them" (Deford et al., 2021). In other words, the selection process preferentially targets pairs of districts which share the longest boundaries. This promotes district compactness, as it is likely that once we merge and re-partition a pair of districts which initially share a long boundary, the boundary between the pair of newly generated districts will be shorter (Deford et al., 2021). The preferential method of district selection promotes a state's compactness priority in redistricting, thus increasing the likelihood that the maps generated through `ReCom` will be plausible.

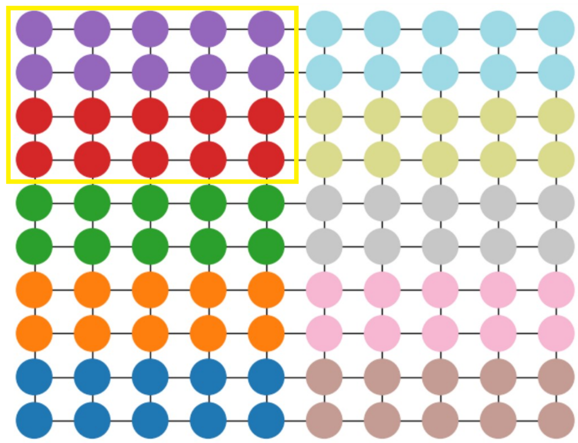


Figure 4.3: A simplified map of ten population-balanced districts, each with ten nodes. We will arbitrarily select the purple and red for demonstrative purposes (DeFord, 2019).

Although it is possible to select more than two districts at once to merge, restricting our selection to two districts at a time greatly reduces the runtime necessary for each step of `ReCom` (Deford et al., 2021). We do not lose sophistication by choosing two districts rather than k districts (the total number of districts in the state) or some ℓ districts (where $2 < \ell < k$), and thus we simplify the steps of `ReCom` by merging and re-partitioning two districts at each step.

4.2.2 Stage 2: Merging the selected districts

The next stage of the `ReCom` algorithm is to merge the two selected districts, creating one super-district. This super-district will contain all of the nodes from the two previously distinct districts, with edges still connecting the sub-units (precincts, VTDs, etc.) that geographically border each other. This stage lacks complexity, but the importance of the stage lies in the fact that the two previously separate districts now together make up one super-district, hence eliminating any ties of the sub-units to their previously separated districts. See Figure 4.4.

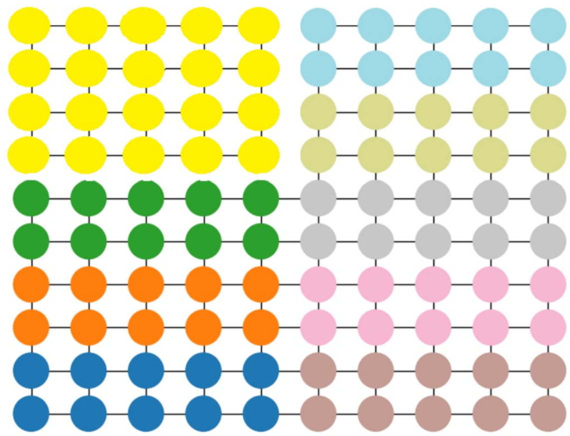


Figure 4.4: In this figure, the purple and red districts from figure 4.3 have been merged into one yellow super-district (DeFord, 2019).

4.2.3 Stage 3: Spanning trees

The next stage of the `ReCom` algorithm is to draw a spanning tree of the super-district.

In Stage 3 of the `ReCom` algorithm, we want to sample uniformly from all potential spanning trees of the super-district (Deford et al., 2021). Wilson's algorithm is remarkable in that it allows for near-uniform sampling from the set of all possible spanning trees for a graph in polynomial time, so we implement it in stage three of the `ReCom` algorithm (DeFord & Duchin, 2022; Wilson, 1996).

Definition 26 (Uniform Spanning Tree). A **uniform spanning tree** (UST) of a graph G is a spanning tree chosen uniformly at random from the set of all possible spanning trees of G (Schweinsberg, 2008).

Specifically, `ReCom` implements the loop-erased random walk (LERW) method of Wilson's algorithm in order to generate uniform spanning trees.

Definition 27 (Loop-erased random walk (LERW)). The **loop-erased random walk (LERW)** method implements a simple random walk on a graph, removing loops as they appear.

In order to create a spanning tree of a graph, Wilson's algorithm implements the LERW method to generate a uniform spanning tree through the following steps (Schweinsberg, 2008):

1. Randomly select two vertices of the graph, x_0 and x_1. Run a LERW from x_0 to x_1, creating the tree T_1 (see definition 11).
2. Once we have T_k, randomly select a vertex x_k. Run a LERW from x_k to T_k, connecting x_k to T_k and creating T_{k+1}
3. Repeat step 2 until all n vertices in the graph are connected in the tree by $n-1$ edges.

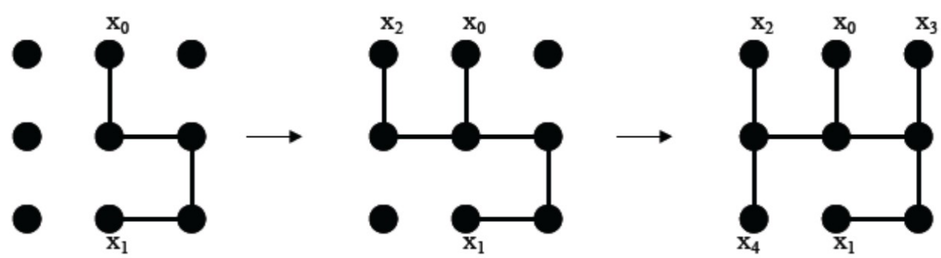


Figure 4.5: This figure shows a simplified version of Wilson's algorithm, generating a spanning tree for nine vertices (Schweinsberg, 2008).

Thus, Wilson's algorithm allows near-uniform sampling from all possible spanning trees of our super-district, and the LERW method ensures that we are able to draw a spanning tree that includes all n vertices of our super-district, connected by $n-1$ edges, and containing no loops.

4.2.4 Stage 4: Edge deletion

The next stage is to remove an edge of the spanning tree, resulting in two population-balanced districts. The meaning of "population-balanced" may change depending on state preferences and the amount of population deviation permitted by the established constraints.

If it is not possible to remove an edge such that we end up with two population-balanced districts, we must return to stage 3 and draw a new spanning tree. Although this situation does occur, Deford et al. (2021) note,

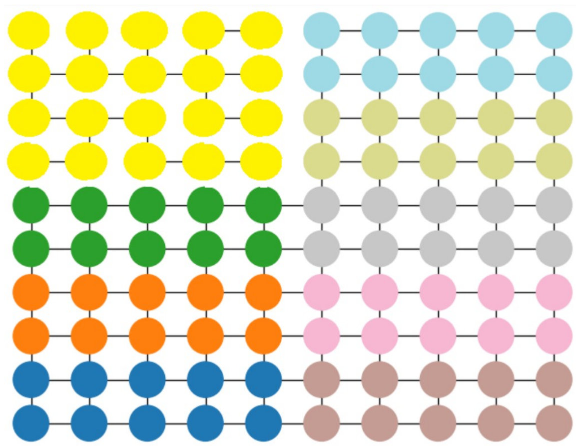


Figure 4.6: This figure shows a possible spanning tree for the super-district from figure 4.4 (DeFord, 2019).

"in practice, the rejection rate is low enough that this chain runs efficiently." Thus, we must account for the possibility of there being no edge whose removal will result in two population-balanced districts, but we assume that it will not occur so frequently that it will disrupt `ReCom`'s operation.

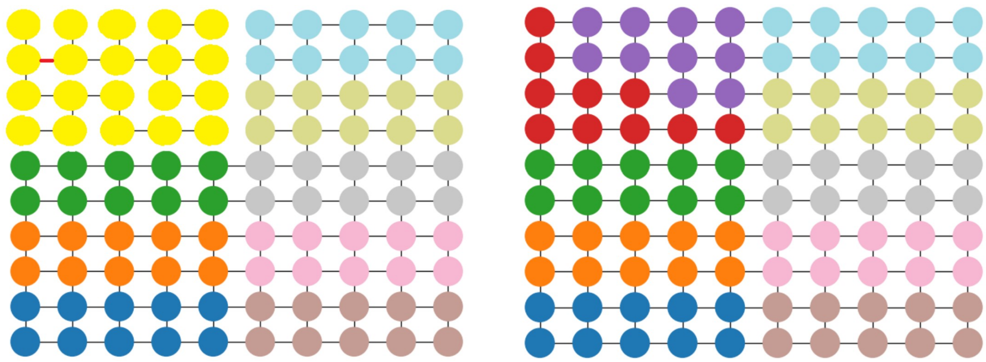
It may also be the case that there exists a set of multiple possible edges which we can select from, each of which, when deleted, generates two districts which satisfy the established population balance constraint. In this case, we sample uniformly from all removable edges.

Once we have removed an edge, we are left with two new population-balanced districts, and hence a new plausible district map which gets added to the ensemble.

4.2.5 Stage 5: Repetition

Generally, full-scale redistricting problems require at least 10,000 steps of the `ReCom` algorithm in order to generate a diverse ensemble which effectively represents the set of plausible redistricting plans (Deford et al., 2021). As was previously stated, although the steps of `ReCom` are somewhat computationally complex, substantially fewer total steps are required for `ReCom` as compared to other ensemble-generation techniques in order to generate a representative ensemble.

After each step of the `ReCom` algorithm generates a new proposed state



(a) Edge deletion (b) Resulting new districts

Figure 4.7: We select the red edge to delete from the spanning tree, resulting in two new population-balanced districts (DeFord, 2019).

(i.e. a new districting plan), we must progress to phase 4 of ensemble generation to ensure the proposed plan meets the constraints established in phase 1 of ensemble generation. Next, we move to phase 5 of ensemble generation, the Metropolis-Hastings algorithm, to determine whether to accept the proposed districting plan into the ensemble (see section 3.3.5). Recall that the Metropolis-Hastings algorithm relies on the calculation of $\alpha = \frac{s(z)}{s(y)}\frac{q(z|y)}{q(y|z)}$ to determine whether to accept the proposed state z into the ensemble. We calculate the transition probability $q(y|z)$ by first generating the set of all possible spanning trees of the super-district which could have been drawn in stage 3 of the `ReCom` algorithm, and all possible edges which could have been cut in the spanning tree in stage 4 of the `ReCom` algorithm to generate two population-balanced districts. We then calculate the probability that we would select the chosen spanning tree out of all possible spanning trees of the super-district, as well as the probability that we would cut the chosen cut edge out of all possible cut edges which would result in two population-balanced districts. Together, these two probabilities compose $q(y|z)$ ($q(z|y)$ is calculated similarly) (Autry et al., 2020). Once we have calculated $q(y|z)$ and $q(z|y)$, we can easily determine the value of α, which defines the probability with which we will accept the proposed state z from the current state y.

4.3 `ReCom` for ensemble generation

We must run at least 10,000 steps of `ReCom` for multiple different initial plans (as described in section 3.3.2) (Deford et al., 2021). Deford et al. (2021) have established that running at least 10,000 steps of `ReCom` ensures that the ensembles are independent of their seed plans. In this way, we can create a diverse ensemble which effectively represents the set of plausible redistricting plans. We can then perform outlier analysis to determine whether a given redistricting plan is an outlier in the context of our ensemble.

Thus, `ReCom` is a sophisticated algorithm which allows researchers to generate ensembles which are representative of the set of possible districting plans. Compared to other techniques such as `Flip` and `Mix`, `ReCom` generates ensembles which are, on average, substantially more compact, and we can generate ensembles which are sufficiently representative and independent of their seed plans in relatively few steps. These features make `ReCom` a powerful tool in the field of redistricting analysis.

Chapter 5

Conclusion

The preceding chapters detail the process of computational redistricting. By building a dual graph of the geographical units of a state, we can establish the process of redistricting as a graph partition problem. The goal of computational redistricting is to repeatedly mimic the process which legislators go through when they draw a new districting plan, generating plausible district plans until we have an ensemble that is representative of the sample space of plausible district plans.

In order to define the sample space of plausible district plans, we implement constraints which replicate the limitations that legislators face in the redistricting process. Some constraints, such as population balance, contiguity, and neutrality, are easy to represent mathematically, while others, such as compactness and Voting Rights Act compliance, are more difficult to implement in an algorithm, largely due to the fact that legislators' guidelines for these constraints in practice are often unclear. However, in representing these constraints mathematically, researchers in the field of computational redistricting can limit ensemble generation to produce only maps in the sample space of plausible district plans.

After establishing these constraints, researchers then generate an ensemble by selecting a seed plan, proposing a modification on that seed plan, ensuring the modification satisfies the established constraints, and implementing the Metropolis-Hastings algorithm to preferentially accept plans with higher compactness scores into the ensemble. The modification proposal process (along with the steps to ensure the modification satisfies the established constraints and the Metropolis-Hastings algorithm) is repeated many times for each seed plan. Researchers emphasize the necessity of relying on multiple seed plans in order to ensure that the resulting ensemble is independent from its starting point and representative of the sample space of plausible district maps.

Deford et al. (2021) describe both the `Flip` modification proposal algorithm as well as the `ReCom` modification proposal algorithm. Although each step of the `Flip` algorithm is much less computationally intensive than one step of the `ReCom` algorithm, this advantage does not hold over time. The `Flip` algorithm requires at least 1,000,000 steps in order to generate a representative ensemble which is independent from starting point, while in some cases only 10,000 `ReCom` steps are required for a representative ensemble. The `ReCom` algorithm uses spanning trees to generate new plans through recombination, resulting in plans which are naturally compact and thus much more likely to be selected by legislators. `ReCom` is a powerful tool in the field of computational redistricting, as it allows researchers to generate ensembles which are representative of the universe of plausible district maps in relatively few steps.

Once we have generated a representative ensemble of the sample space of plausible district maps, we can perform outlier analysis to determine whether the enacted district map may be a product of gerrymandering. We can perform this analysis by comparing the characteristics of the ensemble (i.e. compactness, partisan split, the distribution of minority voters, population balance, etc.) to the characteristics of the enacted plan. If the enacted plan appears to be an outlier with respect to any of the ensemble's characteristics, it may be a product of gerrymandering. Prior to the existence of computational redistricting practices, it was difficult to prove that district maps were instances of gerrymandering. Computational redistricting is meaningful, in that it provides a method for analytically identifying outlier plans compared to an ensemble of plausible district maps. Computational redistricting presents a non-biased method for identifying potential instances of gerrymandering– this ability can have substantial implications in the field of politics.

For instance, in 2018 Governor Tom Wolf of Pennsylvania rejected a Republican-drawn district map after enlisting Professor Moon Duchin of Tufts University to analyze the fairness of the proposed plan. Using the tools of computational redistricting, Duchin identified that the proposed map was "an extreme outlier along partisan lines," and that the map had "a decidedly partisan skew that cannot be explained by Pennsylvania's political geography or the application of traditional districting principles" (Stern, 2018).

Computational redistricting can therefore be a practical tool which informs politicians of potential bias in proposed or enacted plans, providing a safeguard against gerrymandering.

5.1 Disclaimer

It is worth noting that although this thesis has investigated the potential for computational redistricting to provide protection against potential instances of gerrymandering, the same ensemble generation tools can be used to generate plans which purposefully benefit certain political parties. In a 2004 Supreme Court ruling on standards for partisan gerrymandering, Justice Anthony Kennedy foreshadowed this dilemma by writing, "technology is both a threat and a promise" (Suri & Saxe, 2019).

While ensemble generation can aid certain actors in identifying outlier plans as potential instances of gerrymandering, ensemble generation can also aid other actors in generating a wide array of plausible district maps, from which those actors can choose the map which most benefits them. Ellenberg (2017) writes, "Gerrymandering used to be an art, but advanced computation has made it a science." Ellenberg (2017) references the redistricting process which took place in Wisconsin following the 2010 U.S. Census, in which Republican legislators generated an ensemble of plausible district maps which respected population balance and contiguity, and tested which map would give Republican legislators the highest electoral advantage. Ellenberg (2017) writes, "The map they adopted is precisely engineered to assure Republican control in all but the most extreme circumstances." Mathematicians identified that this plan was an outlier in terms of partisan bias and compactness, but math also facilitated the creation of this sophisticated gerrymandered map in the first place.

Judges are still skeptical of the use of outlier analysis to identify potential instances of gerrymandering, but computational redistricting is of key importance now more than ever. As biased politicians develop more and more sophisticated ways to draw gerrymandered maps, methods of outlier analysis must also become more complex to counter this effort at gerrymandering.

5.2 Tennessee

On the morning of Monday, March 27, 2023, a shooter killed six people (including three children) at the Covenant School in Nashville, Tennessee. This mass shooting galvanized advocates for gun control in Tennessee, and protests took place across the state.

Three Democratic members of the Tennessee House of Representatives, Justin Jones, Justin Pearson, and Gloria Johnson, participated in a protest at the Tennessee State Capitol on March 30, 2023. These three lawmakers, also known as the "Tennessee Three," were accused of "disorderly behavior" by the Republican-led State House of Representatives after they "approached the podium between bills without being recognized to speak" (Jones, Burgess, & Gibson, 2023). The State House sought to expel the "Tennessee three," voting to oust Jones and Pearson, while failing to expel Johnson. Both Jones and Pearson have since been reinstated to the Tennessee House by officials in Nashville and Memphis, respectively.

It is clear that the state of Tennessee is in political crisis. Republican lawmakers are ignoring the wishes of thousands of constituents by silencing their elected lawmakers. Edelman (2023) writes that "redistricting brought Tennessee to this moment." Former Attorney General Eric Holder noted that the efforts of Republican lawmakers to expel the Tennessee Three "would not be possible without first rigging the electoral maps to prevent free and fair elections where the will of the people might be fully expressed" (Edelman, 2023). Many other experts claim that the current political climate in Tennessee is direct result of gerrymandering, particularly following the implementation of new district maps following the 2020 U.S. Census (see figure 1.1). Wines (2023) notes that Tennessee is not alone in the experience of extreme polarization intensified by biased redistricting practices; he cites similar political climates in North Carolina, Wisconsin, Missouri, Ohio, Arkansas, and Florida, each of which is experiencing the political ripple effect of gerrymandering.

Tennessee's redistricting practices motivated this thesis at the start of writing, and they continue to motivate this thesis as writing concludes. Future research would entail conducting outlier analysis on Tennessee's newly enacted congressional district map which was created following the 2020 U.S. Census in order to provide analytical evidence for the many existing claims that the enacted map is a product of partisan gerrymandering. Gerrymandering can be a powerful political tool, but computational redistricting techniques can help counteract the efforts of certain legislators to create district plans with partisan bias.

Acknowledgements

First and foremost, I would like to thank Professor Hardin for her guidance and encouragement throughout the process of writing this thesis. Her support, tough questions, and mathematical expertise helped me gain a depth of understanding of computational redistricting that I would not have otherwise had. I would also like to thank Professor Shahriari for encouraging me to pursue the math major and making me aware of the work that the Metric Geometry and Gerrymandering Group is doing, as our conversation directly inspired my thesis topic. I am grateful to all of my friends at Pomona for their support, especially Kamden Baer, who has helped me learn and grow (in math and in general) throughout my time at Pomona. Finally, I would like to thank my mother, Cynthia Leu, as well as Amy Strauss Josefek. Both of these women have encouraged me to pursue what I am passionate about, to embrace the unconventional, and to develop strong convictions. Their love and support have shaped me, and I am so grateful.

References

Anselin, L. (2020). *Contiguity-based spatial weights.* Retrieved from https://geodacenter.github.io/workbook/4a_contig_weights/lab4a.html#:~:text=Contiguity%20means%20that%20two%20spatial,pieces%20on%20a%20chess%20board.

Autry, E. A., Carter, D., Herschlag, G., Hunter, Z., & Mattingly, J. C. (2020). *Multi-scale merge-split markov chain monte carlo for redistricting* (No. arXiv:2008.08054). arXiv. Retrieved 2023-04-13, from http://arxiv.org/abs/2008.08054

Becker, A., Duchin, M., Gold, D., & Hirsch, S. (2021). Computational redistricting and the voting rights act. , *20*(4), 407–441. Retrieved 2023-04-14, from https://www.liebertpub.com/doi/full/10.1089/elj.2020.0704 (Publisher: Mary Ann Liebert, Inc., publishers) doi: 10.1089/elj.2020.0704

Becker, A., & Solomon, J. (2020). *Redistricting algorithms.* MGGG. Retrieved from https://arxiv.org/abs/2011.09504 doi: 10.48550/ARXIV.2011.09504

Boschma, J., Rigdon, R., Manley, B., & Cohen, E. (2022). Redistricting in tennessee. Retrieved 2023-02-25, from https://www.cnn.com/interactive/2022/politics/us-redistricting/tennessee-redistricting-map/

Caldera, S., DeFord, D., Duchin, M., Gutekunst, S. C., & Nix, C. (2020). Mathematics of nested districts: The case of alaska. , *7*(1), 39–51. Retrieved 2023-02-24, from https://www.tandfonline.com/doi/citedby/10.1080/2330443X.2020.1774452 (Publisher: Taylor & Francis) doi: 10.1080/2330443X.2020.1774452

Chen, J., & Stephanopoulos, N. O. (2021). The race-blind future of voting rights. , *130*(4), 862–946. Retrieved 2023-02-23, from https://www.yalelawjournal.org/article/the-race-blind

-future-of-voting-rights

DeFord, D. (2019). Computational Approaches for Political Redistricting Part II: MCMC and GerryChain. Massachussetts Institute of Technology. Retrieved from https://people.csail.mit.edu/ddeford//IAP_2019_Part2.pdf

DeFord, D., & Duchin, M. (2022). Random walks and the universe of districting plans. In M. Duchin & O. Walch (Eds.), *Political geometry* (pp. 341–381). Springer International Publishing. Retrieved 2023-04-03, from https://link.springer.com/10.1007/978-3-319-69161-9_17 doi: 10.1007/978-3-319-69161-9_17

Deford, D., Duchin, M., & Solomon, J. (2021, mar 31). Recombination: A Family of Markov Chains for Redistricting. *Harvard Data Science Review*, *3*(1). (https://hdsr.mitpress.mit.edu/pub/1ds8ptxu)

Democracy Docket. (2021a). *Alabama congressional redistricting challenge (SCOTUS).* Retrieved 2023-04-14, from https://www.democracydocket.com/cases/alabama-congressional-redistricting-challenge-scotus/

Democracy Docket. (2021b). *Protecting minority voting rights.* Retrieved 2023-04-14, from https://www.democracydocket.com/analysis/protecting-minority-voting-rights/

Duchin, M., & Spencer, D. (2021). Models, race, and the law. *Yale Law Journal Forum*, *130*, 744–797.

Duchin, M., & Tenner, D. M. (2021). Models, race, and the law. , *130*. Retrieved from https://www.yalelawjournal.org/forum/models-race-and-the-law

Edelman, A. (2023). *How redistricting brought tennessee to this moment.* Retrieved 2023-04-14, from https://www.nbcnews.com/politics/politics-news/redistricting-brought-tennessee-moment-rcna78953

Ellenberg, J. (2017). How computers turned gerrymandering into a science. Retrieved 2023-04-14, from https://www.nytimes.com/2017/10/06/opinion/sunday/computers-gerrymandering-wisconsin.html

Hebert, J. G., Smith, P. M., Vandenberg, M. E., & DeSanctis, M. B. (2010). *The realist's guide to redistricting: Avoiding the legal pitfalls* (2nd ed.). American Bar Association.

Jones, V., Burgess, K., & Gibson, R. (2023). *Tennessee three: What to know about the house democrats targeted for expulsion by republicans.* Retrieved 2023-04-14, from https://www.tennessean.com/story/

news/politics/2023/04/07/tennessee-house-expulsion-gloria-johnson-justin-jones-justin-pearson/70080338007/

Metric Geometry and Gerrymandering Group. (2018). *Comparison of Districting Plans for the Virginia House of Delegates.* Retrieved from https://mggg.org/VA-report.pdf

Mumpower, J. E. (2021, May). *A Guide to Local Government Redistricting in Tennessee* (Tech. Rep.). Tennessee Comptroller of the Treasury. Retrieved from https://comptroller.tn.gov/content/dam/cot/pa/documents/lg/GuideToRedistrict.pdf

Scales, L. (2020). *Why chicago's 4th congressional district — those 'earmuffs' — are about fairness and not gerrymandering.* Retrieved 2023-04-14, from https://chicago.suntimes.com/2020/3/11/21164850/illinois-4th-congressional-district-gerrymandering-voting-rights-act-census-liliana-scales (Section: Other Views)

Schweinsberg, J. (2008). The loop-erased random walk and the uniform spanning tree on the four-dimensional discrete torus. University of California at San Diego. Retrieved from https://mathweb.ucsd.edu/~jschwein/LERW.pdf

Shahriari, S. (2021). *An invitation to combinatorics.* Cambridge University Press.

Stern, M. J. (2018). Pennsylvania governor rejects latest GOP gerrymander, ensuring court-drawn map for midterms. Retrieved 2023-04-14, from https://slate.com/news-and-politics/2018/02/pennsylvania-governor-rejects-new-gop-gerrymander-ensuring-court-drawn-map-for-2018.html (Section: The Slatest)

Suri, M., & Saxe, K. (2019). *Want to fix gerrymandering? then the supreme court needs to listen to mathematicians.* Retrieved 2023-04-14, from https://www.pbs.org/newshour/nation/want-to-fix-gerrymandering-then-the-supreme-court-needs-to-listen-to-mathematicians (Section: Nation)

U.S. Bureau of the Census. (1994, November). Chapter 14: Voting Districts. In *Geographic Areas Reference Manual.* U.S. Department of Commerce. Retrieved from https://www2.census.gov/geo/pdfs/reference/GARM/Ch14GARM.pdf

U.S. Bureau of the Census. (2013). *Guide to state and local census geography.* Retrieved from https://www.census.gov/geographies/reference-files/2010/geo/state-local-geo-guides-2010.html

U.S. Bureau of the Census. (2023). *About Congressional Districts.*

Retrieved from https://www.census.gov/programs-surveys/geography/guidance/geo-areas/congressional-dist.html#:~:text=Each%20congressional%20district%20is%20to,the%20districts%20within%20each%20state.

Voting rights act. (1965). Retrieved 2023-04-14, from https://www.archives.gov/milestone-documents/voting-rights-act

Weisstein, E. W. (2023a). *Geometric dual graph* [Text]. Retrieved 2023-02-14, from https://mathworld.wolfram.com/ (Publisher: Wolfram Research, Inc.)

Weisstein, E. W. (2023b). *Planar graph* [Text]. Retrieved 2023-02-14, from https://mathworld.wolfram.com/ (Publisher: Wolfram Research, Inc.)

Wilson, D. B. (1996). Generating Random Spanning Trees More Quickly than the Cover Time. In *Proceedings of the twenty-eighth annual acm symposium on theory of computing* (p. 296–303). New York, NY, USA: Association for Computing Machinery. Retrieved from https://doi.org/10.1145/237814.237880 doi: 10.1145/237814.237880

Wines, M. (2023). If tennessee's legislature looks broken, it's not alone. Retrieved 2023-04-14, from https://www.nytimes.com/2023/04/13/us/tennessee-house-republicans.html