JOURNAL OF COMPUTATIONAL AND GRAPHICAL STATISTICS
2020, VOL. 00, NO. 0, 1–14
https://doi.org/10.1080/10618600.2020.1739532

Taylor & Francis
Taylor & Francis Group

Automated Redistricting Simulation Using Markov Chain Monte Carlo

Benjamin Fifield[a], Michael Higgins[b], Kosuke Imai[a,c,d], and Alexander Tarr[e]

[a]Institute for Quantitative Social Science, Harvard University, Cambridge, MA; [b]Department of Statistics, Kansas State University, Manhattan, KS; [c]Department of Government, Harvard University, Cambridge, MA; [d]Department of Statistics, Harvard University, Cambridge, MA; [e]Department of Electrical Engineering, Princeton University, Princeton, NJ

ABSTRACT
Legislative redistricting is a critical element of representative democracy. A number of political scientists have used simulation methods to sample redistricting plans under various constraints to assess their impact on partisanship and other aspects of representation. However, while many optimization algorithms have been proposed, surprisingly few simulation methods exist in the published scholarship. Furthermore, the standard algorithm has no theoretical justification, scales poorly, and is unable to incorporate fundamental constraints required by redistricting processes in the real world. To fill this gap, we formulate redistricting as a graph-cut problem and for the first time in the literature propose a new automated redistricting simulator based on Markov chain Monte Carlo. The proposed algorithm can incorporate contiguity and equal population constraints at the same time. We apply simulated and parallel tempering to improve the mixing of the resulting Markov chain. Through a small-scale validation study, we show that the proposed algorithm can approximate a target distribution more accurately than the standard algorithm. We also apply the proposed methodology to data from Pennsylvania to demonstrate the applicability of our algorithm to real-world redistricting problems. The open-source software package is available so that researchers and practitioners can implement the proposed methodology. Supplementary materials for this article are available online.

ARTICLE HISTORY
Received October 2018
Revised December 2019

KEYWORDS
Gerrymandering; Graph cuts; Metropolis–Hastings algorithm; Parallel tempering; Simulated tempering; Swendsen–Wang algorithm

1. Introduction

Legislative redistricting is a critical element of representative democracy. Scholars have found that redistricting influences turnout and representation (e.g., Ansolabehere, Snyder, and Stewart 2000; McCarty, Poole, and Rosenthal 2009). Unfortunately, redistricting is also potentially subject to partisan gerrymandering. After the controversial 2003 redistricting in Texas, for example, Republicans won 21 congressional seats in the 2004 election (Democrats won 11) whereas they had only 15 seats in 2002. Moreover, the United States Supreme Court has recently ruled in *Rucho v. Common Cause* that federal courts cannot address partisan gerrymandering, essentially leaving state legislatures, state courts, and Congress to resolve the issue. Scholars have proposed numerous remedies for partisan gerrymandering, including geographical compactness and partisan symmetry requirements (e.g., Grofman and King 2007; Fryer and Holden 2011).

The development of automated redistricting algorithms, which is the goal of this article, began in the 1960s. Vickrey (1961) argued that such an “automatic and impersonal procedure” can eliminate gerrymandering (p. 110). After *Baker v. Carr* (1962) where the Supreme Court ruled that federal courts may review the constitutionality of state legislative apportionment, citizens, policy makers, and scholars became interested in redistricting. Weaver and Hess (1963) and Nagel (1965) were among the earliest attempts to develop automated redistricting algorithms. Since then, a large number of methods have been developed to find an *optimal* redistricting plan for a given set of criteria (e.g., Bozkaya, Erkut, and Laporte 2003; Chou and Li 2006; Fryer and Holden 2011; Liu, Tam Cho, and Wang 2016; Tam Cho and Liu 2016). These optimization methods may serve as useful tools when drawing district boundaries (see Altman, MacDonald, and McDonald 2005, for an overview).

However, the main interest of scholars has been to characterize the *population distribution* of possible redistricting plans under various criteria so that they can detect instances of gerrymandering and better understand the causes and consequences of redistricting (e.g., Cirincione, Darling, and O’Rourke 2000; McCarty, Poole, and Rosenthal 2009; Chen and Rodden 2013). Because redistricting in any given state depends critically on its geographical features including the residential patterns of voters, it is essential to identify the distribution of possible redistricting maps under the basic conditions of contiguity and equal population. In a majority of the U.S. states, for example, state legislators control the redistricting process and approve redistricting plans through standard statutory means. Therefore, an important question is how to effectively constrain these politicians through means such as compactness requirements (e.g., Niemi et al. 1990), and prevent

CONTACT Kosuke Imai imai@harvard.edu Department of Government and Department of Statistics, Harvard University, Cambridge, MA 02138.
An earlier version of this article was entitled, “A New Automated Redistricting Simulator Using Markov Chain Monte Carlo” (Fifield et al. 2014) and was presented at the 2014 Annual Summer Meeting of the Society for Political Methodology (The University of Georgia).
Supplementary materials for this article are available online. Please go to www.tandfonline.com/r/JCGS.

the manipulation of redistricting for partisan ends. Simulation methods allow scholars to answer these questions by approximating distributions of possible electoral outcomes under various constraints.

Yet, until recently, surprisingly few simulation algorithms have existed in the published scholarship. In fact, most of these existing studies use essentially the same Monte Carlo simulation algorithm where a geographical unit is randomly selected as a "seed" for each district and then neighboring units are added to contiguously grow this district until it reaches the prespecified population threshold (e.g., Cirincione, Darling, and O'Rourke 2000; Chen and Rodden 2013). Unfortunately, no theoretical justification is given for these simulation algorithms, and hence they are unlikely to yield a representative sample of redistricting plans for a target population. In addition, although a commonly used algorithm of this type (Cirincione, Darling, and O'Rourke 2000) is implemented by Altman and McDonald (2011) in their open-source software, the algorithm scales poorly.

To fulfill this methodological gap, in Section 2, we propose a new automated redistricting simulator using Markov chain Monte Carlo (MCMC). We first formulate the task of drawing district boundaries as the problem of graph-cuts, that is, partitioning a graph into several connected subgraphs. We then adapt and modify the Swendsen–Wang algorithm to sample from arbitrary distributions over a fixed number of contiguous districts (Swendsen and Wang 1987; Barbu and Zhu 2005). We then show how this algorithm can be applied to sample redistricting plans that incorporate the equal population constraint commonly imposed by law. Finally, we apply simulated and parallel tempering to improve the mixing of the resulting Markov chain (Marinari and Parisi 1992; Geyer and Thompson 1995).[1] We provide an open-source R software package redist, which is freely available on the Comprehensive R Archive Network (CRAN; *https://CRAN.R-project.org/package=redist*) so that researchers and practitioners can implement the proposed methodology (Fifield, Tarr, and Imai 2015).

Unlike the aforementioned standard simulation algorithms, the proposed algorithms are designed to yield a representative sample of redistricting plans under contiguity and equal population constraints. Since the first version of this article was made available in 2014 (Fifield et al. 2014), MCMC algorithms have become a commonly used method for simulating redistricting plans (e.g., Mattingly and Vaughn 2014; Chikina, Frieze, and Pegden 2017; Herschlag, Ravier, and Mattingly 2017; DeFord, Duchin, and Solomon 2019). Our algorithms are similar to the algorithm independently proposed at about the same time by Mattingly and Vaughn (2014). While the latter algorithm samples redistricting plans by swapping single precincts at each iteration, our algorithms allow multiple, larger collections of precincts to be swapped, improving the mixing of the Markov chain. We also offer theoretical and empirical analyses of our algorithms in an effort to better understand their properties and performance.

In Section 3, we conduct a small-scale validation study where all possible redistricting plans under various constraints can be enumerated in a reasonable amount of time. We show that the proposed algorithms successfully approximate the target population distribution while the standard algorithm fails even in this small-scale problem. We then apply our algorithms to a large redistricting problem using data from Pennsylvania. We find that small changes to the adopted redistricting plan can nearly eliminate the partisan bias. Finally, Section 4 discusses remaining challenges and open problems for the development of automated redistricting simulation algorithms.

2. The Proposed Methodology

In this section, we describe the proposed methodology. We begin by formulating redistricting as a graph-cut problem and propose an MCMC algorithm to sample redistricting plans from arbitrary distributions over the set of n contiguous districts. Next, we show how our algorithm can be used to sample plans that have an equal population constraint, which is the most basic requirement imposed in real-world redistricting problems. We then discuss how to apply parallel tempering to our MCMC algorithm to improve the mixing of the Markov chain. Finally, a brief discussion comparing our algorithm with the existing redistricting algorithms is also given.

2.1. Redistricting as a Graph-Cut Problem

Consider a redistricting problem where a state consisting of m geographical units (e.g., census blocks or voting precincts) must be divided into n contiguous districts. We formulate this redistricting problem as that of graph-cuts, where a graph is partitioned into a set of connected subgraphs (Altman 1997; Mehrotra, Johnson, and Nemhauser 1998). Figure 1 illustrates the basic idea through two small examples used in our validation study (see Section 3.1). In each plot, a state is represented by a graph where nodes are geographical units and edges between two nodes imply their contiguity. Thus, redistricting a state into n districts is equivalent to removing some edges of a graph (light gray) and forming n connected subgraphs.

Formally, let $G = (V, E)$ represent an undirected graph where $V = \{1, 2, \ldots, m\}$ is the set of nodes (i.e., geographical units of redistricting) to be partitioned, and E is the set of undirected edges connecting neighboring nodes. This means that if two units i and j are contiguous, there is an edge between their corresponding nodes on the graph, $i \sim j \in E$. We partition the set of nodes V into n "districts" $\pi = \{V_1, V_2, \ldots, V_n\}$ where each district V_ℓ is a nonempty subset of V. Such a partition π generates a subgraph $G(\pi) = (V, E(\pi)) \subseteq G$ where an edge $i \sim j \in E(\pi)$ if and only if nodes i and j belong to the same district of the partition V_ℓ, that is,

$$E(\pi) = \{i \sim j \in E : \exists\ V_\ell \in \pi \text{ s.t. } i, j \in V_\ell\}. \qquad (1)$$

Because districts are formed by removing edges from E (i.e., "cutting" them) to obtain $E(\pi)$, redistricting can be thought of as a graph cut problem. Finally, since each resulting district must be contiguous, we require each district of the partition to be connected. In this article, we call a partition comprised of n connected districts *valid* and denote the set of all valid partitions by $\Omega(G, n)$.

[1] The application of simulated tempering is presented in Appendix S4.

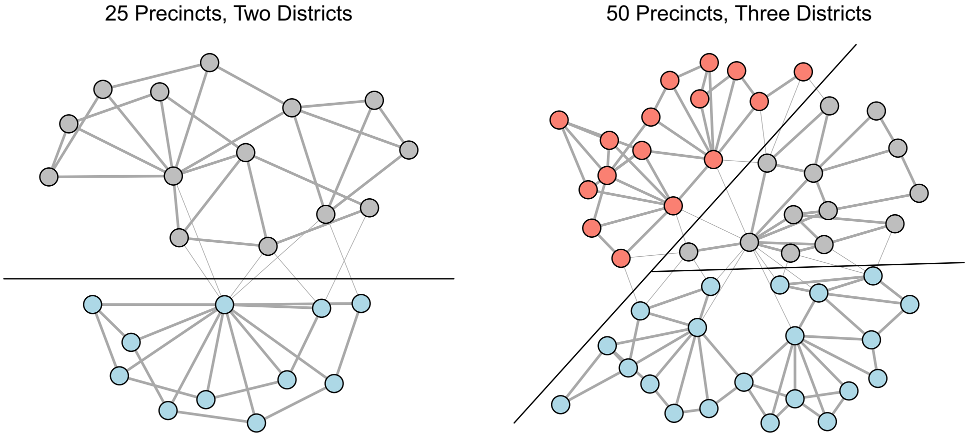

Figure 1. Redistricting as a graph-cut problem. A state is represented by a graph where nodes are geographical units and edges between two nodes imply their contiguity. Under this setting, redistricting is equivalent to removing or cutting some edges (light gray) to form connected subgraphs, which correspond to districts. Different districts are represented by different colors. Two illustrative examples, used in our validation study in Section 3.1, are given here.

2.2. The Basic Algorithm for Sampling Contiguous Districts Without Constraints

The goal of this article is to develop an algorithm that uniformly samples valid redistricting plans with the equal population constraint. We achieve this in several steps. We first describe the basic algorithm that samples from an arbitrary distribution f over the set of valid partitions, $\Omega(G, n)$ (Algorithm 1 in Sections 2.2 and 2.3). We then show how to modify this basic algorithm to incorporate the equal population constraint, which is the most basic requirement imposed in real-world redistricting problems (Algorithms 1.1 and 1.2 in Section 2.4). Finally, we apply parallel tempering to improve the mixing of the proposed MCMC algorithm (Algorithm 2 in Section 2.5).

Our basic MCMC algorithm is designed to obtain a dependent but representative sample from any distribution over valid redistricting plans. In particular, we modify the SWC-1 algorithm of Barbu and Zhu (2005), which uses a Metropolis–Hastings step (Metropolis et al. 1953; Hastings 1970) to extend the Swendsen–Wang algorithm (Swendsen and Wang 1987) to arbitrary distributions over graph partitions. In contrast to our method, graph partitions sampled from the original SWC-1 algorithm are not restricted to be contiguous. Our basic algorithm begins with a valid partition π_0 (e.g., an actual redistricting plan adopted by the state) and transitions from a valid partition π_t to another valid partition π_{t+1} at each iteration $t + 1$. We begin by describing the basic algorithm for sampling contiguous districts.

Figure 2 illustrates one iteration of Algorithm 1 using the 50 precinct example with three districts given in the right panel of Figure 1. Our algorithm begins by randomly "turning on" edges in $E(\pi_t)$ where each edge is turned on with probability q. In the left upper plot of Figure 2, the edges that are turned on are indicated with darker gray. Next, we identify components that are connected through these "turned-on" edges and are on the boundaries of districts in π_t. Each such connected component is indicated by a dotted polygon in the right upper plot. Third, among these, a subset of nonadjacent connected components are randomly selected as shown in the left lower plot (two in this case). These connected components are reassigned to adjacent districts to create a candidate partition after verifying that this reassignment does not shatter any district so that the number of districts remain unchanged. Finally, we accept or reject the candidate partition according to a probability that depends on, among other things, "turned-on" edges, and "turned-off" edges from each of the selected connected components that are connected to adjacent districts and are highlighted in the left lower plot. Section 2.3 provides a detailed explanation about how this acceptance probability is computed.

We now formally describe the basic algorithm.

Algorithm 1 (Sampling contiguous redistricting plans). We initialize the algorithm by obtaining a valid partition $\pi_0 = \{V_1^{(0)}, V_2^{(0)}, \dots, V_n^{(0)}\}$, where each district $V_\ell^{(0)}$ is contiguous, that is, connected in the graph, and then repeat the following steps at each iteration $t + 1$,

Step 1 ("Turn on" edges): Starting from $\pi_t = \{V_1^{(t)}, V_2^{(t)}, \dots, V_n^{(t)}\}$ with graph $G(\pi_t) = (V, E(\pi_t))$, form the edge set $E_{on}(\pi_t) \subseteq E(\pi_t)$, where each edge $e \in E(\pi_t)$ is independently added to $E_{on}(\pi_t)$ with probability q. Let **CP** denote the set of connected components formed by $E_{on}(\pi_t)$.

Step 2 (Gather connected components along boundaries):

(a) Identify all nodes which lie along a district boundary by comparing $E(\pi_t)$ with E.

(b) Gather boundary connected components by performing a breadth-first search over all boundary nodes. Let $B(\mathbf{CP}, \pi_t)$ denote this set.

Step 3 (Select nonadjacent connected components along boundaries):

(a) Generate R from a distribution with integer support over the closed interval $[1, |B(\mathbf{CP}, \pi_t)|]$.

(b) Initialize $\mathbf{V_{CP}} = \emptyset$. Repeat the following procedure until $|\mathbf{V_{CP}}| = R$:

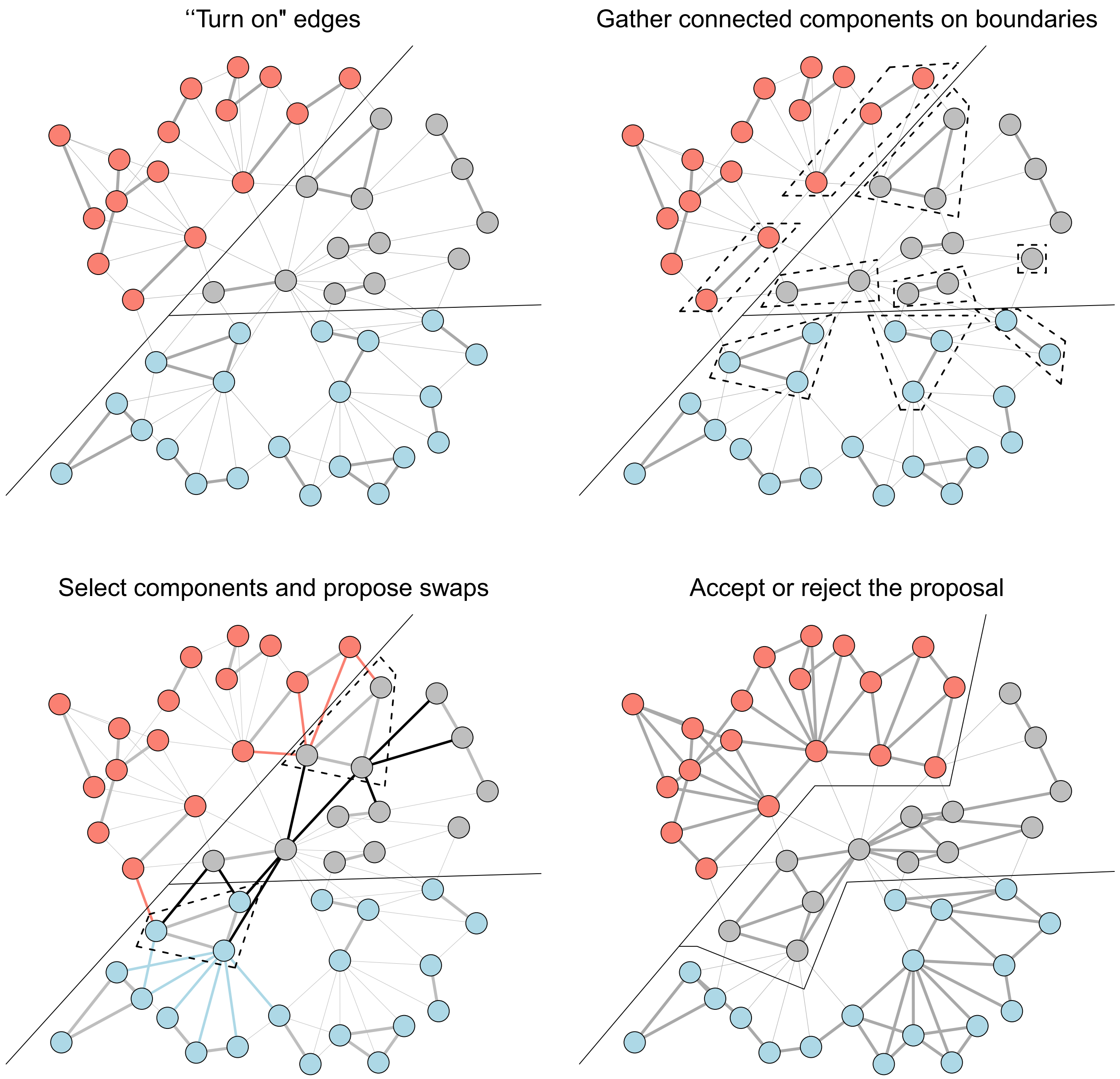
“Turn on" edges
Gather connected components on boundaries
Select components and propose swaps
Accept or reject the proposal

Figure 2. The basic algorithm for sampling contiguous districts without constraints. The plots illustrate the proposed algorithm (Algorithm 1) using the 50 precinct data given in the right panel of Figure 1. First, in the left upper plot, each edge other than those which are cut in Figure 1 is “turned on” (dark gray) independently with certain probability. Second, in the right upper plot, connected components on the boundaries are identified (dashed polygons). Third, in the left lower plot, a certain number of nonadjacent connected components on boundaries are randomly selected (dashed polygons) and the acceptance ratio is calculated by counting certain edges (colored edges). Finally, in the right lower plot, the proposed swap is accepted according to the Metropolis–Hastings ratio.

(i) Randomly sample C from $B(\mathbf{CP}, \pi_t)$ without replacement, and add it to $\mathbf{V_{CP}}$.

(ii) If C is adjacent to another element of $\mathbf{V_{CP}}$, or if the removal of $\mathbf{V_{CP}}$ results in a noncontiguous district, remove C from $\mathbf{V_{CP}}$ and $B(\mathbf{CP}, \pi_t)$, and return to Step (i).

Step 4 (Propose swaps): Initialize $\pi' = \pi_t$. For each $C \in \mathbf{V_{CP}}$, perform the following procedure:

(a) Suppose C is currently assigned to district $V_\ell^{(t)}$. Propose to assign C to a different, randomly selected neighboring district in π'.

(b) Suppose C is reassigned to district $V_{\ell'}^{(t)}$. Update π' by setting $V_\ell^{(t)} = V_\ell^{(t)} \setminus C$, $V_{\ell'}^{(t)} = V_{\ell'}^{(t)} \cup C$, and $\pi' = \left\{V_1^{(t)}, V_2^{(t)}, \ldots, V_\ell^{(t)}, \ldots, V_{\ell'}^{(t)}, \ldots, V_n^{(t)}\right\}$.

Step 5 (Accept or reject the proposal): Generate $u \sim U(0, 1)$. Set

$$\pi_{t+1} = \begin{cases} \pi' & \text{if } u \leq \alpha(\pi', \mathbf{CP} \mid \pi_t), \\ \pi_t & \text{otherwise}, \end{cases} \tag{2}$$

where the acceptance probability $\alpha(\pi', \mathbf{CP} \mid \pi_t)$ is defined in Theorem 1.

Several remarks are in order. First, in Step 1, we choose q to be relatively small such that it typically yields a large number of small connected components on boundaries. In the validation studies, we set q to 0.05, and for the Pennsylvania study, we set q to 0.04.

Second, in Step 3(a), we choose a distribution such that $R \ll |B(\mathbf{CP}, \pi)|$ holds for most cases and the acceptance probability is in an ideal range of 20%–40%. In the applications of this article, for example, we use the zero-truncated Poisson distribution, rejecting and resampling values of R which fall outside the interval $[1, |B(\mathbf{CP}, \pi)|]$. We choose λ such that the chance of rejection is small, with $\lambda = 2$ in our validation studies in Section 3.1 and $\lambda = 10$ in the Pennsylvania study presented in Section 3.2.

Alternatively, as in Barbu and Zhu (2005), we could fix R to 1 and use a large value of q. Unfortunately, this strategy, though sensible in other settings, is not effective in redistricting with contiguous districts. The reason is that larger connected components typically include more units from the interior of each district. This in turn makes it more likely to shatter districts and often dramatically lowers the acceptance probability, which leads to slower convergence. Instead, our strategy is to improve the mixing of the Markov chain through simultaneous swaps of small connected components.

Third, Step 3(b) checks whether or not swapping connected components $\mathbf{V_{CP}}$ leads to an invalid redistricting plan, that is, a district is no longer contiguous. This requirement does not exist in the original algorithm of Barbu and Zhu (2005). Such “shattering” could occur, for example, if a district contains a narrow sliver that has width of a single precinct, and one of the precincts in the middle of the sliver is selected to be swapped. This check also enforces that $\mathbf{V_{CP}}$ consists of nonadjacent connected components, and hence $\mathbf{V_{CP}}$ is the unique set of components to be reassigned in the move $\pi_t \to \pi'$.

Finally, the acceptance probability in Step 5 is based on the following Metropolis criterion,

$$\alpha(\pi', \mathbf{CP} \mid \pi) = \min\left(1, \frac{q(\pi, \mathbf{CP} \mid \pi')}{q(\pi', \mathbf{CP} \mid \pi)} \cdot \frac{g(\pi')}{g(\pi)}\right), \tag{3}$$

where $q(\pi', \mathbf{CP} \mid \pi) = q(\pi' \mid \mathbf{CP}, \pi) q(\mathbf{CP} \mid \pi)$ denotes the probability that, starting from partition π, connected component set $\mathbf{CP}$ is formed in Step 1, and π' is formed through reassignment of elements in $\mathbf{CP}$ in Steps 2–4. In addition, $g(\pi) \propto f(\pi)$ is the unnormalized target distribution for sampling. Barbu and Zhu (2005) computed the acceptance ratio by marginalizing out $\mathbf{CP}$ in the proposal distributions, eliminating dependence on $\mathbf{CP}$. However as we discuss in Section 2.3, the marginalization of the proposal distributions is intractable in most redistricting problems. Thus, we cannot eliminate this dependence on $\mathbf{CP}$, leading to slower convergence of the Markov chain.

Next, we present the acceptance probability used in Step 5 of Algorithm 1.

Theorem 1 (The acceptance probability of Algorithm 1). Let π and π' be a pair of partitions which differ in the district assignment of $\mathbf{V_{CP}}$, with $|\mathbf{V_{CP}}| = R$ and $R \leq |B(\mathbf{CP}, \pi')|$, and let $g(\pi) \propto f(\pi)$ be the unnormalized target distribution over $\Omega(G, n)$. Then, the acceptance probability for the move $\pi \to \pi'$ through one iteration of Algorithm 1 is

$$\begin{aligned} &\alpha(\pi', \mathbf{CP} \mid \pi) \\ &= \min\left(1, \frac{P\left(\mathbf{V_{CP}} \mid \pi', \mathbf{CP}, R\right) F(|B(\mathbf{CP}, \pi)|) \left(1-q\right)^{|\mathcal{C}(\pi', \mathbf{V_{CP}})|}}{P\left(\mathbf{V_{CP}} \mid \pi, \mathbf{CP}, R\right) F(|B(\mathbf{CP}, \pi')|) \left(1-q\right)^{|\mathcal{C}(\pi, \mathbf{V_{CP}})|}} \cdot \frac{g(\pi')}{g(\pi)}\right), \end{aligned} \tag{4}$$

where

$$\mathcal{C}(\pi, \mathbf{V_{CP}}) = \left\{ i \sim j \in E(\pi) : \exists\, C \in \mathbf{V_{CP}} \; s.t.\; i \in C, j \notin C \right\} \tag{5}$$

is a Swendsen–Wang cut, denoting the set of edges in $E(\pi)$ that need to be cut to form $\mathbf{V_{CP}}$, $F(\cdot)$ is the cumulative distribution function for R whose support is a subset of the set of positive integers, and $P(\mathbf{V_{CP}} \mid \pi, \mathbf{CP}, R)$ is the probability of sampling $\mathbf{V_{CP}}$ in Step 3(b) of Algorithm 1.

The proof is given in Appendix S1. Note that in the case of sampling contiguous redistricting plans without constraints, the target distribution is uniform over all contiguous n-partitions of the graph, $\Omega(G, n)$, and $g(\pi) = g(\pi')$ cancels in the ratio of Equation (4).

Finally, the following theorem shows that under a set of assumptions, the unique stationary distribution of Algorithm 1 is $g(\pi)$, whose support is restricted to $\Omega(G, n)$.

Theorem 2 (Unique stationary distribution of Algorithm 1). Suppose that the following two conditions hold:

1. (Irreducibility) For all pairs $\pi, \pi' \in \Omega(G, n)$, partition π can transition to partition π' through a finite number of iterations of Algorithm 1.
2. (Aperiodicity) There exists a pair of partitions π, π' and set of connected components $\mathbf{CP}$ such that $0 < \alpha(\pi', \mathbf{CP} \mid \pi) < 1$.

Then, the unique stationary distribution of the Markov chain given by Algorithm 1 is $f(\pi)$.

The proof is given in Appendix S2. We remark that although it is difficult to provide a formal argument, our experience suggests that Conditions 1 and 2 may hold for real-world redistricting data unless additional constraints are imposed (see Section 2.4 for a discussion of how we modify the algorithm to incorporate the equal population constraint). For the small-scale validation example we have examined, in which the complete enumeration of valid partitions is possible, we were able to verify that the two conditions are satisfied. In addition, for all the empirical examples we have analyzed including the New Hampshire and Pennsylvania data, we have found at least one set of initial and candidate partitions such that $0 < \alpha(\pi', \mathbf{CP} \mid \mathbf{CP}) < 1$.

2.3. Approximating the Acceptance Probability

The acceptance probability given in Equation (4) of Theorem 1 differs from the ratio derived in Barbu and Zhu (2005) in the two terms, $F(|B(\mathbf{CP}, \pi)|)$ and $P(\mathbf{V_{CP}}, \mid \pi, \mathbf{CP}, R)$, which correspond to the probability of sampling R and $\mathbf{V_{CP}}$ in Step 3, respectively. Note that the computation of $F(|B(\mathbf{CP}, \pi)|)$ is straightforward.

As mentioned earlier, in our applications, we sample R from the zero-truncated Poisson distribution truncated from above by $|B(\mathbf{CP}, \pi)|$. Hence, the required cumulative distribution function can be easily computed. The key question is then how to compute $P(\mathbf{V_{CP}}, \mid \pi, \mathbf{CP}, R)$.

In the original algorithm of Barbu and Zhu (2005), R is fixed to 1, and $\mathbf{V_{CP}}$ is selected uniformly at random from $\mathbf{CP}$, so $P(\mathbf{V_{CP}}, \mid \pi', \mathbf{CP}, R)/P(\mathbf{V_{CP}}, \mid \pi, \mathbf{CP}, R) = 1$. For redistricting problems, however, this cancellation does not occur, even when $R = 1$, due to the possibility of shattering and the requirement that components $\mathbf{V_{CP}}$ lie along district boundaries. In addition, as mentioned earlier, for our applications, setting $R = 1$ and using a large value of q lead to a high rate of rejection and poor mixing. Thus, we let R vary, which further complicates computation of $P(\mathbf{V_{CP}}, \mid \pi, \mathbf{CP}, R)$ due to the nonadjacency restriction of $\mathbf{V_{CP}}$. Below, we propose an approximation of the acceptance probability when the exact computation of this probability is computationally infeasible.

Recall that in Step 3(b), we iteratively sample without replacement R connected components from the sampling set $B(\mathbf{CP}, \pi)$ to form the set $\mathbf{V_{CP}}$. During this process, a sampled connected component whose addition to $\mathbf{V_{CP}}$ would lead to either component adjacency or the creation of disconnected districts is rejected and removed from $B(\mathbf{CP}, \pi)$. Let $B_R^{\dagger}(\mathbf{CP}, \pi)$ represent the collection of nonadjacent connected component sets $\mathbf{V_{CP}} \subset B(\mathbf{CP}, \pi)$, with $|\mathbf{V_{CP}}| = R$, whose removal from the graph partition π does not result in disconnected districts. Ideally, the rejection sampling strategy described above would correspond to uniform sampling from $B_R^{\dagger}(\mathbf{CP}, \pi)$, in which case,

$$P(\mathbf{V_{CP}} \mid \pi, \mathbf{CP}, R) = \frac{1}{|B_R^{\dagger}(\mathbf{CP}, \pi)|}. \tag{6}$$

However, since components in $B(\mathbf{CP}, \pi)$ have a differing number of neighbors, some elements of $B_R^{\dagger}(\mathbf{CP}, \pi)$ are more likely to be sampled than others. Consequently, $P(\mathbf{V_{CP}} \mid \pi, \mathbf{CP}, R)$ has a more complicated form that depends on $B_R^{\dagger}(\mathbf{CP}, \pi)$.

Additionally, identification of $B_R^{\dagger}(\mathbf{CP}, \pi)$ is computationally intensive even for the case $R = 1$ because it requires checking whether each connected component $C \in B(\mathbf{CP}, \pi)$ can shatter its district V_ℓ; this has worst-case complexity $O(|V_\ell| + |E(V_\ell)|)$. The problem quickly becomes intractable when $R > 1$, even for small graphs, since we have to check both connectedness and nonadjacency for a combinatorial number of subsets of $B(\mathbf{CP}, \pi)$. We therefore approximate this probability.

Fortunately, in large-scale problems, so long as q and R are small, the number of connected component sets $\mathbf{V_{CP}}$ that can shatter a district is likely to be small relative to the total number of boundary connected component sets. When this is the case, at most $\mathbf{V_{CP}} \in B_R^{\dagger}(\mathbf{CP}, \pi)$ can be formed by sampling R nonadjacent components from $B(\mathbf{CP}, \pi)$. Additionally, small q and R tend to lead to scenarios where $R \ll |B(\mathbf{CP}, \pi)|$. In this case, without-replacement sampling can be approximated by with-replacement sampling since the probability of sampling the same or a neighboring component in R draws is small. Therefore, we propose the following approximation,

$$P(\mathbf{V_{CP}} \mid \pi, \mathbf{CP}, R) \approx R! \left(\frac{1}{|B(\mathbf{CP}, \pi)|}\right)^R. \tag{7}$$

Substituting the approximation into the acceptance probability defined in Equation (4), we obtain

$$\begin{aligned} &\alpha(\pi', \mathbf{CP} \mid \pi) \\ &\approx \min\left(1, \left(\frac{|B(\mathbf{CP}, \pi)|}{|B(\mathbf{CP}, \pi')|}\right)^R \frac{F(|B(\mathbf{CP}, \pi)|)\,(1-q)^{|\mathcal{C}(\pi', \mathbf{V_{CP}})|}}{F(|B(\mathbf{CP}, \pi')|)\,(1-q)^{|\mathcal{C}(\pi, \mathbf{V_{CP}})|}} \cdot \frac{g(\pi')}{g(\pi)}\right). \end{aligned} \tag{8}$$

This approximation is valid under the assumption that we rarely reject samples drawn in Step 3(b) for adjacency or shattering issues. Indeed, in the Pennsylvania study, we find that only 190 out of 30,000 iterations of the algorithm (or <0.1%) rejected a proposed swap due to it either shattering an existing district or being adjacent to a swap proposed in the same iteration. Even for the small-scale validation study, at most 545 out of 10,000 iterations of the algorithm rejected a proposed swap. Finally, we only need the approximation to work in the ratio $P(\mathbf{V_{CP}}, \mid \pi, \mathbf{CP}, R)/P(\mathbf{V_{CP}}, \mid \pi', \mathbf{CP}, R)$ rather than the numerator and denominator separately.

Nevertheless, we acknowledge that it is difficult to develop a rigorous theoretical justification for the proposed approximation. Therefore, in Appendix S8, swe conduct small-scale validation studies, in which a stronger approximation based on $B_R^{\dagger}(\mathbf{CP}, \pi)$ can be computed, to examine both the accuracy of the proposed approximation and the effects of ignoring adjacency when computing the selection probability $P(\mathbf{V_{CP}} \mid \mathbf{CP}, \pi, R)$. We find that the stronger approximation nearly perfectly samples from the target population (see the first row of Figure S7), suggesting the effects of component adjacency have a negligible impact on the ratio of selection probabilities. We also find that even in small maps, where our weaker approximation is expected to perform poorly, the proposed approximation is reasonably accurate (see the second row of Figure S7) and becomes more accurate as the size of redistricting problem increases (see Figures S7–S9).

2.4. Incorporating the Equal Population Constraint

The basic algorithm we have developed so far samples from arbitrary distributions over the set of redistricting maps with n contiguous districts. We can incorporate the constraints imposed in typical redistricting processes through our choice of the target unnormalized distribution g. Such constraints include equal population, geographical compactness, and preservation of communities of interest. Here, we focus on the equal population constraint, which is the most basic factor to be considered for redistricting. While the proposed framework can accommodate other constraints as shown in Section 3.2, we leave the challenge of simultaneously incorporating multiple constraints to future research.

Let p_i denote the population size of node i where the population parity for the state is given by $\bar{p} = \sum_{i=1}^{m} p_i/n$. Then, the equal population constraint is defined as,

$$\max_{1 \le \ell \le n} \left| \frac{\sum_{i \in V_\ell} p_i}{\bar{p}} - 1 \right| \le \delta, \tag{9}$$

where δ determines the maximal deviation from the population parity. For example, $\delta = 0.03$ implies that the population of all districts must be within 3% of the population parity.

How can we uniformly sample redistricting plans that meet this population constraint using Algorithm 1? The major challenge we must deal with is that our algorithm generates samples from the set of contiguous partitions, $\Omega(G, n)$, while we would like to generate samples only from the much smaller set of plans meeting the constraint. One possible strategy for dealing with this issue, which is often used in the literature, is to sample uniformly from $\Omega(G, n)$ and discard any candidate partition that violates Equation (9). In Algorithm 1, for example, after Step 4, one could check whether the candidate partition π' satisfies the constraint, and if not, go back to Step 3. This procedure is summarized in the following algorithm, which we henceforth refer to in the text as the basic algorithm with hard constraint.

Algorithm 1.1 (Sampling contiguous redistricting plans with hard constraint). We initialize the algorithm by obtaining a valid partition $\pi_0 = \{V_1^{(0)}, V_2^{(0)}, \ldots, V_n^{(0)}\}$, where each district $V_\ell^{(0)}$ is contiguous, that is, connected in the graph, and then repeat the following steps at each iteration $t + 1$,

Step 1 (Run the basic algorithm): Starting from $\pi_t = \{V_1^{(t)}, V_2^{(t)}, \ldots, V_n^{(t)}\}$ with graph $G(\pi_t) = (V, E(\pi_t))$, run the first four steps of Algorithm 1 to obtain the proposed sample π'.

Step 2 (Reject samples failing to meet constraint): For a specified population constraint δ, reject π' and return to Step 3 of Algorithm 1 if

$$\max_{1 \le \ell \le n} \left| \frac{\sum_{i \in V_\ell^{(t)}} p_i}{\bar{p}} - 1 \right| > \delta,$$

otherwise continue to Step 3 of Algorithm 1.1.

Step 3 (Accept or reject the proposal): Generate $u \sim U(0, 1)$. Set

$$\pi_{t+1} = \begin{cases} \pi' & \text{if } u \le \alpha(\pi', \mathbf{CP} \mid \pi_t), \\ \pi_t & \text{otherwise,} \end{cases} \tag{10}$$

where the approximated acceptance probability $\alpha(\pi', \mathbf{CP} \mid \pi_t)$ is given by

$$\alpha(\pi', \mathbf{CP} \mid \pi_t) \approx \min\left(1, \left(\frac{|B(\mathbf{CP}, \pi_t)|}{|B(\mathbf{CP}, \pi')|}\right)^R \frac{F(|B(\mathbf{CP}, \pi_t)|)\,(1-q)^{|\mathcal{C}(\pi', \mathbf{V}_{\mathbf{CP}})|}}{F(|B(\mathbf{CP}, \pi')|)\,(1-q)^{|\mathcal{C}(\pi_t, \mathbf{V}_{\mathbf{CP}})|}}\right).$$

There are several potential problems with this approach. First, the additional rejection sampling done in Step 2 further complicates exact calculation of the acceptance probability. For simplicity, we use the same approximation given in Section 2.3, but this approximation may be poor especially for small values of the constraint δ. Second, there may be a vast majority of plans proposed through Steps 1–4 do not meet the equal population constraint. It may become impossible for the Markov chain to reach from one valid redistricting plan to another, which would violate the irreducibility assumption in Theorem 2.

Alternatively, researchers could run Algorithm 1 to generate samples from the uniform distribution and then simply discard any invalid sampled plans post hoc. Unfortunately, this naive strategy, which overcomes the theoretical issues of the previous algorithm, will also discard many sampled plans and hence generate vanishingly few valid plans in a reasonable amount of time.

To overcome the issues with the preceding methods, we modify Algorithm 1 by first choosing the stationary distribution of our Markov chain such that redistricting plans likely to meet the equal population constraint are oversampled. The idea behind this strategy is to use those invalid partitions that are likely to be proposed by Algorithm 1 as a way for the Markov chain to transition between valid partitions. We use the following Gibbs distribution as our target distribution,

$$f_\beta(\pi) = \frac{1}{z(\beta)} \exp\left(-\beta \sum_{V_\ell \in \pi} \left| \sum_{i \in V_\ell} \frac{p_i}{\bar{p}} - 1 \right|\right), \tag{11}$$

where $\beta \ge 0$ is called the inverse temperature and $z(\beta)$ is the normalizing constant. Thus, $f_\beta(\pi)$ has peaks around plans that meet the equal population constraint.

To sample from this distribution, we modify the acceptance probability in Equation (8) by replacing $g(\pi')/g(\pi)$ with $g_\beta(\pi')/g_\beta(\pi)$, where g_β is the unnormalized form of the Gibbs distribution, that is, $g_\beta(\pi) \propto f_\beta(\pi)$. Fortunately, once this ratio is computed for the initial state, we can compute the ratio for subsequent states by only computing district population changes due to reassignment of each $C \in \mathbf{V}_{\mathbf{CP}}$. Thus, the ratio computation has $O(|\mathbf{V}_{\mathbf{CP}}|)$ complexity, which has a negligible impact on the speed of the algorithm.

As the final step, we discard invalid sampled plans that do not satisfy Equation (9) and reweight the remaining sampled plans by $1/g_\beta(\pi)$ to approximate the uniform sampling from the population of valid redistricting plans meeting the constraint. If we resample the sampled plans with replacement using this importance weight, then the procedure is equivalent to the sampling/importance resampling (SIR) algorithm (Rubin 1987). The steps described above are summarized in the following algorithm, which we refer to as the basic algorithm with soft constraint.

Algorithm 1.2 (Sampling contiguous redistricting plans with soft constraint). We initialize the algorithm by obtaining a valid partition $\pi_0 = \{V_1^{(0)}, V_2^{(0)}, \ldots, V_n^{(0)}\}$, where each district $V_\ell^{(0)}$ is contiguous, that is, connected in the graph, and then perform the following steps,

Step 1 (Run the basic algorithm with Gibbs distribution): Generate M samples using Algorithm 1 with edge cut probability q, Gibbs parameter β, and approximated acceptance probability

$$\alpha(\pi', \mathbf{CP} \mid \pi) \approx \min\left(1, \left(\frac{|B(\mathbf{CP}, \pi)|}{|B(\mathbf{CP}, \pi')|}\right)^R \frac{F(|B(\mathbf{CP}, \pi)|)\,(1-q)^{|\mathcal{C}(\pi', \mathbf{V}_{\mathbf{CP}})|}}{F(|B(\mathbf{CP}, \pi')|)\,(1-q)^{|\mathcal{C}(\pi, \mathbf{V}_{\mathbf{CP}})|}} \cdot \frac{g_\beta(\pi')}{g_\beta(\pi)}\right).$$

Step 2 (Reject samples failing to meet constraint): For a specified population constraint δ, identify all samples generated through Step 1 such that

$$\max_{1 \le \ell \le n} \left| \frac{\sum_{i \in V_\ell} p_i}{\bar{p}} - 1 \right| > \delta,$$

and discard them.

Step 3 (Resample using SIR): Draw S samples with replacement from the set resulting from Step 2 using sampling weights $1/g_\beta(\pi)$.

2.5. Improving the Convergence With Tempering

The fundamental challenge of incorporating the equal population constraint through Algorithm 1.2 is that it becomes exceedingly difficult for the Markov chain to *efficiently* traverse from one valid partition to another. On one hand, if the inverse temperature parameter β of f_β is set too high, the Markov chain may never visit certain parts of the target distribution. On the other hand, if the β is set too low, many of the sampled plans will be invalid. We consider simulated (Marinari and Parisi 1992) and parallel tempering (Geyer 1991) with Algorithm 1.2 to aid in efficient exploration of the sample space without compromising the desired stationary distribution. In the applications of this article, we opted to use parallel tempering due to its ease of implementation as well as its robustness to parameter selection. We discuss the details of parallel tempering below and refer the reader to Appendix S4 for more information on simulated tempering (Algorithm S1).

In parallel tempering, r copies of an MCMC algorithm are run at r different temperatures,[2] and after a fixed number of iterations we exchange the corresponding temperatures between two randomly selected adjacent chains using the Metropolis criterion. The idea is that chains at higher temperatures better explore the different parts of the distribution since the low probability states separating one set of valid partitions from another set of valid partitions now have a higher probability. The swapping mechanism allows the chains at lower temperatures to jump between different parts of the target distribution more easily without changing the stationary distribution of the Markov chain.

The nature of parallel tempering suggests that it should be implemented in a parallel architecture to minimize computation time. Altekar et al. (2004) described such an implementation using parallel computing and MPI, which we use as the basis for implementing our algorithm for uniformly sampling redistricting plans meeting a specified population constraint. The algorithm, which we henceforth refer to as the parallel tempering algorithm with soft constraint, is given below.

Algorithm 2 (Sampling contiguous plans with parallel tempering and soft constraint). Given r initial valid partitions $\pi_0^{(0)}, \pi_0^{(1)}, \ldots, \pi_0^{(r-1)}$, a sequence of r increasing inverse temperatures $\beta^{(0)} > \beta^{(1)} > \cdots > \beta^{(r-1)} = 0$ with $\beta^{(0)}$ the target inverse temperature for inference, a swapping interval T, and a population constraint δ, the parallel tempering with soft constraint algorithm performs the following steps,

Step 1 (Generate samples): Generate MT samples by repeating the following steps M times

(a) (Run the basic algorithm with Gibbs distribution): For each chain $i \in \{0, 1, \ldots, r-1\}$, using the current partition $\pi_t^{(i)}$ and the corresponding inverse temperature $\beta^{(i)}$, obtain a valid partition $\pi_{t+T}^{(i)}$ by running T iterations of Algorithm 1 with the following approximate acceptance probability,

$$\alpha(\pi', \mathbf{CP} \mid \pi_t^{(i)}) \approx \min\left(1, \left(\frac{|B(\mathbf{CP}, \pi_t^{(i)})|}{|B(\mathbf{CP}, \pi')|}\right)^R \frac{F(|B(\mathbf{CP}, \pi_t^{(i)})|)\,(1-q)^{|\mathcal{C}(\pi', \mathbf{V_{CP}})|}}{F(|B(\mathbf{CP}, \pi')|)\,(1-q)^{\left|\mathcal{C}(\pi_t^{(i)}, \mathbf{V_{CP}})\right|}} \cdot \frac{g_{\beta^{(j)}}(\pi')}{g_{\beta^{(j)}}(\pi_t^{(i)})}\right). \tag{12}$$

This step is executed concurrently for each chain.

(b) (Propose a temperature exchange between two chains): Randomly select two adjacent chains j and k and exchange information about the temperatures $\beta^{(j)}, \beta^{(k)}$ and the unnormalized likelihoods of the current partitions $g_{\beta^{(j)}}\left(\pi_{t+T}^{(j)}\right), g_{\beta^{(k)}}\left(\pi_{t+T}^{(k)}\right)$ using MPI

(c) (Accept or reject the temperature exchange): Exchange temperatures (i.e., $\beta^{(j)} \leftrightarrows \beta^{(k)}$) with probability $\gamma\left(\beta^{(j)} \leftrightarrows \beta^{(k)}\right)$ where

$$\gamma\left(\beta^{(j)} \leftrightarrows \beta^{(k)}\right) = \min\left(1, \frac{g_{\beta^{(j)}}\left(\pi_{t+T}^{(k)}\right) g_{\beta^{(k)}}\left(\pi_{t+T}^{(j)}\right)}{g_{\beta^{(j)}}\left(\pi_{t+T}^{(j)}\right) g_{\beta^{(k)}}\left(\pi_{t+T}^{(k)}\right)}\right). \tag{13}$$

All previously generated samples are assumed to have been generated at the current temperature of the chain

Step 2 (Reject samples failing to meet constraint): Taking the samples generated under temperature $\beta^{(0)}$, identify all samples such that

$$\max_{1 \le \ell \le n} \left| \frac{\sum_{i \in V_\ell} p_i}{\bar{p}} - 1 \right| > \delta,$$

and discard them.

Step 3 (Resample using SIR): Draw S samples with replacement from the set resulting from Step 2 using sampling weights $1/g_{\beta^{(0)}}(\pi)$.

The mixing performance of Algorithm 2 may be affected by the choice of the temperature sequence $\beta^{(i)}$, the swapping interval T, and the number of chains r. While much work has been done on the choice of temperatures (e.g., Earl and Deem 2005), no sequence has been shown to be best. However, in general, sequences with geometric spacing have been shown to produce reasonable results (see, e.g., Kone and Kofke 2005; Atchadé, Roberts, and Rosenthal 2011). For this reason, we use the sequence $\beta^{(i)} = \left(\beta^{(0)}\right)^{1-\frac{i}{r-1}}, i \in \{0, 1, \ldots, r-1\}$ for our implementation, which we find works well in our applications. The swapping interval should be set high enough to allow for chains to adequately explore the local state space around the modes of the target distribution, but not so high that these regions are oversampled. We use a value of $T = 100$ for our algorithm, which we found to work well in our applications. For

[2] The temperature referenced here is different from the temperature in the expression for f_β. For the sake of simplicity, we absorb the distribution parameter β into the sequence of tempering parameters $\beta^{(0)}, \ldots, \beta^{(r-1)}$.

the number of chains used, if r is too small, then swaps between chains are unlikely and mixing is still poor. If r is too large, too many swaps are required before cooler chains have access to states generated from the hot chains.

2.6. Comparison With Existing Algorithms

A number of researchers use Monte Carlo simulation algorithms to sample possible redistricting plans under various criteria to detect instances of gerrymandering and understand the causes and consequences of redistricting (e.g., Engstrom and Wildgen 1977; O'Loughlin 1982; Cirincione, Darling, and O'Rourke 2000; McCarty, Poole, and Rosenthal 2009; Chen and Rodden 2013). Most of these studies use a similar Monte Carlo simulation algorithm where a geographical unit is randomly selected as a "seed" for each district and then neighboring units are added to contiguously grow this district until it reaches the prespecified population threshold. A representative of such algorithms, proposed by Cirincione, Darling, and O'Rourke (2000) and implemented by Altman and McDonald (2011) in their open-source BARD package, is given here.

Algorithm 3 (The standard redistricting simulator (Cirincione, Darling, and O'Rourke 2000)). For each district, we repeat the following steps.

Step 1: From the set of unassigned units, randomly select the seed unit of the district.

Step 2: Identify all unassigned units adjacent to the district.

Step 3: Randomly select one of the adjacent units and add it to the district.

Step 4: Repeat Steps 2 and 3 until the district reaches the predetermined population threshold.

Additional criteria can be incorporated by modifying Step 3 to select certain units. For example, to improve the compactness of the resulting districts, one may choose an adjacent unassigned unit that falls entirely within the minimum bounding rectangle of the emerging district. Alternatively, an adjacent unassigned unit that is the closest to emerging district can be selected (Chen and Rodden 2013).

Nevertheless, the major problem of these simulation algorithms is their adhoc nature. For example, as the documentation of BARD package warns, the creation of earlier districts may make it impossible to yield contiguous districts. These algorithms also rely on rejection sampling to incorporate constraints, which is an inefficient strategy since many of the samples created are discarded. More importantly, the algorithms come with no theoretical result and are not even designed to uniformly sample redistricting plans. In contrast, the proposed algorithms described in Sections 2.2–2.5 are built upon the well-known theories and strategies about the MCMC methods. Furthermore, although our algorithm also uses rejection sampling to incorporate constraints, the rejected samples are still used to assist in finding valid partitions, which is a more efficient strategy. The disadvantage of our algorithms, however, is that they yield a dependent sample and hence their performance will hinge upon the degree of mixing. Thus, we now turn to the assessment of the empirical performance of the proposed algorithms.

3. Empirical Studies

In this section, we assess the performance of the proposed MCMC algorithms (Algorithms 1, 1.1, 1.2, and 2) in two ways. First, we conduct a small-scale validation study where, due to the size of the tested maps, all valid redistricting plans can be enumerated. We show that the proposed MCMC algorithms can approximate the target distribution more accurately than the standard algorithm (Algorithm 3). The proposed basic algorithms (Algorithms 1.1 and 1.2) also scales much better than the standard algorithm. Second, we apply the proposed parallel tempering algorithm (Algorithm 2) to actual redistricting data from Pennsylvania. We demonstrate that small changes to the actual redistricting map can nearly eliminate partisan bias.

To conduct these analyses, we utilize precinct-level shape files and electoral returns data from the Harvard Election Data Archive to determine precinct adjacency and voting behavior. We supplement this data with basic demographic information from the U.S. Census Bureau P.L. 94-171 summary files, which are disseminated to the 50 states to obtain population parity in decennial redistricting.

3.1. A Small-Scale Validation Study

We conduct a validation study based on the 25 precinct set, which is shown as a graph in Figure 1 (see Fifield, Imai, et al. 2019, for a more comprehensive validation study). We begin by considering the problem of partitioning this graph into three contiguous districts. Even for this small-scale study, the enumeration of all valid partitions is a nontrivial problem (see Appendix S3, which describes our enumeration algorithm). Of the roughly $3^{25}/6 \approx 1.41 \times 10^{11}$ possible partitions, only 117,688 have three contiguous districts, and 3617 have district populations within 20% of parity.

The results of the proposed algorithms are based on a single chain of 10,000 draws while those of the standard algorithm are based on the same number of independent draws. For all simulations, we set q to 0.05 and $\lambda = 2$, where λ is the mean of the truncated Poisson distribution. To implement parallel tempering (Algorithm 2), we specify a sequence of temperatures $\{\beta^{(i)}\}_{i=0}^{r-1}$. For the population deviation of 20%, we chose a target temperature of $\beta^{(r-1)} = 5.4$, and for the population deviation of 10%, we chose a target temperature of $\beta^{(r-1)} = 9$. In both cases, we use $\beta^{(0)} = 0$ and set $r = 10$.

For both analyses, we conducted an initial grid search over possible values of $\beta^{(r-1)}$ and kept track of the acceptance probability of the chain and the number of samples that were drawn within the target level of population parity. We selected a target temperature such that a sufficient number of plans meet the population constraint, and that the acceptance probability is in the recommended 20%–40% range (Geyer and Thompson 1995). As described in Sections 2.4 and 2.5, we use a subset of draws taken under the target temperature, discard those that fall outside the population target, and then resample the remaining draws using the importance weights $1/g_{\beta^{(i)}}(\pi)$.

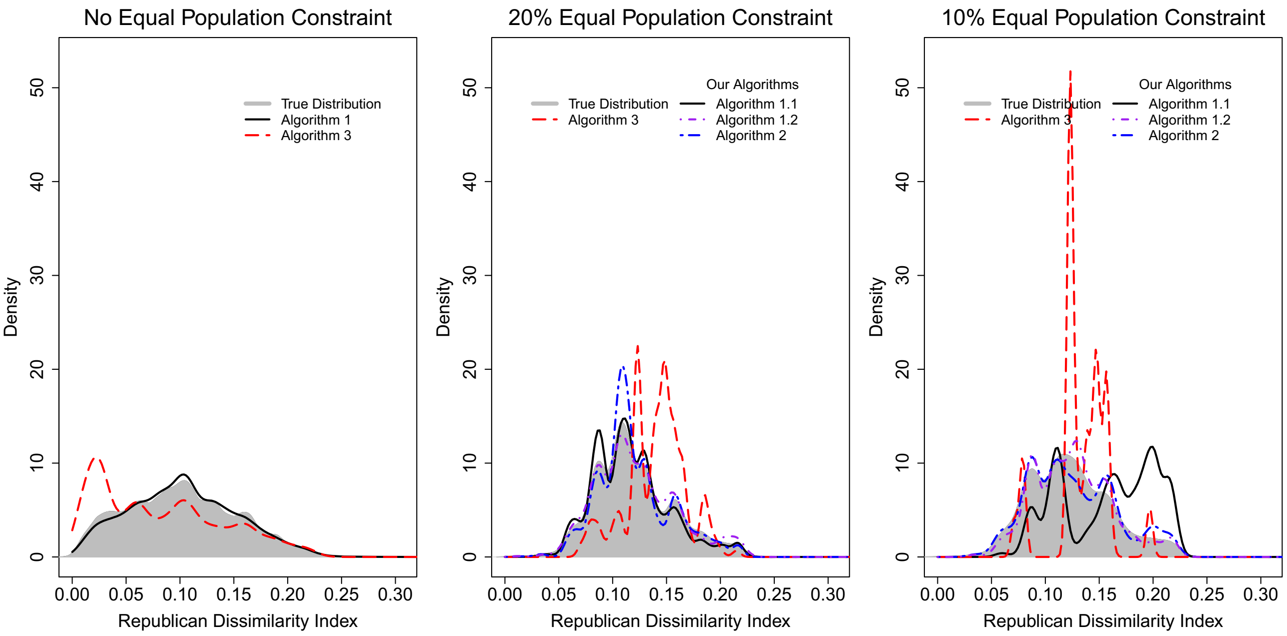

Figure 3. A small-scale validation study with three districts using the Republican Dissimilarity Index. The underlying data are the 25 precinct set shown in the left plot of Figure 1. We find that the proposed algorithm with a hard constraint (Algorithms 1 and 1.1; solid black lines), which discards invalid plans, approximates the true target distribution (gray histograms) well when no (left column) or a moderate (middle column) equal population constraint is imposed. However, the algorithm exhibits poor performance when a stricter equal population constraint (right column) is imposed. The proposed algorithm with a soft constraint (Algorithm 1.2; purple dot-dashed line), which is based on Gibbs distribution, or parallel tempering (Algorithm 2; blue dot-dashed line) perform well even in this case. In contrast, the standard algorithm (Algorithm 3; red dashed lines) as implemented in the `BARD` package fails to approximate the target distribution in all cases and the performance deteriorates as the constraint becomes stricter.

We investigate the degree to which each algorithm can approximate the target distribution using the Republican dissimilarity index, which is a common measure of residential segregation,

$$D = \frac{1}{2}\sum_{\ell=1}^{n}\frac{t_\ell}{T}\cdot\frac{|p_\ell - P|}{P(1-P)}, \tag{14}$$

where ℓ indexes districts in a state, t_ℓ is the population of district ℓ, p_ℓ is the share of district ℓ that identifies as Republican, T is the total population in the state, and P is the share of Republicans across all districts. The dissimilarity index represents the proportion of the minority group that would have to be moved to be randomly distributed across districts within a state (Massey and Denton 1988). We have examined other statistics and the results are, as expected, similar to those presented here.

Figure 3 presents the results. The left column shows that when the equal population constraint is not imposed, the proposed basic algorithm (Algorithm 1; black solid lines) approximates the target distribution well while the standard algorithm (Algorithm 3; red dashed lines), as implemented in the `BARD` package (Altman and McDonald 2011), fails even in this easiest case. In the middle and right columns, we impose the equal population constraint where only up to 20% and 10% deviation from population parity is allowed, respectively. Not surprisingly, the standard algorithm (Algorithm 3; red dashed lines) again completely fails. The proposed algorithm with a hard constraint (Algorithm 1.1; black solid line), which simply discards invalid plans as discussed in Section 2.4, also has difficulty in approximating the target distribution. However, the proposed algorithm with a soft constraint (Algorithm 1.2; purple dot-dashed line), which uses the Gibbs distribution of Equation (11) as discussed in Section 2.4, performs well in both cases. The proposed algorithm with parallel tempering (Algorithm 2; blue dot-dash lines) also performs reasonably well even when a strict equal population constraint is imposed.

3.2. Pennsylvania: Local Simulations

3.2.1. The Setup

Next, we analyze the 2008 election data from Pennsylvania to examine how small changes to the actual congressional map (Figure 4) would affect partisan bias. Pennsylvania represents a challenging application because its 2008 plan had a total of 19 congressional districts and 9256 precincts. Our experience suggests that the proposed algorithm cannot reliably approximate the uniform distribution of all valid redistricting plans under a strong equal population constraint for such a large state. Thus, we conduct "local simulations," in which the goal is to obtain a representative sample of valid redistricting plans within a prespecified degree of deviation from the 2008 redistricting plan. This represents a realistic application because many redistricting cases do not drastically alter the existing plans. In Appendix S6, we conduct "global simulations," in which the target distribution is uniform on all valid redistricting plans under an equal population constraint, using the data from New Hampshire with only two districts and 327 precincts.

We use the proposed algorithm with parallel tempering (Algorithm 2), as described in Section 2.5. To conduct local simulations, we replace Equation (11) with the following,

$$f_\beta(\pi) = \frac{1}{z(\beta)}\exp\left(-\frac{\beta}{19}\sum_{V_\ell\in\pi}\psi(V_\ell)\right), \tag{15}$$

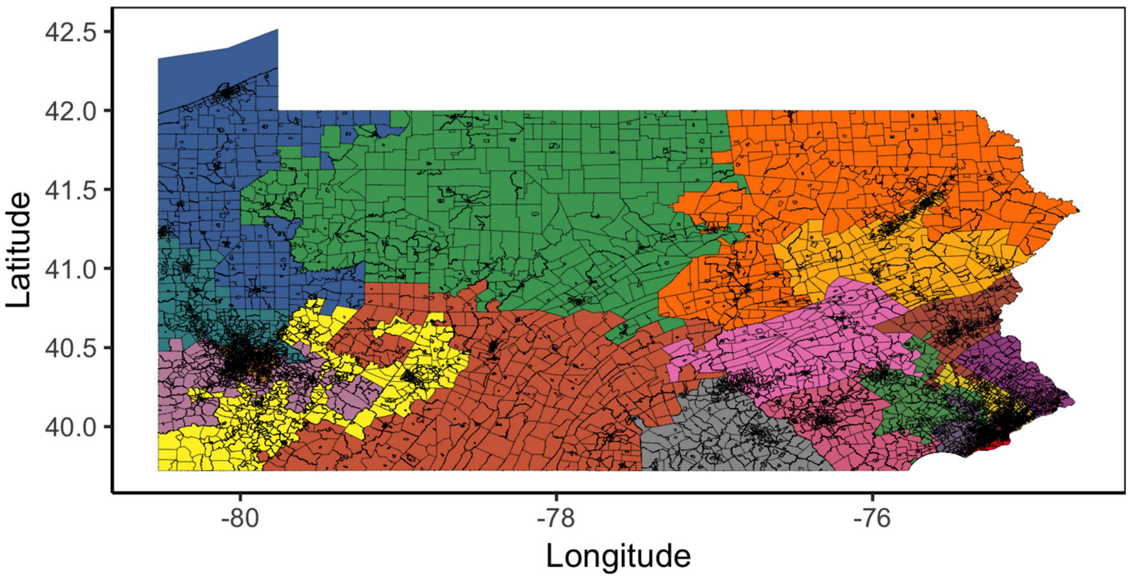

Figure 4. Pennsylvania's 2008 redistricting plan. Pennsylvania has 9256 precincts that were divided into 19 congressional districts in their 2008 redistricting plan.

where $\psi(V_\ell)$ represents the proportion of precincts in district $\ell = 1, 2, \ldots, 19$ that are not shared with the corresponding district of the 2008 redistricting plan. We then wish to approximate the population of valid redistricting plans which satisfies the constraint $\sum_{\ell=1}^{19} \psi(V_\ell)/19 \leq \delta$ for some δ. For example, if we choose $\delta = 0.05$, then we obtain a sample of valid redistricting plans, 5% of whose precincts are switched from other districts of the 2008 plan.

Following the recommendations given in the literature (Kone and Kofke 2005; Atchadé, Roberts, and Rosenthal 2011), we chose $\beta^{(i)}$ to be geometrically spaced, that is, $\beta^{(i)} = \left(\beta^{(0)}\right)^{1-\frac{i}{r-1}}, i \in \{0, 1, \ldots, r-1\}$. After a preliminary grid search over possible values of the target temperature and of R, we set the target temperature to be 2500, $q = 0.04$, and $\lambda = 10$ (where λ is the mean of the truncated Poisson distribution), which led to a good mixing behavior, reasonable acceptance probabilities, and provided a sufficient number of valid plans for subsequent analyses. Since we are only exploring the local neighborhood of the 2008 plan, we initialize the Markov chain three times from the 2008 plan, but using different random seeds so that they start to explore different parts of the target distribution. We run each of the 10 temperature chains for 120,000 simulations for each initialization and thin the chain by 12. We then base inference off of those 10,000 simulations for each of the three initialized MPI chains.

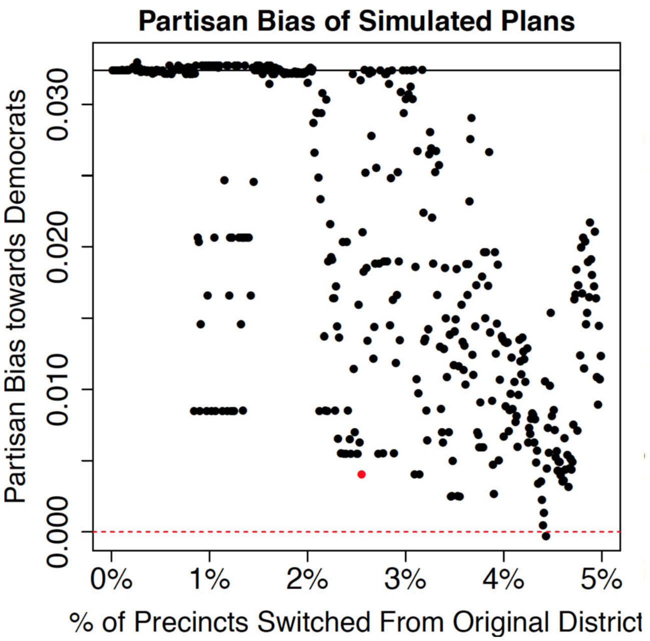

Figure 5. Local simulations for the 2008 Pennsylvania redistricting plan. The plot shows that the average partisan bias of the simulated plans decreases nearly to zero as more precincts are switched out of the original redistricting plan. The solid line represents the partisan bias of the original plan, while the red dashed line indicates an unbiased redistricting plan, in which the seats gained from a vote swing of any value is equal to the number of seats lost in response to a vote swing of equal magnitude in the opposite direction. The solid red circle represents the minimal-bias plan when less than 3% of all precincts are swapped.

3.2.2. Empirical Findings

We examine how the partisan bias changes as the simulated plan moves farther from the 2008 plan. Following the literature (see Grofman and King 2007), we use deviation from partisan symmetry as a measure of partisan bias. This measure evaluates the partisan implications of the counterfactual election outcomes under hypothetical uniform swings of various degrees across districts. Specifically, we consider all vote swings where the statewide 2008 two-party vote shares are between 40% and 60%. These vote shares are then aggregated according to each redistricting plan, yielding the counterfactual two-party vote shares at the district level and the number of hypothetical Democrat and Republican winners of the election. The resulting information is summarized as the seats-votes curve (Tufte 1973), which is a nondecreasing step-function $f(x)$ evaluated within the range of vote shares x from 40% to 60%.

Formally, let $f^*(x)$ be a nondecreasing step-function that is symmetric around the 50%–50% two-party vote share. Then, the partisan bias is defined as the standardized difference between the area below $f(x)$ and the area below $f^*(x)$,

$$\text{partisan bias} = \frac{1}{\eta}\int_{0.5-\eta}^{0.5+\eta}\left(f(x) - f^*(x)\right)dx$$
$$= \frac{1}{\eta}\int_{0.5-\eta}^{0.5+\eta} f(x)dx - 1, \qquad (16)$$

where the area under $f^*(x)$ is always equal to a uniform vote swing η. In our case, $0 \leq \eta \leq 0.1$. This partisan bias measure is scaled so that -1 indicates a maximally Republican-biased plan (where no vote swing of any magnitude favoring Democrats can ever flip a seat away from Republicans), while 1 indicates a maximally Democrat-biased plan (where no vote swing of any magnitude favoring Republicans can ever flip a seat away from Democrats). A bias measure of 0 represents an unbiased plan, where uniform vote swings result in equal seat gains or losses for each party.

Figure 5 presents the analysis where the solid black line indicates the partisan bias and electoral competitiveness of the 2008 plan, and the red dashed line represents an unbiased plan. The 2008 plan is biased in favor of the Democrats, with a standardized partisan bias measure of approximately 0.035. We find that the partisan bias of this plan can be nearly eliminated by swapping approximately 3% of all precincts in Pennsylvania.

We further explore this finding by closely examining the minimally biased plan found by swapping fewer than 3% of precincts in the local simulations (represented by the red solid circle in Figure 5). This specific plan was obtained by swapping 277

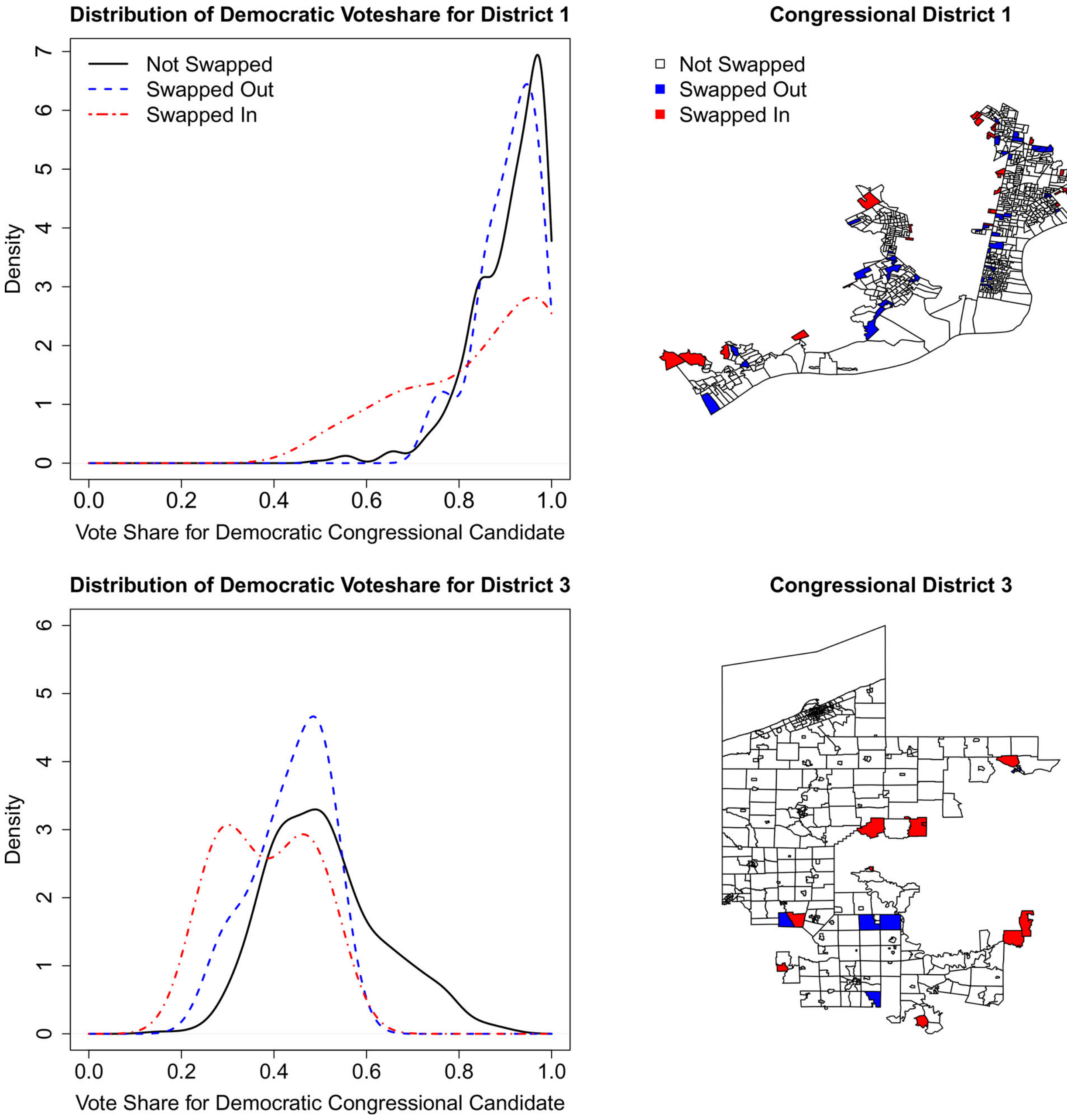

Figure 6. Partisan patterns of the swapped and unswapped precincts in Pennsylvania local simulations. The left column plots the distribution of the Democratic congressional vote-share for the precincts left untouched (black solid line), swapped in (red dot-dashed line), and swapped out (blue dashed line) of a congressional district to obtain a nearly unbiased plan. The right column shows the geography of the swapped and unswapped precincts in Congressional District 1 (top row) and Congressional District 3 (bottom row). In both congressional districts, the algorithm swaps out more partisan precincts for more moderate precincts to achieve more symmetric responses to voter swings.

precincts from the original 2008 map into new districts. Figure 6 plots the distributions of Democratic voteshare in swapped and unswapped precincts in two congressional districts. District 1 (top row) encompasses Central and South Philadelphia, and supported President Obama with nearly 90% of its vote. As a result, Democratic control of the seat is heavily insulated from partisan swings. The minimally biased plan substitutes out 48 precincts with an average Democratic vote share of 90.5% for 34 new precincts from the surrounding suburbs with an average Democratic vote share of 84%. While still heavily Democratic, these changes make the new district more vulnerable to voter swings in the direction of the Republicans.

The bottom row of Figure 6 conducts the same analysis for District 3, located in the northwest corner of Pennsylvania. In 2008, this was the only district in Pennsylvania to flip parties—in the case of this district, from Republicans to Democrats. In a Democrat-biased redistricting plan, Republican-held seats should be more vulnerable to swings. Indeed, Democrats flipped 21 seats in the House of Representatives in 2008. To achieve a more unbiased statewide plan, the proposed algorithm swaps in more conservative precincts (average Democratic vote share of 38.2%) for moderate precincts (average Democratic vote share of 43.5%), thereby making it more resilient to such swings. While the change in total Republican votes (an unbiased plan would have gained approximately 1000 additional Republican votes in District 3) would not have been enough to keep the seat from switching parties, the proposed change would have increased the bar necessary to flip the seat in Democratic-leaning uniform swing scenarios.

4. Concluding Remarks

Over the last half century, a number of automated redistricting algorithms have been proposed in the methodological literature. Most of these algorithms have been designed to find an optimal redistricting plan given a certain set of criteria. However, many researchers have been interested in characterizing the distribution of redistricting plans under various constraints. Unfortunately, few such simulation algorithms exist and even the commonly used ones do not have a theoretical justification.

In this article, we propose an automated redistricting simulator using Markov chain Monte Carlo. Unlike the existing standard algorithm, the proposed algorithms have the theoretical property to approximate a target distribution. In a small-scale validation study, we show that the proposed algorithms work well whereas the standard algorithm fails to obtain a representative sample. Even in more realistic settings where the computational challenge is greater, our analyses show the promising performance of the proposed algorithms. Nevertheless, it is still unclear whether these algorithms scale to even larger states than those considered here.

Future research should consider further improvements of the proposed simulation algorithms, including a strategy to incorporate multiple constraints and a method to generate overdispersed starting values. It is also important to conduct more validation studies so that we can better understand the conditions under which the proposed methodologies fail. An initial such validation study shows some promising results regarding the empirical performance of the MCMC algorithm proposed in this article (see Fifield, Imai, et al. 2019).

Supplementary Materials

In the supplementary materials, we provide proofs of the theorems presented in the article, as well as additional empirical examples.

Acknowledgments

We thank Jowei Chen, Jacob Montgomery, and seminar participants at Chicago Booth, Dartmouth, Duke, Microsoft Research, and SAMSI for useful comments and suggestions. We thank James Lo, Jonathan Olmsted, and Radhika Saksena for their advice on computation. The replication archive for this article is available in Fifield, Higgins, et al. (2019). The open-source R package `redist` for implementing the proposed methodology is available in Fifield, Tarr, and Imai (2015). Replication materials can be found in Dataverse at *https://doi.org/10.7910/DVN/VCIW2I*.

ORCID

Benjamin Fifield http://orcid.org/0000-0002-2247-0201
Michael Higgins http://orcid.org/0000-0003-4969-2733
Kosuke Imai http://orcid.org/0000-0002-2748-1022

References

Altekar, G., Dwarkadas, S., Huelsenbeck, J. P., and Ronquist, F. (2004), “Parallel Metropolis Coupled Markov Chain Monte Carlo for Bayesian Phylogenetic Inference,” *Bioinformatics*, 20, 407–415. [8]

Altman, M. (1997), “The Computational Complexity of Automated Redistricting: Is Automation the Answer,” *Rutgers Computer & Technology Law Journal*, 23, 81–142. [2]

Altman, M., MacDonald, K., and McDonald, M. (2005), “From Crayons to Computers: The Evolution of Computer Use in Redistricting,” *Social Science Computer Review*, 23, 334–346. [1]

Altman, M., and McDonald, M. P. (2011), “BARD: Better Automated Redistricting,” *Journal of Statistical Software*, 42, 1–28. [2,9,10]

Ansolabehere, S., Snyder, J. M., and Stewart, C. (2000), “Old Voters, New Voters, and the Personal Vote: Using Redistricting to Measure the Incumbency Advantage,” *American Journal of Political Science*, 44, 17–34. [1]

Atchadé, Y. F., Roberts, G. O., and Rosenthal, J. S. (2011), “Towards Optimal Scaling of Metropolis-Coupled Markov Chain Monte Carlo,” *Statistics and Computing*, 21, 555–568. [8,11]

Barbu, A., and Zhu, S.-C. (2005), “Generalizing Swendsen-Wang to Sampling Arbitrary Posterior Probabilities,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27, 1239–1253. [2,3,5,6]

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, 12–26. [1]

Chen, J., and Rodden, J. (2013), “Unintentional Gerrymandering: Political Geography and Electoral Bias in Legislatures,” *Quarterly Journal of Political Science*, 8, 239–269. [1,2,9]

Chikina, M., Frieze, A., and Pegden, W. (2017), “Assessing Significance in a Markov Chain Without Mixing,” *Proceedings of the National Academy of Sciences of the United States of America*, 114, 2860–2864. [2]

Chou, C.-I., and Li, S. P. (2006), “Taming the Gerrymander—Statistical Physics Approach to Political Districting Problem,” *Physica A: Statistical Mechanics and its Applications*, 369, 799–808. [1]

Cirincione, C., Darling, T. A., and O’Rourke, T. G. (2000), “Assessing South Carolina’s 1990s Congressional Districting,” *Political Geography*, 19, 189–211. [1,2,9]

DeFord, D., Duchin, M., and Solomon, J. (2019), “Recombination: A Family of Markov Chains for Redistricting,” Tech. Rep., arXiv no. 1911.05725. [2]

Earl, D. J., and Deem, M. W. (2005), "Parallel Tempering: Theory, Applications, and New Perspectives," *Physical Chemistry Chemical Physics*, 7, 3910–3916. [8]

Engstrom, R. L., and Wildgen, J. K. (1977), "Pruning Thorns From the Thicket: An Empirical Test of the Existence of Racial Gerrymandering," *Legislative Studies Quarterly*, 2, 465–479. [9]

Fifield, B., Higgins, M., Imai, K., and Tarr, A. (2014), "A New Automated Redistricting Simulator Using Markov Chain Monte Carlo," Tech. Rep., Department of Politics, Princeton University. [1,2]

——— (2019), "Replication Data for: Automated Redistricting Simulation Using Markov Chain Monte Carlo," available at *https://doi.org/10.7910/DVN/VCIW2I*. [13]

Fifield, B., Imai, K., Kawahara, J., and Kenny, C. T. (2019), "The Essential Role of Empirical Validation in Legislative Redistricting Simulation," Tech. Rep., Department of Government and Department of Statistics, Harvard University, available at *https://imai.fas.harvard.edu/research/files/enumerate.pdf*. [9,13]

Fifield, B., Tarr, A., and Imai, K. (2015), "redist: Markov Chain Monte Carlo Methods for Redistricting Simulation," Comprehensive R Archive Network (CRAN), available at *https://CRAN.R-project.org/package=redist*. [2,13]

Fryer, R., and Holden, R. (2011), "Measuring the Compactness of Political Districting Plans," *Journal of Law and Economics*, 54, 493–535. [1]

Geyer, C. J. (1991), "Markov Chain Monte Carlo Maximum Likelihood," *Interface Foundation of North America*, Retrieved From the University of Minnesota Digital Conservancy, available at *http://hdl.handle.net/11299/58440*. [8]

Geyer, C. J., and Thompson, E. A. (1995), "Annealing Markov Chain Monte Carlo With Applications to Ancestral Inference," *Journal of the American Statistical Association*, 90, 909–920. [2,9]

Grofman, B., and King, G. (2007), "The Future of Partisan Symmetry as a Judicial Test for Partisan Gerrymandering After *Lulac v. Perry*," *Election Law Journal*, 6, 2–35. [1,11]

Hastings, W. K. (1970), "Monte Carlo sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109. [3]

Herschlag, G., Ravier, R., and Mattingly, J. C. (2017), "Evaluating Partisan Gerrymandering in Wisconsin," Tech. Rep., Department of Mathematics, Duke University. [2]

Kone, A., and Kofke, D. A. (2005), "Selection of Temperature Intervals for Parallel-Tempering Simulations," *The Journal of Chemical Physics*, 122, 206101. [8,11]

Liu, Y. Y., Tam Cho, W. K., and Wang, S. (2016), "PEAR: A Massively Parallel Evolutionary Computation Approach for Political Redistricting Optimization and Analysis," *Swarm and Evolutionary Computation*, 30, 78–92. [1]

Marinari, E., and Parisi, G. (1992), "Simulated Tempering: A New Monte Carlo Scheme," *Europhysics Letters*, 19, 451–458. [2,8]

Massey, D., and Denton, N. (1988), "The Dimensions of Racial Segregation," *Social Forces*, 67, 281–315. [10]

Mattingly, J. C., and Vaughn, C. (2014), "Redistricting and the Will of the People," Tech. Rep., Department of Mathematics, Duke University. [2]

McCarty, N., Poole, K. T., and Rosenthal, H. (2009), "Does Gerrymandering Cause Polarization?," *Amerian Journal of Political Science*, 53, 666–680. [1,9]

Mehrotra, A., Johnson, E., and Nemhauser, G. L. (1998), "An Optimization Based Heuristic for Political Districting," *Management Science*, 44, 1100–1114. [2]

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1092. [3]

Nagel, S. S. (1965), "Simplified Bipartisan Computer Redistricting," *Stanford Law Journal*, 17, 863–899. [1]

Niemi, R. G., Grofman, B., Carlucci, C., and Hofeller, T. (1990), "Measuring Compactness and the Role of a Compactness Standard in a Test for Partisan and Racial Gerrymandering," *Journal of Politics*, 52, 1155–1181. [1]

O'Loughlin, J. (1982), "The Identification and Evaluation of Racial Gerrymandering," *Annals of the Association of American Geographers*, 72, 165–184. [9]

Rubin, D. B. (1987), "Comment: A Noniterative Sampling/Importance Resampling Alternative to the Data Augmentation Algorithm for Creating a Few Imputation When Fractions of Missing Information Are Modest: The SIR Algorithm," *Journal of the American Statistical Association*, 82, 543–546. [7]

Swendsen, R. H., and Wang, J. S. (1987), "Nonuniversal Critical Dynamics in Monte Carlo Simulations," *Physical Review Letters*, 58, 86–88. [2,3]

Tam Cho, W., and Liu, Y. (2016), "Toward a Talismanic Redistricting Tool: A Computational Method for Identifying Extreme Redistricting Plans," *Election Law Journal*, 15, 351–366. [1]

Tufte, E. R. (1973), "The Relationship Between Seats and Votes in Two-Party Systems," *American Political Science Review*, 67, 540–554. [11]

Vickrey, W. (1961), "On the Prevention of Gerrymandering," *Political Science Quarterly*, 76, 105–110. [1]

Weaver, J. B., and Hess, S. W. (1963), "A Procedure for Nonpartisan Districting: Development of Computer Techniques," *Yale Law Journal*, 73, 288–308. [1]