

# Sequential Monte Carlo for Sampling Balanced and Compact Redistricting Plans\*

Cory McCartan<sup>†</sup>      Kosuke Imai<sup>‡</sup>

December 25, 2020

## Abstract

Random sampling of graph partitions under constraints has become a popular tool for evaluating legislative redistricting plans. Analysts detect partisan gerrymandering by comparing a proposed redistricting plan with an ensemble of sampled alternative plans. For successful application, sampling methods must scale to large maps with many districts, incorporate realistic legal constraints, and accurately and efficiently sample from a selected target distribution. Unfortunately, most existing methods struggle in at least one of these three areas. We present a new Sequential Monte Carlo (SMC) algorithm that draws representative redistricting plans from a realistic target distribution of choice. Because it samples directly, the SMC algorithm can efficiently explore the relevant space of redistricting plans better than the existing Markov chain Monte Carlo algorithms that yield dependent samples. Our algorithm can simultaneously incorporate several constraints commonly imposed in real-world redistricting problems, including equal population, compactness, and preservation of administrative boundaries. We validate the accuracy of the proposed algorithm by using a small map where all redistricting plans can be enumerated. We then apply the SMC algorithm to evaluate the partisan implications of several maps submitted by relevant parties in a recent high-profile redistricting case in the state of Pennsylvania. We find that the proposed algorithm is roughly 40 times more efficient in sampling from the target distribution than a state-of-the-art MCMC algorithm. Open-source software is available for implementing the proposed methodology.

**Key Words:** gerrymandering, graph partition, importance sampling, spanning trees

---

\*We thank Ben Fifield, Greg Herschlag, Mike Higgins, Chris Kenny, Jonathan Mattingly, Justin Solomon, and Alex Tarr for helpful comments and conversations. Imai thanks Yunkyu Sohn for his contributions at an initial phase of this project. Open-source software is available for implementing the proposed methodology (Fifield et al., 2020c).

<sup>†</sup>Ph.D. student, Department of Statistics, Harvard University. 1 Oxford Street, Cambridge 02138. Email: [cmccartan@g.harvard.edu](mailto:cmccartan@g.harvard.edu)

<sup>‡</sup>Professor, Department of Government and Department of Statistics, Harvard University. 1737 Cambridge Street, Institute for Quantitative Social Science, Cambridge 02138. Email: [imai@harvard.edu](mailto:imai@harvard.edu), URL: <https://imai.fas.harvard.edu/>

# 1 Introduction

In first-past-the-post electoral systems, legislative districts serve as the fundamental building block of democratic representation. In the United States, congressional redistricting, which redraws district boundaries in each state following decennial Census, plays a central role in influencing who is elected and hence what policies are eventually enacted. Because the stakes are so high, redistricting has been subject to intense political battles. Parties often engage in *gerrymandering* by manipulating district boundaries in order to amplify the voting power of some groups while diluting that of others.

In recent years, the availability of granular data about individual voters has led to sophisticated partisan gerrymandering attempts that cannot be easily detected. At the same time, many scholars have focused their efforts on developing methods to uncover gerrymandering by comparing a proposed redistricting plan with a large collection of alternative plans that satisfy the relevant legal requirements. A primary advantage of such an approach over the use of simple summary statistics is its ability to account for idiosyncrasies of physical and political geography specific to each state.

For its successful application, a sampling algorithm for drawing alternative plans must (1) scale to large maps with thousands of geographic units and many districts, (2) simultaneously incorporate a variety of real-world legal constraints such as population balance, geographical compactness, and the preservation of administrative boundaries, (3) efficiently yield a large number of plans that are as independent of one another as possible, and (4) ensure these samples are representative of a specific target population, against which a redistricting plan of interest can be evaluated. Although some have been used in several recent court challenges to existing redistricting plans, existing algorithms run into limitations with regards to at least one of these four key requirements.

Optimization-based (e.g., [Mehrotra et al., 1998](#); [Macmillan, 2001](#); [Bozkaya et al., 2003](#); [Liu et al., 2016](#)) and constructive Monte Carlo (e.g., [Cirincione et al., 2000](#); [Chen and Rod-](#)

den, 2013; Magleby and Mosesson, 2018) methods can be made scalable and incorporate many constraints. But they are not designed to sample from any specific target distribution. As a result, the resulting plans tend to differ systematically, for example, from a uniform distribution under certain constraints (Cho and Liu, 2018; Fifield et al., 2020a,b). The absence of an explicit target distribution makes it difficult to interpret the ensembles generated by these methods and use them for statistical outlier analysis to detect gerrymandering.

MCMC algorithms (e.g., Mattingly and Vaughn, 2014; Wu et al., 2015; Chikina et al., 2017; DeFord et al., 2019; Carter et al., 2019; Fifield et al., 2020a) can in theory sample from a specific target distribution, and incorporate constraints through the use of an energy function. In practice, however, existing algorithms struggle to mix and traverse through a highly complex sampling space, making scalability difficult and accuracy hard to prove. Some of these algorithms make proposals by flipping precincts at the boundary of existing districts (e.g., Mattingly and Vaughn, 2014; Fifield et al., 2020a), rendering it difficult to transition between points in the state space, especially as more constraints are imposed. More recent algorithms by DeFord et al. (2019) and Carter et al. (2019) use spanning trees to make their proposals, and this has allowed these algorithms to yield more global moves and improve mixing. Yet the very essence of the MCMC approach is to generate dependent samples, and on large-scale problems, this dependence may lead to low efficiency.

In Sections 3 and 4, we present a new Sequential Monte Carlo (SMC) algorithm, based on a similar spanning tree construction to DeFord et al. (2019) and Carter et al. (2019), that addresses the above four key challenges. Unlike optimization-based and constructive Monte Carlo methods, the SMC algorithm samples from a specific and customizable target distribution. Our algorithm scales better than MCMC algorithms because it generates diverse, high-quality samples while directly incorporating the three most common constraints imposed on the redistricting process—contiguity, population balance, and geographic compactness. SMC also removes the need to draw enough samples to ensure the entire sample

space is explored, which is essential for the successful application of MCMC algorithms.

The proposed algorithm proceeds by splitting off one district at a time, building up the redistricting plan piece by piece (see Figure 3 for an illustration). Each split is accomplished by drawing a spanning tree and removing one edge, which splits the spanning tree in two. We also extend the SMC algorithm so that it preserves administrative boundaries and certain geographical areas as much as possible, which is another common constraint considered in many real-world redistricting cases. An open-source software package is available for implementing the proposed algorithm (Fifield et al., 2020c).

In Section 5, we validate the SMC algorithm using a 50-precinct map, for which all potential redistricting plans can be enumerated (Fifield et al., 2020b). We demonstrate that the proposed algorithm samples accurately from a range of target distributions on these plans. Section 6 applies the SMC algorithm to the 2011 Pennsylvania congressional redistricting, which was subject to a 2017 court challenge, and compares its performance on this problem with the existing approaches. We find that the proposed SMC algorithm is roughly 40 times more efficient in sampling than a state-of-the-art MCMC algorithm applied to the same problem. Section 7 concludes and discusses directions for future work. We now introduce the Pennsylvania case, which serves as our motivating empirical application.

## 2 The 2011 Pennsylvania Congressional Redistricting

We study the 2011 Pennsylvania congressional redistricting because it illustrates the salient features of the redistricting problem. We begin by briefly summarizing the background of this case and then explain the role of sampling algorithms used in the expert witness reports.

### 2.1 Background

Pennsylvania lost a seat in Congress during the reapportionment of the 435 U.S. House seats which followed the 2010 Census. In Pennsylvania, the General Assembly, which is the state’s legislative body, draws new congressional districts, subject to gubernatorial veto.

At the time, the General Assembly was controlled by Republicans, and Tom Corbett, also a Republican, served as governor. In the 2012 election, which took place under the newly adopted 2011 districting map, Democrats won 5 House seats while Republicans took the remaining 13. This result stands in sharp contrast to a 7–12 split after the 2010 election as well as a 12–7 Democratic advantage before 2010.

In June 2017, the League of Women Voters filed a lawsuit alleging that the 2011 plan adopted by the Republican legislature violated the state constitution by diluting the political power of Democratic voters. The case worked its way through the state court system, and on January 22, 2018, the Pennsylvania Supreme Court issued its ruling, writing that the 2011 plan “clearly, plainly and palpably violates the Constitution of the Commonwealth of Pennsylvania, and, on that sole basis, we hereby strike it as unconstitutional.”

The court ordered that the General Assembly adopt a remedial plan and submit it to the governor, who would in turn submit it to the court, by February 15, 2018. In case no plan was submitted to and approved by that deadline, the court provided that all of the parties to the lawsuit could submit their own plans by the same date, and the court would review them and itself impose a final remedial plan. In its ruling, the court laid out specific requirements that must be satisfied by all proposed plans:

any congressional districting plan shall consist of: congressional districts composed of compact and contiguous territory; as nearly equal in population as practicable; and which do not divide any county, city, incorporated town, borough, township, or ward, except where necessary to ensure equality of population.

While the compactness and administrative boundary constraints are not required by the U.S. or Pennsylvania constitutions, they have been historically held up as “guiding principles” in many states and were explicitly imposed by the court in this case.

The leaders of the Republican Party in the General Assembly drew a new map, but the Democratic governor, Tom Wolf, refused to submit it to the court, claiming that it, too, was an unconstitutional gerrymander. Instead, the court received remedial plans from seven

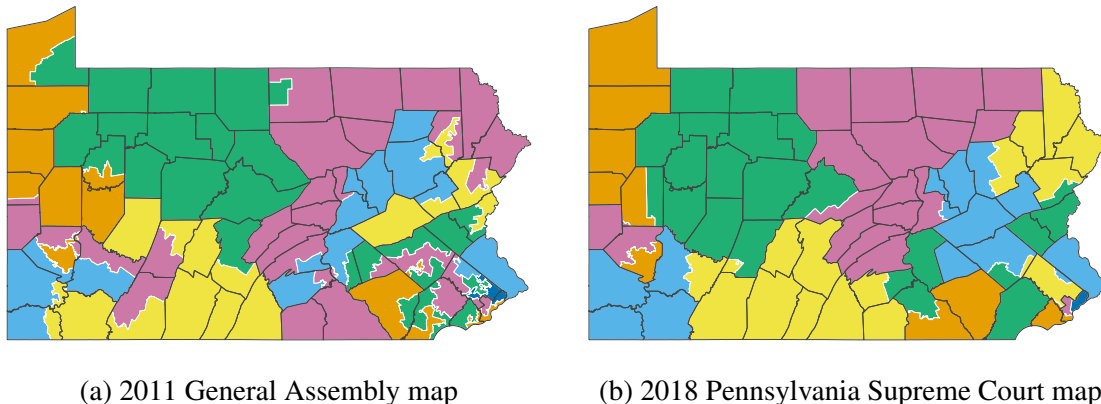


Figure 1: Comparison of the 2011 map drawn by the General Assembly and the final map imposed by the Supreme court in 2018. County lines are shown in dark gray, and district boundaries that do not coincide with county boundaries are in white.

parties: the petitioners, the League of Women Voters; the respondents, the Republican leaders of the General Assembly; the governor, a Democrat; the lieutenant governor, also a Democrat; the Democratic Pennsylvania House minority leadership; the Democratic Pennsylvania Senate minority leadership; and the intervenors, which included Republican party candidates and officials. Ultimately, the Supreme Court drew its own plan and adopted it on February 19, 2018, arguing that it was “superior or comparable to all plans submitted by the parties.” Figure 1 shows the remedial plan created by the Supreme Court as well as the 2011 map adopted by the General Assembly, which were found on the court’s case page.

The constraints explicitly laid out by the court, as well as the numerous remedial plans submitted by the parties, make the 2011 Pennsylvania redistricting a useful case study that evaluates redistricting plans using sampling algorithms.

## 2.2 The Role of Sampling Algorithms

The original finding that the 2011 General Assembly plan was a partisan gerrymander was in part based on different outlier analyses performed by two academic researchers, Jowei Chen and Wesley Pegden, who served as the petitioner’s expert witnesses. Chen randomly generated two sets of 500 redistricting plans according to a constructive Monte Carlo algorithm based on [Chen and Rodden \(2013\)](#). He considered population balance,

contiguity, compactness, avoiding county and municipal splits, and, in the second set of 500, avoiding pairing off incumbents. Pegden ran an MCMC algorithm for one trillion steps, and computed upper bounds of  $p$ -values using the method of [Chikina et al. \(2017\)](#). This method was also used in a follow-up analysis by Moon Duchin, who served as an expert for the governor ([Duchin, 2018](#)). Both petitioner experts concluded that the 2011 plan was an extreme outlier according to compactness, county and municipal splits, and the number of Republican and Democratic seats implied by statewide election results.

The respondents also retained several academic researchers as their expert witnesses. One of them, Wendy Tam Cho, directly addressed the sampling-based analyses of Chen and Pegden. Cho criticized Chen’s analysis for not sampling from a specified target distribution. She also criticized Pegden’s analysis by arguing that his MCMC algorithm only made local explorations of the space of redistricting plans, and could not therefore have generated a representative sample of all valid plans (see also [Cho and Rubinstein-Salzedo, 2019](#), and [Chikina et al. \(2019\)](#)). We do not directly examine the intellectual merits of the specific arguments put forth by the expert witnesses. However, these methodological debates are also relevant for other cases where simulation algorithms have been extensively used by expert witnesses (e.g., [Rucho v. Common Cause \(2019\)](#); [Common Cause v. Lewis \(2019\)](#); [Covington v. North Carolina \(2017\)](#); [Harper v. Lewis \(2020\)](#)).

The expert witness reports in the Pennsylvania case highlight the difficulties in practically applying existing sampling algorithms to actual redistricting problems. First, the distributions that some of these algorithms sample from are not made explicit, leaving open the possibility that the generated ensemble is systematically different from the true set of all valid plans. Second, even when the distribution is known, the MCMC algorithms used to sample from it may be prohibitively slow to mix and cannot yield a representative sample. These challenges motivate us to design an algorithm that generates more diverse samples from a specific target distribution and incorporates most common redistricting constraints,

while minimizing the impact on scalability, theoretical validity, and empirical performance.

### 3 Sampling Balanced and Compact Districts

In this section, we formally characterize the target distribution of our sampling algorithm. Our goal is to sample redistricting plans with contiguous districts which are both balanced in population and geographically compact.

#### 3.1 The Setup

Redistricting plans are ultimately aggregations of geographic units such as counties, voting precincts, or Census blocks. The usual requirement that the districts in a plan be contiguous necessitates consideration of the spatial relationship between these units. The natural mathematical structure for this consideration is a graph  $G = (V, E)$ , where  $V = \{v_1, v_2, \dots, v_m\}$  consists of  $m$  nodes representing the geographic units of redistricting and  $E$  contains edges connecting units which are legally adjacent.

A redistricting plan on  $G$  consisting of  $n$  districts is described by a function  $\xi : V \rightarrow \{1, 2, \dots, n\}$ , where  $\xi(v) = i$  implies that node  $v$  is in district  $i$ . We let  $V_i(\xi)$  and  $E_i(\xi)$  denote the nodes and edges contained in district  $i$  under a given redistricting plan  $\xi$ , so  $G_i(\xi) = (V_i(\xi), E_i(\xi))$  represents the induced subgraph that corresponds to district  $i$  under the plan. We sometimes suppress the dependence on  $\xi$  when this is clear from context, writing  $G_i = (V_i, E_i)$ . Since each node belongs to one and only one district, we have  $V = \bigcup_{i=1}^n V_i(\xi)$  and  $V_i(\xi) \cap V_{i'}(\xi) = \emptyset$  for any redistricting plan  $\xi$ . In addition, we require that all nodes of a given district be connected.

Beyond connectedness, redistricting plans are almost always required to have nearly equal population in every district. To formalize this requirement, let  $\text{pop}(v)$  denote the population of node  $v$ . Then the population of a district is given by

$$\text{pop}(V_i) := \sum_{v \in V_i(\xi)} \text{pop}(v).$$



We quantify the discrepancy between a given plan and this ideal by the *maximum population deviation* of the plan,

$$\text{dev}(\xi) := \max_{1 \leq i \leq n} \left| \frac{\text{pop}(V_i)}{\text{pop}(V)/n} - 1 \right|,$$

where  $\text{pop}(V)$  denotes the total population. Some courts and states have imposed hard maximums on this quantity, e.g.,  $\text{dev}(\xi) \leq D = 0.05$  for state legislative redistricting.

The proposed algorithm samples plans by way of spanning trees on each district, i.e., subgraphs of  $G_i(\xi)$  which contain all vertices, no cycles, and are connected. Let  $T_i$  represent a spanning tree for district  $i$  whose vertices and edges are given by  $V_i(\xi)$  and a subset of  $E_i(\xi)$ , respectively. The collection of spanning trees from all districts together form a spanning forest  $F = (T_1, \dots, T_n)$ . Each spanning forest implies a redistricting plan where  $\xi(F)(v) = i$  for all  $v \in T_i$ . However, a single redistricting plan may correspond to multiple spanning forests because each district may have more than one spanning tree.

For a given redistricting plan, we can compute the exact number of spanning forests in polynomial time using the determinant of a submatrix of the graph Laplacian, according to the Matrix Tree Theorem of Kirchhoff (see [Tutte \(1984\)](#)). Thus, for a graph  $H$ , if we let  $\tau(H)$  denote the number of spanning trees on the graph, we can represent the number of spanning forests that correspond to a redistricting plan  $\xi$  as

$$\tau(\xi) := \prod_{i=1}^n \tau(G_i(\xi)).$$

This fact will play an important role in the definition of our sampling algorithm and its target distribution, as we explain next.

### 3.2 The Target Distribution

The algorithm is designed to sample a plan  $\xi$  with probability

$$\pi(\xi) \propto \exp\{-J(\xi)\} \tau(\xi)^\rho \mathbf{1}_{\{\xi \text{ connected}\}} \mathbf{1}_{\{\text{dev}(\xi) \leq D\}}, \quad (1)$$

where the indicator functions ensure that the plans meet population balance and connectedness criteria,  $J$  encodes additional constraints on the types of plans preferred, and  $\rho \in \mathbb{R}_0^+$

is chosen to control the compactness of the generated plans. As done in Section 6, we often use a reasonably strict population constraint such as  $D = 0.001$ . But, when no such constraint exists,  $D$  can be set to its maximum value  $n - 1$ , rendering the constraint moot.

This class of target distributions is the same as what the algorithm of [Herschlag et al. \(2017\)](#) is designed to sample from. The lack of restrictions on  $J$  means that by setting  $\rho = 0$  and choosing an appropriate  $J$ , any target distribution supported on connected plans which meet hard population constraint may be obtained. Of course, the algorithm operates efficiently only when the additional constraints imposed by  $J$  are not too severe. In practice, we find that the most stringent constraints are those involving population deviation, compactness, and administrative boundary splits. As shown later, we address this issue by designing our algorithm to directly satisfy these constraints. Additional constraints do not generally have a substantial effect on the sampling efficiency.

The generality of the additional constraint function  $J$  is intentional, as its exact form and number imposed on the redistricting process varies by state and by the type of districts being drawn. Hard constraints, such as requiring that a certain number of majority-minority districts exist, may be realized by setting  $J(\xi) = 0$  for valid plans, and  $J(\xi) = \infty$  for plans that violate the constraints. Softer constraints may be incorporated by choosing a  $J$  which is small for preferred plans and large otherwise.

For example, a preference for plans close to an existing plan  $\xi_{sq}$  may be encoded as

$$\begin{aligned} J_{sq}(\xi) &= -\frac{\beta}{\log n} \text{VI}(\xi, \xi_{sq}) \\ &:= -\frac{\beta}{\log n} \sum_{i,j=1}^n \frac{\text{pop}(V_i(\xi) \cap V_j(\xi_{sq}))}{\text{pop}(V)} \left( \log \left( \frac{\text{pop}(V_i(\xi) \cap V_j(\xi_{sq}))}{\text{pop}(V_j(\xi_{sq}))} \right) + \log \left( \frac{\text{pop}(V_i(\xi) \cap V_j(\xi_{sq}))}{\text{pop}(V_i(\xi))} \right) \right), \end{aligned} \quad (2)$$

where  $\beta \in \mathbb{R}^+$  controls the strength of the constraint. The function  $\text{VI}(\cdot, \cdot)$  represents the variation of information (also known as the shared information distance), which is the difference between the joint entropy and the mutual information of the distribution of population over the new districts  $\xi$  relative to the existing districts  $\xi_{sq}$  ([Cover and Thomas, 2006](#)). When  $\xi$  is any relabelling of  $\xi_{sq}$ , then  $J_{sq}(\xi) = 0$ . In contrast, when  $\xi$  evenly splits

the nodes of each district of  $\xi_{sq}$  between the districts of  $\xi$ , then  $J_{sq}(\xi) = \beta$ . This distance measure will prove useful later in measuring the diversity of a sample of redistricting plans.

If instead we wish to encode a soft preference for plans which split as few administrative units as possible, we could use

$$J_{spl}(\xi) = \begin{cases} \beta \text{ spl}(\xi) & \text{if } \text{spl}(\xi) \leq s_{\max} \\ \infty & \text{otherwise} \end{cases},$$

where  $\text{spl}(\cdot)$  is the number of administrative splits (formally defined in Section 4.5), and  $\beta \in \mathbb{R}^+$  again controls the strength of the soft constraint and  $s_{\max}$  is the maximum number of administrative splits allowed.

There exist other formulations of constraints, and considerations in choosing a set of weights that balance constraints against each other (see e.g., [Bangia et al., 2017](#); [Herschlag et al., 2017](#); [Fifield et al., 2020a](#)). Here, we focus on sampling from the broad class of distributions characterized by equation (1), which have been used in other work; we do not address the important but separate problem of picking a specific instance of this class to apply to a given redistricting problem.

Even a small number of constraints incorporated into  $J$  can dramatically limit the number of valid plans and considerably complicate the process of sampling. The Markov chain algorithms developed to date partially avoid this problem by moving toward maps with lower  $J$  over a number of steps, but in general including more constraints makes it even more difficult to transition between valid redistricting plans. Approaches such as simulated annealing ([Bangia et al., 2017](#); [Herschlag et al., 2017](#)) and parallel tempering ([Fifield et al., 2020a](#)) have been proposed to handle multiple constraints, but these can be difficult to calibrate in practice and provide few, if any, theoretical guarantees.

When only hard constraints are used and  $\rho = 0$  in equation (1), the distribution is uniform across all plans satisfying the constraints. As we will discuss next, larger values of  $\rho$  create a preference for more compact districts;  $\rho = 1$  is a computationally convenient

choice which produces satisfactorily compact districts.

### 3.3 Spanning Forests and Compactness

One common redistricting requirement is that districts be geographically compact, though nearly every state leaves this term undefined. Dozens of numerical compactness measures have been proposed, with the Polsby–Popper score (Polsby and Popper, 1991) perhaps the most popular. Defined as the ratio of a district’s area to that of a circle with the same perimeter as the district, the Polsby–Popper score is constrained to  $[0, 1]$ , with higher scores indicating more compactness.

The Polsby–Popper score has been shown to correlate reasonably well with humans’ subjective evaluation (Kaufman et al., 2020), but it is far from a perfect measure. One challenge in particular is its sensitivity to the underlying geography and the scale on which it is measured. In states with rugged coastlines or other geographical irregularities, even the most compact districts could have Polsby–Popper scores that are lower than those of gerrymandered districts in other states. And measuring the geography at a 1-meter versus a 1-kilometer scale can shift the Polsby–Popper scores dramatically. This sensitivity makes it difficult to compare the compactness redistricting plans across states.

To address this challenge, some have proposed a graph-theoretic measure known as *edge-cut compactness* (Dube and Clark, 2016; DeFord et al., 2019). This measure counts the number of edges that must be removed from the original graph to partition it according to a given plan. Formally, it is defined as

$$\text{rem}(\xi) := 1 - \frac{\sum_{i=1}^n |E_i(\xi)|}{|E(G)|},$$

where we have normalized to the total number of edges.

Plans that involve cutting many edges will necessarily have long internal boundaries, driving up their average district perimeter (and driving down their Polsby–Popper scores), while plans that cut as few edges as possible will have relatively short internal boundaries

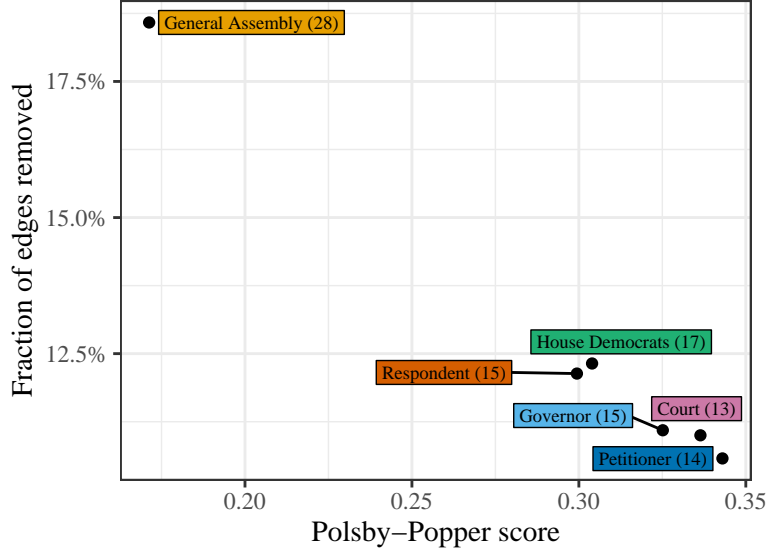


Figure 2: Compactness measures for the six plans from the 2011 Pennsylvania redistricting case. The number of counties split by each plan is given in parentheses within each label.

and much more compact districts. Additionally, given the high density of voting units in urban areas, plans which cut fewer edges will tend to avoid drawing district lines through the heart of these urban areas. This has the welcome side effect of avoiding splitting cities and towns, and in doing so helping to preserve “communities of interest,” another common redistricting consideration. The agreement between this measure and the more traditional Polsby–Popper score is clearly visible in Figure 2, which shows the compactness and number of county splits for various maps in the 2011 Pennsylvania congressional redistricting.

Empirically, this graph-based compactness measure tends to be highly correlated with  $\log \tau(G) - \log \tau(\xi)$ . Indeed, in practice, we often observe a correlation in excess of 0.99. It is difficult to precisely characterize this relationship except in special cases because  $\tau(\xi)$  is calculated as a matrix determinant (McKay, 1981). However, it is well known that this quantity is strongly controlled by the product of the degrees of each node in the graph,  $\prod_{i=1}^m \deg(v_i)$  (Kostochka, 1995). Removing an edge from a graph decreases the degree of the vertices at either end by one, so we would expect  $\log \tau(G)$  to change by approximately  $2\{\log \bar{d} - \log(\bar{d} - 1)\}$  with this edge removal, where  $\bar{d}$  is the average degree of the graph. This implies a linear relation  $\log \tau(G) - \log \tau(\xi) \approx \text{rem}(\xi) \cdot 2\{\log \bar{d} - \log(\bar{d} - 1)\}$ , and

hence

$$\tau(\xi)^\rho \propto \exp(-C \rho \text{rem}(\xi)),$$

where  $C$  is an arbitrary constant.

As a result, a greater value of  $\rho$  in the target distribution corresponds to a preference for fewer edge cuts and therefore a redistricting plan with more compact districts. This and the considerations given in the literature (Dube and Clark, 2016; DeFord et al., 2019) suggest that the target distribution in equation (1) with  $\rho = 1$  (or another positive value) is a good choice for sampling compact districts. Of course, if another compactness metric is desired, one can simply set  $\rho = 0$  and incorporate the alternative metric into  $J$ . This will preserve the algorithm's efficiency to the extent that the alternative metric correlates with the edge-removal measure of compactness.

## 4 The Proposed Algorithm

The proposed algorithm samples redistricting plans by sequentially drawing districts over  $n - 1$  iterations of a splitting procedure. The algorithm begins by partitioning the original graph  $G = (V, E)$  into two induced subgraphs:  $G_1 = (V_1, E_1)$ , which will constitute a district in the final map, and the remainder of the graph  $\tilde{G}_1 = (\tilde{V}_1, \tilde{E}_1)$ , where  $\tilde{V}_1 = V \setminus V_1$  and  $\tilde{E}_1$  consists of all the edges between vertices in  $\tilde{V}_1$ . Next, the algorithm takes  $\tilde{G}_1$  as an input graph and partitions it into two induced subgraphs, one which will become a district  $G_2$  and the remaining graph  $\tilde{G}_2$ . The algorithm repeats the same splitting procedure until the final  $(n - 1)$ -th iteration whose two resulting partitions,  $G_{n-1}$  and  $\tilde{G}_{n-1} = G_n$ , become the final two districts of the redistricting plan.

Figure 3 provides an illustration of this sequential procedure. To sample a large number of redistricting plans from the target distribution given in equation (1), at each iteration, the algorithm samples many candidate partitions, discards those which fail to meet the population constraint, and then resamples a certain number of the remainder according to

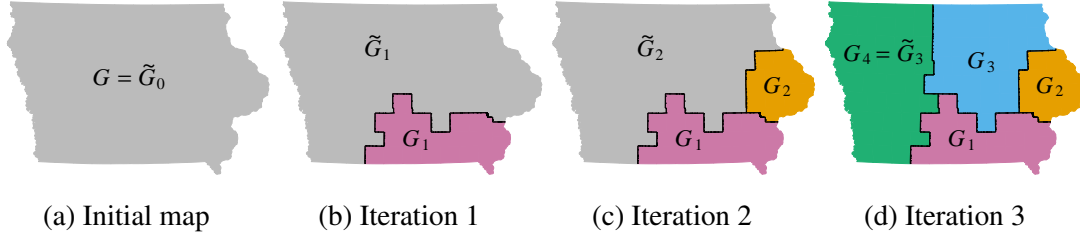


Figure 3: The sequential splitting procedure applied to the state of Iowa, where four congressional districts are created at the county level.

importance weights, using the resampled partitions at the next iteration. The rest of this section explains the details of the proposed algorithm.

## 4.1 The Splitting Procedure

We first describe the splitting procedure, which is similar to the merge-split Markov chain proposals of [DeFord et al. \(2019\)](#) and [Carter et al. \(2019\)](#). It proceeds by drawing a random spanning tree  $T$ , identifying the  $k_i$  most promising edges to cut within the tree, and selecting one such edge at random to create two induced subgraphs. Spanning trees are an attractive way to split districts, as the removal of a single edge induces a partition with two connected components, and spanning trees can be sampled uniformly ([Wilson, 1996](#)).

After splitting, the resulting partition is checked for compliance with the population constraint, by ensuring the population of the new district  $G_i$  falls within the bounds

$$P_i^- = \max \left\{ \frac{\text{pop}(V)}{n}(1 - D), \text{pop}(\tilde{V}_{i-1}) - \frac{n-i}{n} \text{pop}(V)(1 + D) \right\} \quad \text{and} \\ P_i^+ = \min \left\{ \frac{\text{pop}(V)}{n}(1 + D), \text{pop}(\tilde{V}_{i-1}) - \frac{n-i}{n} \text{pop}(V)(1 - D) \right\}.$$

These bounds ensure not only that the new district has population deviation below  $D$ , but that it will be possible for future iterations to generate valid districts out of  $\tilde{G}_i$ . If  $\text{pop}(V_i) \notin [P_i^-, P_i^+]$ , then the entire redistricting plan is rejected and the sampling process begins again. While the rate of rejection varies by map and by iteration, we generally encounter acceptance rates between 5% and 30%, which are not so low as to make sampling from large maps intractable. Algorithm 1 details the steps of the splitting procedure, where we

---

**Algorithm 1** Splitting procedure to generate one district

---

*Input:* initial graph  $\tilde{G}_{i-1}$  and a parameter  $k_i \in \mathbb{Z}^+$ .

- (a) Draw a single spanning tree  $T$  on  $\tilde{G}_{i-1}$  uniformly from the set of all such trees using Wilson's algorithm.
- (b) Each edge  $e \in E(T)$  divides  $T$  into two components,  $T_e^{(1)}$  and  $T_e^{(2)}$ . For each edge, compute the following population deviation for the two districts that would be induced by cutting  $T$  at  $e$ ,

$$d_e^{(1)} = \left| \frac{\sum_{v \in T_e^{(1)}} \text{pop}(v)}{\text{pop}(V)/n} - 1 \right| \quad \text{and} \quad d_e^{(2)} = \left| \frac{\sum_{v \in T_e^{(2)}} \text{pop}(v)}{\text{pop}(V)/n} - 1 \right|.$$

Let  $d_e = \min\{d_e^{(1)}, d_e^{(2)}\}$ , and index the edges in ascending order by this quantity, so that we have  $d_{e_1} \leq d_{e_2} \leq \dots \leq d_{e_{m_i-1}}$ , where  $m_i = |V_i|$ .

- (c) Select one edge  $e^*$  uniformly from  $\{e_1, e_2, \dots, e_{k_i}\}$  and remove it from  $T$ , creating a spanning forest  $(T_{e^*}^{(1)}, T_{e^*}^{(2)})$  which induces a partition  $(G_i^{(1)}, G_i^{(2)})$ .
  - (d) If  $d_{e^*}^{(1)} \leq d_{e^*}^{(2)}$ , i.e., if  $T_{e^*}^{(1)}$  induces a district that is closer to the optimal population than  $T_{e^*}^{(2)}$  does, set  $G_i = G_i^{(1)}$  and  $\tilde{G}_i = G_i^{(2)}$ ; otherwise, set  $G_i = G_i^{(2)}$  and  $\tilde{G}_i = G_i^{(1)}$ .
  - (e) If  $\text{pop}(V_i) \in [P_i^-, P_i^+]$ , continue on to the next iteration. Otherwise, reject this map and begin the entire sampling process anew.
- 

take  $\tilde{G}_0 = G$  at the first iteration.

## 4.2 The Sampling Probability

The above sequential splitting procedure does not generate plans from the target distribution  $\pi$ . We denote the sampling measure—including the rejection procedure—by  $q$ , and write the sampling probability (for a given connected plan  $\xi$ ) as

$$q(\xi) = q(G_1, G_2, \dots, G_n) = q(G_{n-1} \mid \tilde{G}_{n-2}) \cdots q(G_2 \mid \tilde{G}_1) q(G_1 \mid \tilde{G}_0), \quad (3)$$

where we have used the fact that each new district  $G_i$  depends only on the leftover map area  $\tilde{G}_{i-1}$  from the previous iteration.

The sampling probability at each iteration can be written as the probability that we cut an edge along the boundary of the new district, integrated over all spanning trees which



could be cut to form the district, i.e.,

$$q(G_i \mid \tilde{G}_{i-1}) = \sum_{T \in \mathcal{T}(\tilde{G}_{i-1})} q(G_i \mid T) \tau(\tilde{G}_{i-1})^{-1}, \quad (4)$$

where  $\mathcal{T}(\cdot)$  represents the set of all spanning trees of a given graph, and we have relied on the fact that Wilson's algorithm draws spanning trees uniformly.

The key that allows us to calculate  $q(G_i \mid T)$  is that for certain choices of  $k_i$  (the number of edges considered to be cut at iteration  $i$ ), the probability that an edge is cut is independent of the trees that are drawn. Let  $ok(T)$  represent the number of edges on any spanning tree  $T$  that induce balanced partitions with population deviation below  $D$ , i.e.,

$$ok(T) := |\{e \in E(T) : d_e \leq D\}|.$$

Then define  $K_i := \max_{T \in \mathcal{T}(\tilde{G}_{i-1})} ok(T)$ , the maximum number of such edges across all spanning trees. Furthermore, let  $\mathcal{C}(G, H)$  represent the set of edges joining nodes in a graph  $G$  to nodes in a graph  $H$ . We have the following result (Appendix A for the proof).

**Proposition 1.** *The probability of sampling a connected redistricting plan  $\xi$  induced by  $\{G_i, \tilde{G}_i\}_{i=1}^{n-1}$ , using parameters  $\{k_i\}_{i=1}^{n-1}$  with  $k_i \geq K_i$ , is*

$$q(\xi) \propto \mathbf{1}_{\{\text{dev}(\xi) \leq D\}} \frac{\tau(\xi)}{\tau(G)} \prod_{i=1}^{n-1} k_i^{-1} |\mathcal{C}(G_i, \tilde{G}_i)|.$$

### 4.3 Sequential Importance Sampling

The factored sampling probability in Proposition 1 suggests a sequential importance sampling approach (Doucet et al., 2001; Liu et al., 2001), to generate draws from the target distribution, rather than simply resampling or reweighting after the final stage. A sequential approach is also useful in operationalizing the rejection procedure that is used to enforce the population constraint.

The proposed procedure is governed by  $\alpha \in (0, 1]$ , which has no effect on the target distribution nor the accuracy of the algorithm. Rather,  $\alpha$  is chosen to maximize the efficiency of sampling (see Section 4.4). To generate  $S$  redistricting plans, at each iteration of

---

**Algorithm 2** Sequential Monte Carlo (SMC) Algorithm

---

*Input:* graph  $G$  to be split into  $n$  districts, target distribution parameters  $\rho \in \mathbb{R}_0^+$  and constraint function  $J$ , and sampling parameters  $\alpha \in (0, 1]$  and  $k_i \in \mathbb{Z}^+$ , with  $i \in \{1, 2, \dots, n-1\}$ .

(a) Generate an initial set of  $S$  plans  $\{\tilde{G}_0^{(1)}, \tilde{G}_0^{(2)}, \dots, \tilde{G}_0^{(S)}\}$  and corresponding weights  $\{w_0^{(1)}, w_0^{(2)}, \dots, w_0^{(S)}\}$ , where each  $\tilde{G}_0^{(j)} := G$  and  $w_0^{(j)} = 1$ .

(b) For  $i \in \{1, 2, \dots, n-1\}$ :

(1) Until there are  $S$  valid plans:

(i) Sample a partial plan  $\tilde{G}_{i-1}$  from  $\{\tilde{G}_{i-1}^{(1)}, \tilde{G}_{i-1}^{(2)}, \dots, \tilde{G}_{i-1}^{(S)}\}$  according to weights

$$\left( \prod_{l=1}^{i-1} w_l^{(j)} \right)^\alpha.$$

(ii) Split off a new district from  $\tilde{G}_{i-1}$  through one iteration of the splitting procedure (Algorithm 1), creating a new plan  $(G_i, \tilde{G}_i)$ .

(iii) If the newly sampled plan  $(G_i, \tilde{G}_i)$  satisfies the population bounds, save it; otherwise, reject it.

(2) Calculate weights for each of the new plans

$$w_i^{(j)} = \tau(G_i'^{(j)})^{\rho-1} \frac{k_i}{|\mathcal{C}(G_i'^{(j)}, \tilde{G}_i'^{(j)})|}.$$

(c) Calculate final weights

$$w^{(j)} = \exp\{-J(\xi'^{(j)})\} \left( \prod_{i=1}^{n-1} w_i^{(j)} \right)^{(1-\alpha)} \left( \tau(G_{n-1}^{(j)}) \right)^{\rho-1}. \quad (5)$$

(d) Output the  $S$  final plans  $\{\xi'^{(j)}\}_{j=1}^S$ , where  $\xi'^{(j)} = (G_1^{(j)}, \dots, G_{n-1}^{(j)}, G_n^{(j)})$ , and the final weights  $\{w^{(j)}\}_{j=1}^S$ .

---

the splitting procedure  $i \in \{1, 2, \dots, n-1\}$ , we resample and split the existing plans one at a time, rejecting those which do not meet the population constraints, until we obtain  $S$  new plans for the next iteration. Algorithm 2 details the steps of the full SMC algorithm.

One last resampling of  $S$  plans using the outputted weights can be performed to generate a final sample. Alternatively, the weights can be used directly to estimate the expectation of some statistics of interest, which are functions of redistricting plans, under the target distribution, i.e.,  $H = \mathbb{E}_\pi(h(\xi))$ , where  $\pi$  is given in equation (1), using the self-normalized

importance sampling estimate,  $\hat{H} = \sum_{j=1}^S h(\xi^{(j)})w^{(j)} / \sum_{j=1}^S w^{(j)}$ .

The sampled plans are not completely independent, because the weights in each step must be normalized before resampling, and because the resampling itself introduces some dependence. Precisely quantifying the amount of dependence is difficult. However, as we demonstrate in Section 5, most choices of target distribution  $\pi$  are close to the sampling distribution  $q$ , which means that the weights are not too extreme and hence, the dependence is minimized. And as we show in Section 6, compared with MCMC algorithms, samples generated by the SMC algorithm are more diverse, since there is none of the autocorrelation that is generally found in MCMC algorithms.

The two asymptotically slowest steps of the SMC algorithm are computing  $\tau(G_i)$  for every district  $G_i$  and drawing a spanning tree using Wilson's algorithm for each iteration. All other steps, such as computing  $d_e$  and  $|\mathcal{C}(G_i'^{(j)}, \tilde{G}_i'^{(j)})|$ , are linear in the number of vertices, and are repeated at most once per iteration.<sup>1</sup> Computing  $\tau(G_i)$  requires computing a determinant, which currently has computational complexity  $O(|V_i(\xi)|^{2.373})$  though most implementations are  $O(|V_i(\xi)|^3)$ . Since this must be done for each district of size roughly  $m/n$ , the total complexity is  $O(n \cdot (m/n)^{2.373})$ . For the spanning trees, the expected runtime of Wilson's algorithm is the mean hitting time of the graph, which is  $O(m^2)$  in the worst case. Since we sample a smaller and smaller tree each iteration, the complexity is  $\sum_{i=1}^{n-1} O(\frac{n-i}{n} \cdot m)^2 = O(nm^2)$ . Then, the total complexity for each sample is  $O(nm^2 + m^{2.373}n^{-1.373})$ . Note that when  $\rho = 1$ , we need not compute  $\tau(G_i)$ , and the total complexity is  $O(nm^2)$ .

The weights in the proposed algorithm were chosen so that it will generate valid redistributing plans with probability approximately proportional to the target distribution, up to weight normalization error. This is recorded in the following proposition.

**Proposition 2.** *A set of weighted samples  $\{(\xi^{(j)}, w^{(j)})\}_{j=1}^S$  from Algorithm 2 is proper with*

---

<sup>1</sup>To compute  $d_e$ , we walk depth-first over the tree and store, for each node, the total population of that node and the nodes below it. This allows for  $O(1)$  computation of  $d_e$  for all edges.

respect to the target distribution  $\pi$ , i.e.,

$$\mathbb{E}_q[h(\xi^{(j)})w^{(j)}] = C \mathbb{E}_\pi[h(\xi^{(j)})],$$

for any square integrable function  $h$ , all  $j$ , and  $C$  a common normalizing constant.

Proof is given in Appendix A. As  $S \rightarrow \infty$ , the set of properly weighted samples converges to a set of i.i.d. samples from  $\pi$ .

In some cases, the constraints incorporated into  $J(\xi)$  admit a natural decomposition to the district level as  $\prod_{i=1}^n J'(G_i)$ —for example, a preference for districts which split as few counties as possible, or against districts which would pair off incumbents. In these cases, an extra term of  $\exp\{-J'(G_i^{(j)})\}$  can be added to the weights  $w_i^{(j)}$  in each stage, and the same term can be dropped from the final weights  $w^{(j)}$ . This can be particularly useful for more stringent constraints; incorporating  $J'$  in each stage allows the importance resampling to “steer” the set of redistricting plans towards those which are preferred by the constraints.

As regards the parameter  $\alpha$ , larger values are more aggressive in pruning out unlikely plans (those which are overrepresented in  $q$  versus  $\pi$ ), which may lead to less diversity in the final sample, while smaller values of  $\alpha$  are less aggressive, which can result in more variable final weights and more wasted samples. Liu et al. (2001) recommend a default choice of  $\alpha = 0.5$ ; we find that smaller values (e.g.,  $\alpha = 0.1$ ) work well for some maps.

## 4.4 Implementation Details

The SMC algorithm described above samples from the target distribution given in equation (1), but to practically apply it to non-trivial real-world redistricting problems, there remain two important considerations: the choice of  $k_i$ , since  $K_i$  is typically unknown, and the stabilization of importance weights for efficient sampling when  $\rho \neq 1$ .

### 4.4.1 Choosing $k_i$

The accuracy of the algorithm is theoretically guaranteed only when the number of edges considered for removal at each stage is at least the maximum number of edges across all graphs which induce districts  $G_i$  with  $\text{dev}(G_i) \leq D$ , i.e.,  $k_i \geq K_i$ . Unfortunately,  $K_i$

is almost always unknown in practice. If we set  $k_i = m - 1$ , where  $m$  represents the total number of nodes in the graph, then this condition is certainly satisfied. However, such a choice results in a prohibitively inefficient algorithm—the random edge selected for removal will with high probability induce an invalid partition, leading to a rejection of the entire map. Conversely, if we set  $k_i = 1$ , we gain efficiency by maximizing the chance that the induced districts satisfy the constraint, but lose the theoretical guarantee.

A natural approach is to draw a moderate number of spanning trees  $\mathcal{T}_i \subseteq \mathcal{T}(\tilde{G}_i)$  and compute  $ok(T)$  for each  $T \in \mathcal{T}_i$ . The sample maximum, or the sample maximum plus some small buffer amount, would then be an estimate of the true maximum  $\hat{K}_i$  and an appropriate choice of  $k_i$ . In practice, we find little noticeable loss in algorithmic accuracy even if  $k_i < K_i$ . The following proposition theoretically justifies this finding.

**Proposition 3.** *The probability that an edge  $e$  is selected to be cut at iteration  $i$ , given that the tree  $T$  containing  $e$  has been drawn, and that  $e$  would induce a valid district, satisfies*

$$\max \left\{ 0, q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}) \left( 1 + \frac{1}{k_i} \right) - 1 \right\} \leq q(e = e^* \mid \mathcal{F}) \leq \frac{1}{k_i},$$

where  $\mathcal{F} = \sigma(\{T, \text{pop}(V_i) \in [P_i^-, P_i^+]\})$ .

The proof is deferred to Appendix A. If  $k_i \geq K_i$ , then  $q(e = e^* \mid \mathcal{F})$  is exactly  $k_i^{-1}$ , a fact which is used in the proof of Proposition 1. This result, which is proved using a simple Fréchet bound, shows that as long as  $q(d_e \leq d_{e_{k_i}} \mid \mathcal{F})$  is close to 1, using  $k_i^{-1}$  in Proposition 1 is a good approximation to the true sampling probability.

Having sampled  $\mathcal{T}_i$ , we can compute for each value of  $k$  the sample proportion of trees where a randomly selected edge  $e$  among the top  $k$  of edges of the tree is also among the top  $k$  for the other trees—in effect estimating  $q(d_e \leq d_{e_{k_i}} \mid \mathcal{F})$ . We may then choose  $k_i$  to be the smallest  $k$  for which this proportion exceeds a pre-set threshold (e.g., 0.95). We have found that this procedure, repeated at the beginning of each sampling stage, efficiently selects  $k_i$  without compromising the ability to sample from the target distribution.

#### 4.4.2 Stabilizing importance weights

When  $\rho \neq 1$  or when the constraints imposed by  $J$  are severe, there can be substantial variance in the importance sampling weights. For large maps with  $\rho = 0$ , for instance, since  $\log \tau(\xi) \asymp \text{rem}(\xi)$ , the weights will generally span hundreds if not thousands of orders of magnitude. This reflects the general computational difficulty in sampling uniformly from constrained graph partitions. As [Najt et al. \(2019\)](#) show, sampling of node-balanced graph partitions is computationally intractable in the worst case. In such cases, the importance sampling estimates will be highly variable, and resampling based on these weights may lead to degenerate samples with only one unique map.

When the importance weights are variable but not quite so extreme, we find it useful to truncate the normalized weights (such that their mean is 1) from above at a value  $w_{\max}$  before resampling. The theoretical basis for this maneuver is provided by [Ionides \(2008\)](#), who proved that as long as  $w_{\max} \rightarrow \infty$  and  $w_{\max}/S \rightarrow 0$  as  $S \rightarrow \infty$ , the resulting estimates are consistent and have bounded variance. One such choice we have found to work well for the weights generated by this sampling process is  $w_{\max} = S^{0.4}/100$ , though for particular maps other choices of exponent and constant multiplier may be superior.

Truncation is no panacea, however. As with any method that relies on importance sampling, it is critical to examine the distribution of importance weights to ensure that they will yield acceptable resamples.

### 4.5 Incorporating Administrative Boundary Constraints

Another common requirement for redistricting plans is that districts “to the greatest extent possible” follow existing administrative boundaries such as county and municipality lines.<sup>2</sup> In theory, this constraint can be formulated using a  $J$  function which penalizes maps for every county line crossed by a district. In practice, however, we can more efficiently generate

---

<sup>2</sup>If a redistricting plan must always respect these boundaries, we can simply treat the administrative units as the nodes of the original graph to be partitioned.

desired maps by directly incorporating this constraint into our sampling algorithm.

Fortunately, with a small modification to the proposed algorithm, we can sample restricting plans proportional to a similar target distribution but with the additional constraint that the number of administrative splits not exceed  $n - 1$ . In most states, the number of administrative units that are considered (such as counties) is much larger than the number of districts  $n$ , so this constraint represents a significant improvement from the baseline algorithm, which can in theory yield up to  $n - 1$  splits. Together with this constraint, a further preference for fewer administrative splits can be incorporated through the  $J$  function.

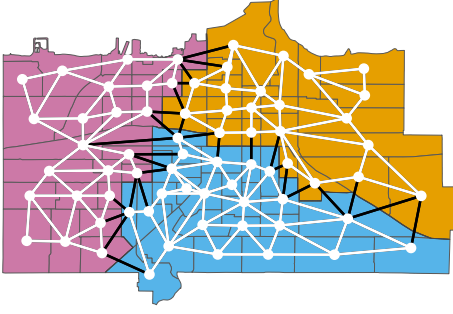
Let  $A$  be the set of administrative units, such as counties. We can relate these units to the nodes by way of a labeling function  $\eta : V \rightarrow A$  that assigns each node to its corresponding unit. Thus, our modified algorithm works for non-administrative units so long as they can be represented by this labeling function. This function induces an equivalence relation  $\sim_\eta$  on nodes, where  $v \sim_\eta u$  for nodes  $v$  and  $u$  iff  $\eta(v) = \eta(u)$ . If we quotient  $G$  by this relation, we obtain the administrative-level multigraph  $G / \sim_\eta$ , where each vertex is an administrative unit and every edge corresponds to an edge in  $G$  which connects two nodes in different administrative units. We can write the number of administrative splits as

$$\text{spl}(\xi) = \left( \sum_{a \in A} \sum_{i=1}^n C(\eta^{-1}(a) \cap \xi^{-1}(i)) \right) - |A|,$$

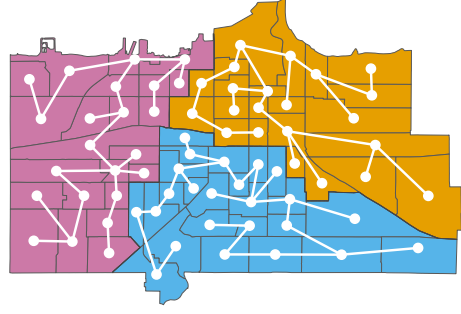
where  $C(\cdot)$  counts the number of connected components in the subgraph  $\eta^{-1}(a) \cap \xi^{-1}(i)$ .

To implement this constraint, we draw the spanning trees in step (a) of the algorithm in two substeps such that we sample from a specific subset of all spanning trees. First, we use Wilson's algorithm to draw a spanning tree on each administrative unit  $a \in A$ , and then we connect these spanning trees to each other by drawing a spanning tree on the quotient multigraph  $\tilde{G}_i / \sim_\eta$ . Figure 4 illustrates this process. This approach is similar to the independently-developed multi-scale merge-split algorithm of [Autry et al. \(2020\)](#).

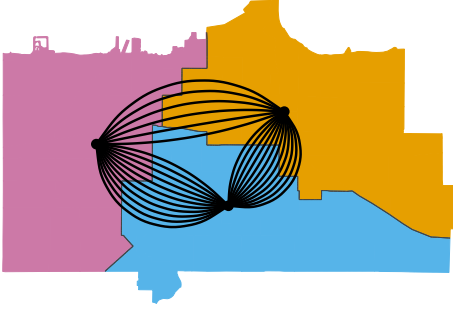
Drawing the spanning trees in two steps limits the trees used to those which, when restricted to the nodes  $\eta^{-1}(a)$  in any administrative unit  $a$ , are still spanning trees. The



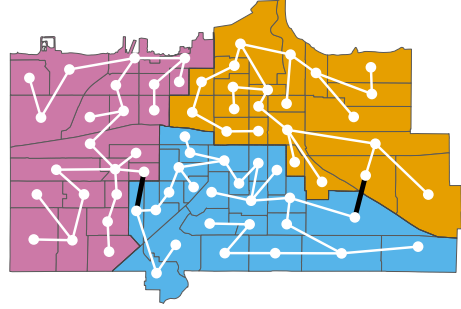
(a) The Erie graph. Edges that cross from one administrative unit to another are colored black.



(b) Spanning trees drawn on each administrative unit.



(c) The quotient multigraph. The number of edges connecting each node is the number of edges that connect each unit in the original graph. We hide self-loops as they are not included in spanning trees.



(d) The final spanning tree, with a spanning tree on the quotient multigraph used to connect the spanning trees on each administrative unit. Notice that removing any edge splits at most one unit.

Figure 4: The two-step spanning tree sampling procedure applied to the city of Erie, Pennsylvania, with three arbitrary administrative units indicated by the colored sections of each map.

importance of this restriction is that cutting any edge in such a tree will either split the map exactly along administrative boundaries (if the edge is on the quotient multigraph) or split one administrative unit in two and preserve administrative boundaries everywhere else. Since the algorithm has  $n - 1$  stages, this limits the support of the sampling distribution to maps with no more than  $n - 1$  administrative splits.

This modification does not make theoretical analysis intractable. Indeed, the two-step construction makes clear that the total number of such spanning trees is given by

$$\tau_\eta(\tilde{G}_i) = \tau(\tilde{G}_i / \sim_\eta) \prod_{a \in A} \tau(\tilde{G}_i \cap \eta^{-1}(a)), \quad (6)$$

where  $\tilde{G}_i \cap \eta^{-1}(a)$  denotes the subgraph of  $\tilde{G}_i$  which lies in unit  $a$ , and we take  $\tau(\emptyset) = 1$ .

Replacing  $\tau$  with  $\tau_\eta$  in the expression for the weights  $w_i^{(j)}$  and  $w^{(j)}$  then gives the modified



algorithm that generates a properly weighted sample from

$$\pi_{\eta}(\xi) \propto \exp\{-J(\xi)\} \tau_{\eta}(\xi)^{\rho} \mathbf{1}_{\{\xi \text{ connected}\}} \mathbf{1}_{\{\text{dev}(\xi) \leq D\}} \mathbf{1}_{\{\text{spl}(\xi) \leq n-1\}}. \quad (7)$$

This idea can in fact be extended to arbitrary levels of nested administrative hierarchy. We can, for example, limit not only the number of split counties but also the number of split cities and Census tracts to  $n - 1$  each, since tracts are nested within cities, which are nested within counties. To do so, we begin by drawing spanning trees using Wilson’s algorithm on the smallest administrative units. We then connect spanning trees into larger and larger trees by drawing spanning trees on the quotient graphs of each higher administrative level. Even with multiple levels of administrative hierarchy, the calculation of the number of spanning trees is still straightforward, by analogy to equation (6).

## 5 An Empirical Validation Study

Although the proposed algorithm has desirable theoretical properties, it is important to empirically assess its performance (Fifield et al., 2020b). We examine whether or not the proposed algorithm is able to produce a sample of redistricting maps that is actually representative of a target distribution. We use a 50-precinct map taken from the state of Florida, and use the efficient enumeration algorithm of Fifield et al. (2020b) to obtain all possible redistricting maps with three contiguous districts. While there are over 4.2 million (4,266,875 to be exact) partitions, only a small number satisfy realistic population and compactness constraints. We demonstrate that the proposed algorithm can efficiently approximate several target distributions on this set under different sets of constraints. Appendix B contains an additional validation study in which this same map is split into four districts.

The left plot of Figure 5 shows this validation map, which we have divided into four arbitrary administrative units. We sample from three different target measures on the validation map, and compare the samples to the true reference distribution based on the enumeration. When there are only a handful of valid partitions, we directly compare the sample

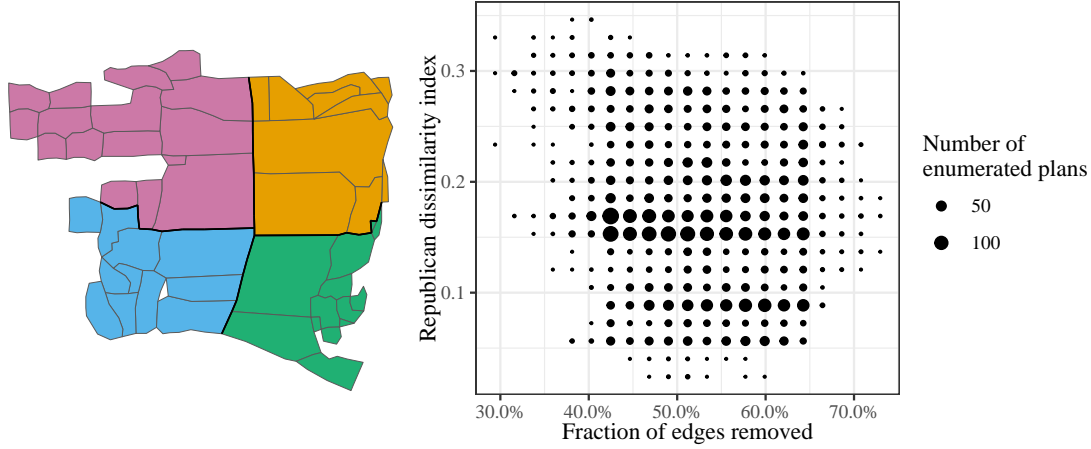


Figure 5: The 50-precinct Florida map used for validation, arbitrarily divided into four administrative units (left), and the joint distribution of Republican dissimilarity and compactness on the map over all partitions into three districts with  $\text{dev}(\xi) \leq 0.05$  (right).

frequency for each partition to the desired uniform distribution. When there are too many partitions to make these individual comparisons, we compare the samples to the reference enumeration by using the Republican dissimilarity index (Massey and Denton, 1988), a commonly-used measure of spatial segregation. The right plot of the figure shows that with this validation map, the dissimilarity index is particularly sensitive to the compactness of districts. This makes the Republican dissimilarity index a good test statistic for comparing distributions that differ primarily in their average compactness.

We also compare the accuracy of the SMC algorithm to that of the “merge-split” spanning tree-based MCMC algorithm (Carter et al., 2019),<sup>3</sup> which uses a spanning tree-based proposal similar to the splitting procedure described in Algorithm 1: it merges adjacent districts, draws a spanning tree on the merged district, and splits it to ensure the population constraint is met. Although the parametrization is different, the stationary distribution of this algorithm is exactly that of equation (1). The merge-split algorithm can also incorporate additional constraints into its Metropolis step, but we do not include any here.

It is difficult to directly compare SMC and MCMC algorithms when run for the same

<sup>3</sup>We use their open-source implementation at <https://git.math.duke.edu/gitlab/gjh/mergesplitcodebase> (accessed July 30, 2020).

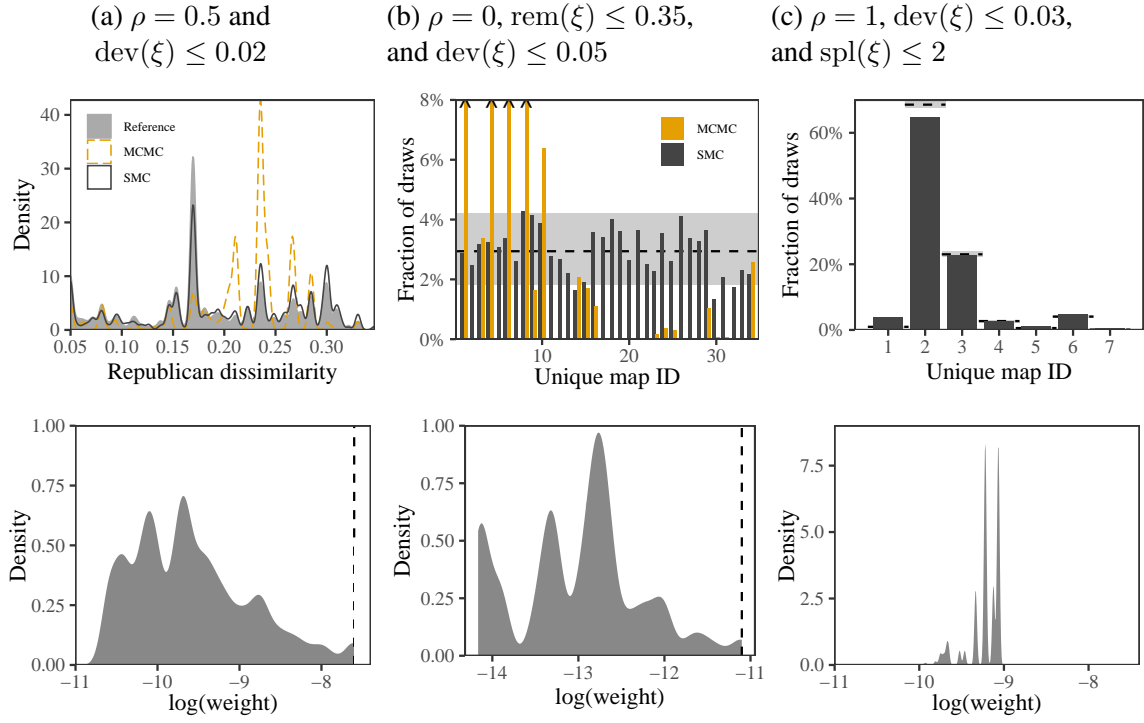


Figure 6: Calibration plots for Republican dissimilarity under various target measures. For target measure (a), density estimates of the the algorithm output and the reweighted enumeration are plotted. For target measures (b) and (c), the sample frequency of individual maps is plotted in gray, with the horizontal line indicating the target frequency and the shaded area indicating the expected range of random variation given the number of effective samples. The distribution of truncated importance weights for each sample is shown below each plot, with the truncation value marked with a vertical line. For target measures (a) and (b), the output of the merge-split MCMC algorithm is also plotted in orange, with values exceeding plot bounds marked with a caret (^).

number of iterations. SMC samples require no burn-in period, while MCMC samples are generally autocorrelated and require convergence monitoring. The comparisons here are intended to highlight the performance of the two algorithms when run for a moderate but reasonable number of iterations. They are not meant to establish the maximum achievable performance of either algorithm when run under optimal settings, since the MCMC algorithm has been shown is known to converge, at least theoretically, with enough samples.

First, we target a moderately constrained target distribution by choosing  $\rho = 0.5$  in equation (1) and setting the population constraint to  $\text{dev}(\xi) \leq 0.02$ . There are only 814 partitions (or 0.019% of all maps) that satisfy this population constraint in the reference enumeration. Since the target distribution is not uniform, we reweight the enumerated

maps according to  $\tau(\xi)^{0.5}$ . We sampled 10,000 plans and reweighted them according to the importance weights, using a normalized weight truncation of  $w_{\max} = 0.05 \times \sqrt{10,000}$  (see the lower panel of Figure 6a for the distribution of weights after truncation). We ran the MCMC merge-split algorithm for 20,000 iterations, and discarded the first 10,000 samples.

The upper panel of Figure 6a shows the resulting density estimates. While the target distribution is highly multimodal, there is good agreement between the SMC sample and reference distribution. In contrast, the MCMC samples fail to accurately capture the left tail of the distribution, and significantly oversample certain values in the right tail.

Second, we target a uniform distribution on the set of maps with  $\text{dev}(\xi) \leq 0.05$  and  $\text{rem}(\xi) \leq 0.35$ . Note that the median fraction of removed edges across all partitions under 5% population deviation was 53%. There are a total of only 34 maps (or 0.0008% of all maps) that satisfy these two constraints. As before, we sampled 10,000 plans truncate to the same value of  $w_{\max}$  (see the lower panel of Figure 6b for the distribution of truncated weights). We discarded any samples which did not meet the compactness constraint, leaving 2,995 SMC samples and 1,364 MCMC samples.

The upper panel of the figure shows the results. Since there are only 34 maps, we can individually identify each and plot the observed map frequencies versus the expected frequency of  $1/34$ . While the SMC samples do not perfectly approximate the target distribution, the variation in sample frequencies is generally within the range that would be expected due to binomial variation with the number of effective samples obtained here (indicated by a grey band). In comparison, the MCMC algorithm was not able to sample accurately from this target distribution in 20,000 iterations.

Finally, to demonstrate the administrative boundary constraint, we sample from the distribution with  $\rho = 1$ ,  $\text{dev}(\xi) \leq 0.03$ , and  $\text{spl}(\xi) \leq 2$ , using the arbitrary administrative boundaries shown in the left plot of Figure 5. The combination of these constraints is extremely strong, allowing only at most two county splits. Indeed, only 7 partitions (or

approximately 0.00016%) satisfy them all. Since the merge-split MCMC algorithm is not specifically designed to enforce this hard constraint, we do not present its results. We sample 10,000 plans using the modified SMC algorithm of Section 4.5. We do not truncate the weights since with  $\rho = 1$  their variance is small, as shown in the lower panel of Figure 6c. As in the first validation exercise, the target measure is not uniform, and the upper panel of Figure 6c plots the sample frequencies of the 7 maps versus their density under the target measure. Despite the severe constraints, the proposed algorithm continues to perform well, although map 1 is oversampled and map 2 is slightly undersampled.

## 6 Analysis of the 2011 Pennsylvania Redistricting

As discussed in Section 2, in the process of determining a remedial redistricting plan to replace the 2011 General Assembly map, the Pennsylvania Supreme Court received submissions from seven parties. In this section, we compare four of these maps to both the original 2011 plan and the remedial plan ultimately adopted by the court. We study the governor’s plan and the House Democrats’ plan; the petitioner’s plan (specifically, their “Map A”), which was selected from an ensemble of 500 plans used as part of the litigation; and the respondent’s plan, which was drawn by Republican officials.

### 6.1 The Setup

To evaluate these six plans, we drew 1,500 reference maps from the target distribution given in equation (7) by using the proposed algorithm along with the modifications presented in Section 4.5 to constrain the number of split counties to 17 (out of a total of 67), in line with the court’s mandate. We set  $\rho = 1$  to put most of the sample’s mass on compact districts, and enforced  $\text{dev}(\xi) \leq 0.001$  to reflect the “one person, one vote” requirement. The parameters  $k_i$  were selected according to the automated procedures laid out in Section 4.4.1, with a threshold value of 0.95. The rejection rate at each iteration averaged 15.1%.

This population constraint translates to a tolerance of around 700 people, in a state

where the median precinct has 1,121 people. Like most research on redistricting, we use precincts because they represent the smallest geographical units for which election results are available. To draw from a stricter population constraint we would need to use the 421,545 Census blocks in Pennsylvania rather than the 9,256 precincts, which would significantly increase the computational burden.

## 6.2 Comparison with a State-of-the-Art MCMC Algorithm

We first compare the computational efficiency of the SMC algorithm with the merge-split MCMC algorithm used in the empirical validation above.<sup>4</sup> To make the comparison, we use the merge-split algorithm to draw 1,500 Pennsylvania redistricting maps of 18 districts with  $\text{dev}(\xi) \leq 0.001$  and  $\rho = 1$  (in our parametrization), the same settings as were used to generate the SMC samples. However, the MCMC algorithm does not enforce a hard constraint on the number of county splits. While these preferences can be encoded in an energy function, here, for the sake of comparison, we did not enforce any additional constraints. Since additional constraints generally lead to substantially less efficiency in MCMC settings, we do not expect this simplification to affect our qualitative findings.

To calculate the efficiency of the MCMC algorithm, we use the standard autocorrelation-based formula (see, e.g. [Geyer, 2011](#)), using the Republican dissimilarity index described above as the summary statistic. For the SMC algorithm, we use the effective sample calculation for functions of interest given in [Owen \(2013\)](#), which uses the distribution of importance sampling weights. However, to avoid the complications that arise from importance resampling being performed at every step of the SMC algorithm, for this comparison only we perform no resampling (i.e., we set  $\alpha = 0$ ) in between iterations of the algorithm, performing only one final resampling after the algorithm has terminated. Since this deprives

---

<sup>4</sup>We do not make a comparison to the recombination (“ReCom”) algorithm of [DeFord et al. \(2019\)](#), which pioneered the spanning tree-based proposal used in the merge-split algorithm, as it does not sample from a specific target distribution.

	Republican dissimilarity		Unique plans	
	SMC	Merge-split	SMC	Merge-split
Nominal Samples	1500	1500	1500	1500
Effective samples	235.6	5.6	467	86
Efficiency	15.7%	0.4%	31.3%	5.7%

Table 1: Comparison of efficiency of the proposed Sequential Monte Carlo (SMC) algorithm and the merge-split MCMC algorithm with spanning tree-based proposals.

the algorithm of the ability to “prune” bad samples, the effective sample size calculated here should be considered a conservative lower bound.

We also count the number of unique redistricting plans generated by each algorithm, as another measure of sampling efficiency that is agnostic to the type of algorithm used.

The results are shown in Table 1. The SMC algorithm is 42 times more efficient than the MCMC algorithm, according to the usual effective sample calculations. And the SMC algorithm generates 5.4 times more unique samples in the same number of iterations.

Another way to view the algorithmic efficiency is by measuring how different the samples are from each other, using the variation of information metric shown in equation (2). Similar plans will have a variation of information near zero, while plans which are extremely different will have a variation of information near the maximum value of 1. Figure 7 shows the distribution of the pair-wise variation of information distance for each algorithm. The distance between most pairs of SMC sample plans is greater than the maximum distance between any pairs of MCMC plans, due to the autocorrelation inherent in Markov chain methods.

We cannot directly compare these algorithms in terms of their runtime since that depends on the specific implementation of each algorithm. An exact theoretical comparison is also difficult. Although the computational complexity for a sample from the SMC algorithm is  $O(nm^2)$  (see Section 4.3), while the complexity of the MCMC proposals is  $O(m^2)$ , the MCMC algorithm only changes two districts at a time, whereas an SMC sample redraws all  $n$  districts. For this particular application, the specific implementations

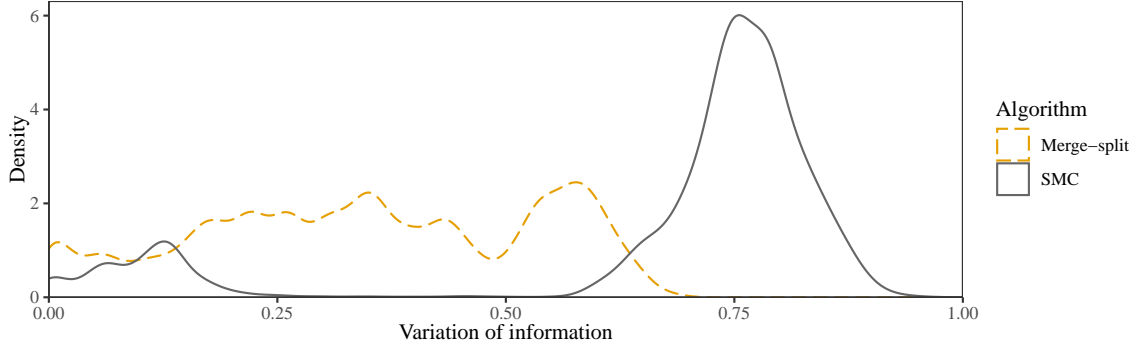


Figure 7: The distribution of the variation of information between all pairs of samples from the SMC and two MCMC algorithms.

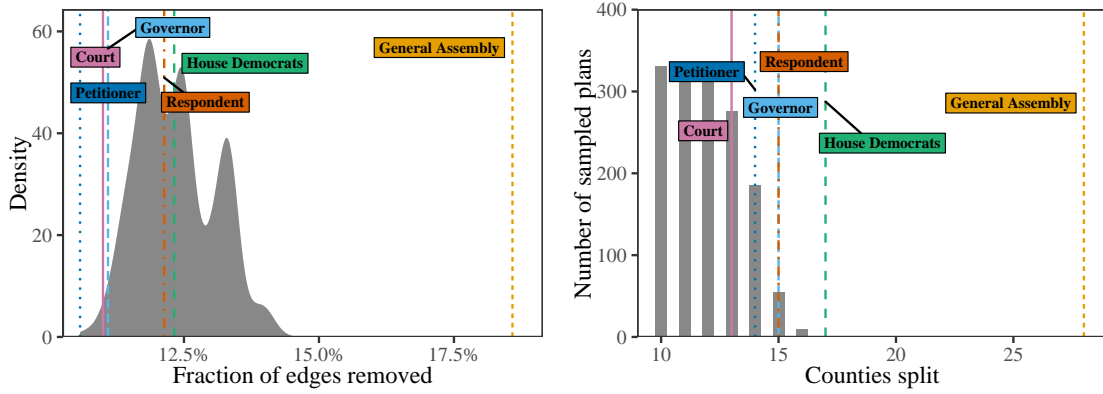


Figure 8: Summary statistics for the six plans, compared to their distribution under the target measure. The left plot shows  $\text{rem}(\xi)$ , where smaller values indicate more compact districts. The right plot shows  $\text{spl}(\xi)$ , whose median value under the target measure is 12.

of the the SMC and merge-split MCMC algorithms we used took 52 and 11 minutes to sample, respectively. This implies that SMC is several times more effective than the state-of-the-art MCMC algorithm in terms of runtime per effective sample. Although additional study is warranted, our results suggest that the proposed algorithm may be substantially more efficient when applied to real-world redistricting problems.

### 6.3 Compactness and County Splits

Figure 8 shows distribution of the fraction of edges removed ( $\text{rem}(\xi)$ ) and the number of county splits ( $\text{spl}(\xi)$ ) across the reference maps generated by our algorithm (grey histograms). The figure also plots these values for each of the six plans using vertical lines of various types. The 2011 General Assembly plan is a clear outlier for both statistics,



being far less compact and splitting far more counties than any of the reference plans and all of the remedial plans. Among the remedial plans, the petitioner’s is the most compact, followed by the court’s and the governor’s, according to the  $\text{rem}(\xi)$  statistic. The House Democratic plan and the respondent’s plan were the least compact based on both statistics, though still well within the normal range, according to the reference maps. In fact, the petitioner’s plan appears to be an outlier in being *too* compact, although this is perhaps not surprising—the map was generated by an algorithm explicitly designed to optimize over criteria such as population balance and compactness (Chen, 2017).

The right plot of Figure 8 shows that all of the submitted plans split between 13 and 17 counties, with the court’s adopted plan splitting the fewest. Yet around half of the reference maps split fewer than 13 counties, with some splitting only 10. This may be a result of the strict population constraint (all six plans were within 1 person of equal population across all districts), or a different prioritization between the various constraints imposed.

## 6.4 Partisan Analysis

While important, the outlier status of the General Assembly plan as regards compactness and county splits is not sufficient to show that it is a *partisan* gerrymander. To evaluate the partisan implications of the six plans, we take a precinct-level baseline voting pattern and aggregate it by district to explore hypothetical election outcomes under the six plans and the reference maps. The baseline pattern is calculated by averaging the vote totals for the three presidential elections and three gubernatorial elections that were held in Pennsylvania from 2000 to 2010 (Ansolabehere and Rodden, 2011). These election data were also used during litigation. While being far from a perfect way to create counterfactual election outcomes, this simple averaging of statewide results is often used in academic research and courts.

We first compute the “gerrymandering index” proposed by Bangia et al. (2017) for the six plans as well as the sampled reference maps. Within each plan, we number the districts by their baseline Democratic vote share, so District 1 is the least Democratic and District 18

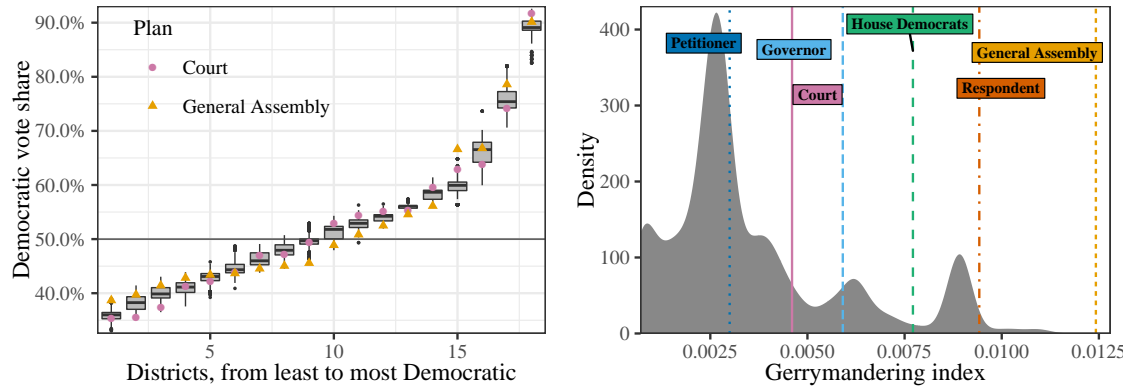


Figure 9: Democratic vote share by districts (left plot), where within each plan districts are ordered by Democratic vote share, and the derived gerrymandering index (right plot), which is computed as the total squared deviation from the mean district-vote share relationship in the left plot.

the most. The left plot of Figure 9, analogous to Figure 7 in [Herschlag et al. \(2017\)](#), presents the distribution of the Democratic vote share for each of the districts across the reference maps, and also shows the values for the General Assembly plan (orange triangles) and the court's adopted plan (purple circles). When compared to the reference maps and the court's plan, the General Assembly plan tends to yield smaller Democratic vote share in competitive districts while giving larger Democratic vote share in non-competitive districts. This finding is consistent with the view that the General Assembly plan is gerrymandered in favor of Republicans by packing Democratic voters in non-competitive districts.

From this district-vote share relationship, we can compute the gerrymandering index by summing the squared deviations from the mean vote share in each district. The right plot of Figure 9 shows the distribution of this index based on the reference maps and indicates the values of the six maps. By this metric, the General Assembly plan is a clear outlier, as are the respondent's plan and the House Democrats' plan. The petitioner's plan has the smallest gerrymandering index among the six studied plans, while the plans adopted by the Governor and the court are within the normal range, according to the reference maps.

While the gerrymandering index is a useful summary of the district-vote share relationship, it weights all deviations equally and does not consider their direction. To address

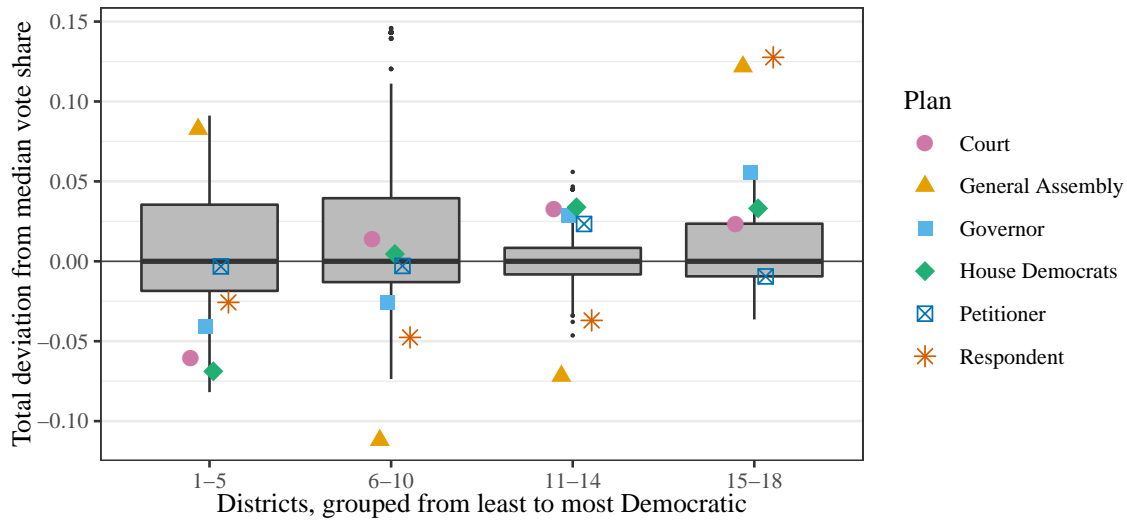


Figure 10: The eighteen districts are put into four groups depending on their Democratic vote share, and the total deviation from the median vote share of each group is plotted for each plan and the reference ensemble. The points are horizontally jittered to improve the visualization without altering their values. Gerrymandering is visible as a pattern of cracking voters in the middle two groups and packing them into the outer groups, diluting their voting power in competitive districts.

this, we group the districts and sum the deviations from median vote share for each district within the group. Positive deviations within a group indicate that voters in these districts tilt more Democratic than would otherwise be expected, while negative deviations indicate the same for Republicans.

Figure 10 shows the results. The General Assembly plan has outlier deviations for all four groups, clearly packing Democratic voters into safe Republican (1–5) and safe Democratic (15–18) districts, while cracking them and diluting their vote shares in the competitive districts (6–10 and 11–14). The respondent’s remedial plan, while not as extreme, maintains the packing in Districts 15–18 and cracking in 6–14. In contrast, the House Democrats’ plan tries the opposite tack, cracking Republican voters Districts 11–14 and packing them into the heavily Republican Districts 1–5. Intriguingly, the court’s adopted plan has a similar pattern to the House Democrats’ plan, while the petitioner’s plan appears to be the most balanced. This may explain the general surprise expressed in the media that Democrats unexpectedly benefited from the court’s new plan (see, e.g. [Cohn, 2018](#)).

## 7 Concluding Remarks

Redistricting sampling algorithms allow for the empirical evaluation of a redistricting plan by generating alternative plans under a certain set of constraints. Researchers and policy-makers can compute various statistics from the redistricting plan of interest and compare them with the corresponding statistics based on these sampled plans. Unfortunately, existing approaches often struggle when applied to real-world problems, owing to the scale of the problems and the number of constraints involved.

The SMC algorithm presented here is able to sample from a specific target distribution, but does not face the same kinds of scalability problems as many existing MCMC algorithms. It also incorporates, by design, the common redistricting constraints of population balance, geographic compactness, and minimizing administrative splits. Additionally, the algorithm’s direct sampling leads to increased efficiency compared to highly dependent Markov chains and greatly reduces concerns about mixing and the need to monitor convergence. We expect these advantages of the proposed SMC algorithm to substantially improve the reliability of outlier analysis in real-world redistricting cases.

Future research should explore the possibility of improving several design choices in the algorithm to further increase its efficiency. Wilson’s algorithm, for instance, can be generalized to sample from edge-weighted graphs. Choosing weights appropriately could lead to trees which induce maps that are more balanced or more compact. And the procedure for choosing edges to cut, while allowing for the sampling probability to be calculated, introduces inefficiencies by leading to many rejected maps. Further improvements in either of these areas should allow us to better sample and investigate redistricting plans over large maps and with even more complex sets of constraints.

## References

- Ansolabehere, S. and Rodden, J. (2011). Pennsylvania Data Files.
- Autry, E., Carter, D., Herschlag, G., Hunter, Z., and Mattingly, J. (2020). Multi-scale merge-split Markov chain Monte Carlo for redistricting. *Working paper*.
- Bangia, S., Graves, C. V., Herschlag, G., Kang, H. S., Luo, J., Mattingly, J. C., and Ravier, R. (2017). Redistricting: Drawing the line. *arXiv preprint arXiv:1704.03360*.
- Bozkaya, B., Erkut, E., and Laporte, G. (2003). A tabu search heuristic and adaptive memory procedure for political districting. *European journal of operational research*, 144(1):12–26.
- Carter, D., Herschlag, G., Hunter, Z., and Mattingly, J. (2019). A merge-split proposal for reversible Monte Carlo Markov chain sampling of redistricting plans. *arXiv preprint arXiv:1911.01503*.
- Chen, J. (2017). Expert report of Jowei Chen, Ph.D. Expert witness report in League of Women Voters v. Commonwealth.
- Chen, J. and Rodden, J. (2013). Unintentional gerrymandering: Political geography and electoral bias in legislatures. *Quarterly Journal of Political Science*, 8(3):239–269.
- Chikina, M., Frieze, A., and Pegden, W. (2017). Assessing significance in a Markov chain without mixing. *Proceedings of the National Academy of Sciences*, 114(11):2860–2864.
- Chikina, M., Frieze, A., and Pegden, W. (2019). Understanding our Markov Chain significance test: A reply to Cho and Rubinstein-Salzedo. *Statistics and Public Policy*, 6(1):50–53.

- Cho, W. K. T. and Liu, Y. Y. (2018). Sampling from complicated and unknown distributions: Monte Carlo and Markov chain Monte Carlo methods for redistricting. *Physica A: Statistical Mechanics and its Applications*, 506:170–178.
- Cho, W. K. T. and Rubinstein-Salzedo, S. (2019). Understanding significance tests from a non-mixing Markov Chain for partisan gerrymandering claims. *Statistics and Public Policy*, 6(1):44–49.
- Cirincione, C., Darling, T. A., and O’Rourke, T. G. (2000). Assessing South Carolina’s 1990s congressional districting. *Political Geography*, 19(2):189–211.
- Cohn, N. (2018). Democrats didn’t even dream of this Pennsylvania map. How did it happen? *The New York Times*.
- Cover, T. M. and Thomas, J. A. (2006). *Elements of information theory*. John Wiley & Sons, 2 edition.
- DeFord, D., Duchin, M., and Solomon, J. (2019). Recombination: A family of Markov chains for redistricting. *arXiv preprint arXiv:1911.05725*.
- Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo methods in practice*. Springer, New York.
- Dube, M. P. and Clark, J. T. (2016). Beyond the circle: Measuring district compactness using graph theory. In *Annual Meeting of the Northeastern Political Science Association*.
- Duchin, M. (2018). Outlier analysis for pennsylvania congressional redistricting.
- Fifield, B., Higgins, M., Imai, K., and Tarr, A. (2020a). Automated redistricting simulation using Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics*, page Forthcoming.

- Fifield, B., Imai, K., Kawahara, J., and Kenny, C. T. (2020b). The essential role of empirical validation in legislative redistricting simulation. *Statistics and Public Policy*, 7(1):52–68.
- Fifield, B., Kenny, C. T., McCartan, C., Tarr, A., and Imai, K. (2020c). redist: Computational algorithms for redistricting simulation. <https://CRAN.R-project.org/package=redist>.
- Geyer, C. (2011). Introduction to Markov chain Monte Carlo. In *Handbook of Markov chain Monte Carlo*, pages 3–48. CRC Press, Boca Raton.
- Herschlag, G., Ravier, R., and Mattingly, J. C. (2017). Evaluating partisan gerrymandering in Wisconsin. *arXiv preprint arXiv:1709.01596*.
- Ionides, E. L. (2008). Truncated importance sampling. *Journal of Computational and Graphical Statistics*, 17(2):295–311.
- Kaufman, A., King, G., and Komisarchik, M. (2020). How to measure legislative district compactness if you only know it when you see it. *American Journal of Political Science*, page Forthcoming.
- Kostochka, A. V. (1995). The number of spanning trees in graphs with a given degree sequence. *Random Structures & Algorithms*, 6(2-3):269–274.
- Liu, J. S., Chen, R., and Logvinenko, T. (2001). A theoretical framework for sequential importance sampling with resampling. In *Sequential Monte Carlo methods in practice*, pages 225–246. Springer.
- Liu, Y. Y., Cho, W. K. T., and Wang, S. (2016). PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis. *Swarm and Evolutionary Computation*, 30:78–92.
- Macmillan, W. (2001). Redistricting in a GIS environment: An optimisation algorithm using switching-points. *Journal of Geographical Systems*, 3(2):167–180.

- Magleby, D. B. and Mosesson, D. B. (2018). A new approach for developing neutral redistricting plans. *Political Analysis*, 26(2):147–167.
- Massey, D. S. and Denton, N. A. (1988). The dimensions of residential segregation. *Social forces*, 67(2):281–315.
- Mattingly, J. C. and Vaughn, C. (2014). Redistricting and the will of the people. *arXiv preprint arXiv:1410.8796*.
- McKay, B. D. (1981). Spanning trees in random regular graphs. In *Proceedings of the Third Caribbean Conference on Combinatorics and Computing*. Citeseer.
- Mehrotra, A., Johnson, E. L., and Nemhauser, G. L. (1998). An optimization based heuristic for political districting. *Management Science*, 44(8):1100–1114.
- Najt, L., Deford, D., and Solomon, J. (2019). Complexity and geometry of sampling connected graph partitions. *arXiv preprint arXiv:1908.08881*.
- Owen, A. B. (2013). *Monte Carlo theory, methods and examples*, chapter 9, page 12.
- Polsby, D. D. and Popper, R. D. (1991). The third criterion: Compactness as a procedural safeguard against partisan gerrymandering. *Yale Law & Policy Review*, 9(2):301–353.
- Tutte, W. T. (1984). *Graph Theory*. Addison-Wesley.
- 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*, pages 296–303.
- Wu, L. C., Dou, J. X., Sleator, D., Frieze, A., and Miller, D. (2015). Impartial redistricting: A Markov Chain approach. *arXiv preprint arXiv:1510.03247*.



## A Proofs of Propositions

**Proposition 1.** *The probability of sampling a connected redistricting plan  $\xi$  induced by  $\{G_i, \tilde{G}_i\}_{i=1}^{n-1}$ , using parameters  $\{k_i\}_{i=1}^{n-1}$  with  $k_i \geq K_i$ , is*

$$q(\xi) \propto \mathbf{1}_{\{\text{dev}(\xi) \leq D\}} \frac{\tau(\xi)}{\tau(G)} \prod_{i=1}^{n-1} k_i^{-1} |\mathcal{C}(G_i, \tilde{G}_i)|.$$

*Proof.* We have factored the overall sampling probability in equation (3) and written the sampling probability at each iteration in equation (4).

Any spanning tree can be decomposed into two other trees and an edge joining them. Let  $T \cup e \cup T'$  denote the spanning tree obtained by joining two other spanning trees,  $T$  and  $T'$ , with an edge  $e$ . Then equation (4) can be written as

$$q(G_i \mid \tilde{G}_{i-1}) = \sum_{\substack{T^{(1)} \in \mathcal{T}(G_i) \\ T^{(2)} \in \mathcal{T}(\tilde{G}_i)}} \sum_{e \in \mathcal{C}(T^{(1)}, T^{(2)})} q(G_i \mid T^{(1)} \cup e \cup T^{(2)}) \tau(\tilde{G}_{i-1})^{-1}.$$

Now,  $q(G_i \mid T^{(1)} \cup e \cup T^{(2)})$  is determined by two factors: whether  $e^* = e$ , i.e., if  $e$  is the edge selected to be cut, and whether  $\text{pop}(V_i) \in [P_i^-, P_i^+]$ . If  $\text{pop}(V_i) \notin [P_i^-, P_i^+]$  then the map is rejected and the probability is zero. Therefore

$$q(G_i \mid T^{(1)} \cup e \cup T^{(2)}) \propto \mathbf{1}_{\{\text{pop}(V_i) \in [P_i^-, P_i^+]\}} q(e^* = e \mid T^{(1)} \cup e \cup T^{(2)}, \text{pop}(V_i) \in [P_i^-, P_i^+]).$$

What determines if  $e^* = e$ ? If  $e$  has  $d_e$  in the top  $k_i$ , then it has a  $1/k_i$  probability of being selected in step (c) and cut. If  $d_e$  is not in the top  $k_i$ , then this probability is zero.

In addition, notice that the forward-looking bounds  $P_i^-$  and  $P_i^+$  are stricter than merely ensuring  $\text{dev}(G_i) \leq D$ . That is, conditional on  $\text{pop}(V_i) \in [P_i^-, P_i^+]$ , we must have  $\text{dev}(G_i) \leq D$ .

Therefore, if a sorted edge  $e_j$  in any spanning tree induces such a balanced partition, we must have  $j \leq K_i$ , where as above  $K_i$  counts the maximum number of such edges across all possible spanning trees. Thus, so long as we set  $k_i \geq K_i$ , we will have  $d_e \leq D$ .

Furthermore, across all spanning trees  $T^{(1)} \in \mathcal{T}(G_i)$  and  $T^{(2)} \in \mathcal{T}(\tilde{G}_i)$ , and connecting edges  $e \in E(T^{(1)}, T^{(2)})$ , the value of  $d_e$  is constant, since removing  $e$  induces the same

districting. Combining these two facts, we have

$$q(e^* = e \mid T^{(1)} \cup e \cup T^{(2)}, \text{pop}(V_i) \in [P_i^-, P_i^+]) = k_i^{-1},$$

which does not depend on  $T^{(1)}$ ,  $T^{(2)}$ , or  $e$ . We may therefore write the sampling probability as

$$\begin{aligned} q(G_i \mid \tilde{G}_{i-1}) &\propto \sum_{\substack{T^{(1)} \in \mathcal{T}(G_i) \\ T^{(2)} \in \mathcal{T}(\tilde{G}_i)}} \sum_{e \in \mathcal{C}(T^{(1)}, T^{(2)})} \frac{\mathbf{1}_{\{\text{pop}(V_i) \in [P_i^-, P_i^+]\}}}{k_i \tau(\tilde{G}_{i-1})} \\ &= \frac{\tau(G_i) \tau(\tilde{G}_i)}{\tau(\tilde{G}_{i-1}) k_i} |\mathcal{C}(G_i, \tilde{G}_i)| \mathbf{1}_{\{\text{pop}(V_i) \in [P_i^-, P_i^+]\}}, \end{aligned} \quad (8)$$

where as in the main text we let  $\mathcal{C}(G, H)$  represent the set of edges joining nodes in a graph  $G$  to nodes in a graph  $H$ . Substituting equation (8) into equation (3), the factored sampling probability telescopes, and we find

$$\begin{aligned} q(\xi) &= \prod_{i=1}^{n-1} q(G_i \mid \tilde{G}_{i-1}) \\ &\propto \prod_{i=1}^{n-1} \frac{\tau(G_i) \tau(\tilde{G}_i)}{\tau(\tilde{G}_{i-1}) k_i} |\mathcal{C}(G_i, \tilde{G}_i)| \mathbf{1}_{\{\text{pop}(V_i) \in [P_i^-, P_i^+]\}} \\ &= \mathbf{1}_{\{\text{dev}(\xi) \leq D\}} \frac{\tau(\xi)}{\tau(G)} \prod_{i=1}^{n-1} k_i^{-1} |\mathcal{C}(G_i, \tilde{G}_i)|. \end{aligned} \quad \square$$

**Proposition 2.** *A set of weighted samples  $\{(\xi^{(j)}, w^{(j)})\}_{j=1}^S$  from Algorithm 2 is proper with respect to the target distribution  $\pi$ , i.e.,*

$$\mathbb{E}_q[h(\xi^{(j)}) w^{(j)}] = C \mathbb{E}_\pi[h(\xi^{(j)})],$$

for any square integrable function  $h$ , all  $j$ , and  $C$  a common normalizing constant.

*Proof.* It suffices to show that the product of the splitting procedure sampling probability (Proposition 1) and the importance sampling weights at each stage equals the target probability:

$$\begin{aligned} q(\xi^{(j)}) w^{(j)} &\prod_{i=1}^{n-1} \left(w_i^{(j)}\right)^\alpha \\ &= q(\xi^{(j)} \mid \text{dev}(\xi^{(j)}) \leq D) \exp\{-J(\xi^{(j)})\} \tau(G_n^{(j)})^{\rho-1} \prod_{i=1}^{n-1} w_i^{(j)} \end{aligned}$$

$$\begin{aligned}
&\propto \mathbf{1}_{\{\text{dev}(\xi) \leq D\}} \exp\{-J(\xi^{(j)})\} \frac{\tau(\xi^{(j)})}{\tau(G)} \tau(G_n^{(j)})^{\rho-1} \prod_{i=1}^{n-1} \frac{|\mathcal{C}(G_i^{(j)}, \tilde{G}_i^{(j)})|}{k_i} \cdot \frac{\tau(G_i^{(j)})^{\rho-1} k_i}{|\mathcal{C}(G_i^{(j)}, \tilde{G}_i^{(j)})|} \\
&\propto \exp\{-J(\xi^{(j)})\} \tau(\xi^{(j)})^\rho \mathbf{1}_{\{\text{dev}(\xi) \leq D\}} \\
&= \pi(\xi^{(j)}). \quad \square
\end{aligned}$$

**Proposition 3.** *The probability that an edge  $e$  is selected to be cut at iteration  $i$ , given that the tree  $T$  containing  $e$  has been drawn, and that  $e$  would induce a valid district, satisfies*

$$\max \left\{ 0, q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}) \left( 1 + \frac{1}{k_i} \right) - 1 \right\} \leq q(e = e^* \mid \mathcal{F}) \leq \frac{1}{k_i},$$

where  $\mathcal{F} = \sigma(\{T, \text{pop}(V_i) \in [P_i^-, P_i^+]\})$ .

*Proof.* We can write

$$q(e = e^* \mid \mathcal{F}) = q(e = e^*, d_e \leq d_{e_{k_i}} \mid \mathcal{F}) = \frac{1}{k_i} q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}),$$

This holds because the edge  $e$  will not be cut unless  $d_e \leq d_{e_{k_i}}$ , i.e., if  $e$  is among the top  $k_i$  edges. We then have immediately that  $q(e = e^* \mid \mathcal{F}) \leq k_i^{-1}$ . Additionally, using the lower Fréchet inequality, we find the lower bound

$$\begin{aligned}
q(e = e^* \mid \mathcal{F}) &= q(e = e^*, d_e \leq d_{e_{k_i}} \mid \mathcal{F}) \\
&\geq \max \left\{ 0, q(e = e^* \mid \mathcal{F}) + q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}) - 1 \right\} \\
&= \max \left\{ 0, \frac{1}{k_i} q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}) + q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}) - 1 \right\} \\
&= \max \left\{ 0, q(d_e \leq d_{e_{k_i}} \mid \mathcal{F}) \left( 1 + \frac{1}{k_i} \right) - 1 \right\}. \quad \square
\end{aligned}$$

## B Additional Validation Example

This section reports the results of another validation study applied to the same 50-precinct Florida map used in Section 5. Here, however, we study partitions into four contiguous districts, rather than three, of which there are over 333 million. In this example, too, the proposed SMC algorithm can efficiently approximate several target distributions on this set of partitions under different sets of constraints.

(a)  $\rho = 0.5$  and  
 $\text{dev}(\xi) \leq 0.025$

(b)  $\rho = 0$ ,  $\text{rem}(\xi) \leq 0.4$ ,  
and  $\text{dev}(\xi) \leq 0.05$

(c)  $\rho = 1$  and  
 $\text{dev}(\xi) \leq 0.015$

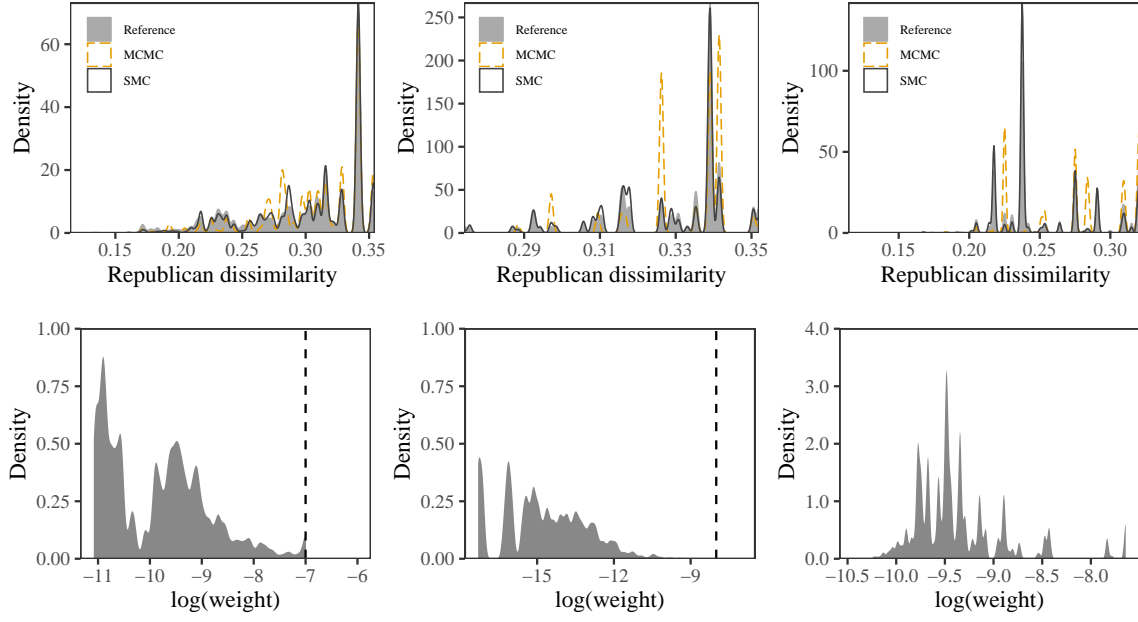


Figure 11: Calibration plots for Republican dissimilarity under various target measures, showing density estimates of the algorithm output and the reweighted enumeration. The distribution of importance weights for each sample is shown below each plot, with the truncation value (if applicable) marked with a vertical line.

First, we target a distribution with moderate compactness and parity constraints by choosing  $\rho = 0.5$  in equation (1) and setting the population constraint to  $\text{dev}(\xi) \leq 0.025$ . There are only 1386 partitions (or 0.00042% of all maps) that satisfy this population constraint in the reference enumeration. Since the target distribution is not uniform, we reweight the enumerated maps according to  $\tau(\xi)^{0.5}$ . We sampled 10,000 plans using the proposed algorithm and reweighted them according to the importance weights, using a normalized weight truncation of  $w_{\max} = 0.1 \times \sqrt{10,000}$  (see the lower panel of Figure 11b for the distribution of weights after truncation). The upper panel of Figure 11b shows the resulting density estimates. While the target distribution is highly multimodal under this summary statistic, there is good agreement between the SMC sample and reference distribution. The MCMC algorithm, too, performs relatively well, despite some spurious peaks in the 0.26–0.33 range, and a lack of samples for the mode at  $\approx 0.23$ .

Second, we target a uniform distribution on the set of maps with  $\text{dev}(\xi) \leq 0.05$  and

$\text{rem}(\xi) \leq 0.4$ . The median fraction of removed edges across all partitions under 5% population deviation was 63%. There are a total of only 108 maps (or 0.00003% of all maps) that satisfy these two constraints. As before, we sampled 10,000 plans and truncated weights to  $w_{\max} = 0.04 \times \sqrt{10,000}$  (see the lower panel of Figure 11a for the distribution of truncated weights). We discarded any samples which did not meet the compactness constraint, leaving 6,873 samples. The upper panel of the figure shows the results. While the SMC algorithm again agrees well with the reference distribution, the MCMC algorithm under-samples the main mode, and significantly oversamples the peaks at  $\approx 0.325$  and  $\approx 0.34$ .

Finally, we sample from a distribution with a strong population parity constraint and compactness constraint by choosing  $\text{dev}(\xi) \leq 0.015$  and setting  $\rho = 1$ . Only 277 maps (or 0.00008% of all maps) that satisfy this population constraint. We sample 10,000 plans but do not truncate the importance weights, since with  $\rho = 1$  their variance is small, as shown in the lower panel of Figure 11c. As in the first validation exercise, the target measure is not uniform, and the upper panel of Figure 11c shows the density estimates for the generated sample and reference set. Once again the agreement between the two is excellent. The MCMC algorithm, in contrast, significantly oversamples several modes while missing the main mode at  $\approx 0.24$  completely.