

Fairmandering: A column generation heuristic for fairness-optimized political districting*

Wes Gurnee[†]

David B. Shmoys[‡]

Abstract

The American winner-take-all congressional district system empowers politicians to engineer electoral outcomes by manipulating district boundaries. Existing computational solutions mostly focus on drawing unbiased maps by ignoring political and demographic input, and instead simply optimize for compactness. We claim that this is a flawed approach because compactness and fairness are orthogonal qualities, and introduce a scalable two-stage method to explicitly optimize for arbitrary piecewise-linear definitions of fairness. The first stage is a randomized divide-and-conquer column generation heuristic which produces an exponential number of distinct district plans by exploiting the compositional structure of graph partitioning problems. This district ensemble forms the input to a master selection problem to choose the districts to include in the final plan. Our decoupled design allows for unprecedented flexibility in defining fairness-aligned objective functions. The pipeline is arbitrarily parallelizable, is flexible to support additional redistricting constraints, and can be applied to a wide array of other regionalization problems. In the largest ever ensemble study of congressional districts, we use our method to understand the range of possible expected outcomes and the implications of this range on potential definitions of fairness.

1 Introduction

Section 4, Article 1: “The Times, Places and Manner of holding Elections for Senators and Representatives, shall be prescribed in each State by the Legislature thereof; but the Congress may at any time by Law make or alter such Regulations.” Other than this brief statement regarding regulatory authority, the Constitution of the United States offers little guidance on the process by which the members of the House of Representatives are elected. The current district-based system, where districts are most often drawn by state legislatures, enable self-interested and self-preserving politicians to secure a partisan advantage, suppress the vote of minority groups, and protect incumbents from competition.

Such practices, broadly known as gerrymandering, enable large discrepancies between the popular vote and political representation, and ensure that electoral outcomes are unresponsive to swings in public opinion. In 2021, 428 congressional, 1938 state senate, and 4826 state house districts will be redrawn, cementing the partisan power balance in America for the next decade.

Given the mathematical richness and societal importance of redistricting, there is a large literature bridging computational and political sciences on the causes [13, 38], consequences [16, 26], and potential remedies of gerrymandering [30, 3, 42]. Until recently, nearly all proposed computational solutions to create more equitable maps focused on creating maximally compact maps drawn without any political data as input [42, 19, 15, 29, 28, 24, 34, 33, 8, 22]. This is in part due to the computational hardness of optimizing for fairness at scale; it is also due to the difficulty of acquiring granular and historical political data, and because some states explicitly ban the use of political data in the redistricting process [23]. While it is true that compactness-optimized maps are blind to partisan bias, it is not true these maps are free from partisan advantage. The two most successful lines of work that begin to address this gap in the literature are multi-level optimization algorithms [40] and ensemble methods, most notably recombination Markov chains [9].

Exact optimization methods are intractable for all but the smallest problems because optimal political districting is NP -hard for any useful objective function [31, 21, 5]. Optimization approaches for fairness therefore either rely on local search techniques [35, 20] or embed a heuristic in some stage of the optimization. The most successful formulation thus far does both by iteratively coarsening the problem input, solving to optimality for multiple objectives, and then uncoarsening with a local search routine at each level of granularity [40]. However, the number of districts in the problem instance limits the achievable coarsening and therefore such a scheme is unable to scale to large states or state legislative boundaries where there are a much greater number of district seats.

Ensemble methods [25, 6, 9] generate a large en-

*Supported in part by NSF grants CCF-1522054, CCF-1526067, and CCF-1740822, DMS-1839346, and CNS-1952063.

[†]Cornell University - rwg97@cornell.edu

[‡]Cornell University - david.shmoys@cornell.edu

semble of random maps to infer a property about the distribution of all feasible district plans. These methods are ideally suited for detecting gerrymandering [18, 10, 7] via outlier analysis with respect to the random ensemble of partisan blind plans and evaluating the effect of proposed rules [4] on the space of legal plans. However, because these techniques generate one map at a time (and are not optimizing for any property), they are not efficient at sampling from the tails of any distribution. This is problematic because extreme solutions, such as the most unfair solution under a new set of rules or the fairest possible map which maximizes other desirable qualities, are often the solutions we are looking for.

Our work is novel in that it combines these strands of literature and inherits the strengths of both. We introduce a new algorithm which creates an exponential ensemble of distinct district plans by exploiting the natural combinatorial compositionality of redistricting by generating an expressive set of districts that is also efficient to optimize over. The ensemble is exponential in k , the number of seats (districts) in a plan, making our approach ideally suited for larger states and state assembly districts. While these district plans are not appropriate for assessing certain statistical relationships given their lack of independence, they are nonetheless useful for understanding the range of possibilities that a state’s geography and rules allow for. Additionally, by decoupling the generation of districts from the optimization of districts, our formulation enables a more flexible range of objective functions that we can use to optimize for arbitrary mappings from votes to seats, supporting a large family of fairness objectives.

Our core claim is that mapmakers should explicitly optimize for fairness rather than using questionable proxies like compactness. The main contribution of this paper is to introduce a new methodology to achieve this, while being extensible to a wide variety of other regionalization problems [12], particularly those where more complicated objectives outside of compactness are important. Our approach has a number of useful properties including parallelizability and scalability, making huge problem instances tractable, reusability of generated districts and flexibility of objectives, enabling rapid iteration of many definitions of fairness, as well as a modular and transparent execution that facilitates certain human-in-the-loop interactions. These properties allow us to solve problems at unprecedented scale and immediately provide value to independent commissions and other redistricting authorities to quickly and cheaply generate a large volume of high quality plans. In addition to detailed empirical results of different parameter configurations of our algorithm, we use its ensemble

capabilities to perform the largest ever ensemble study to understand the range of possible expected electoral outcomes of all multi-district states and the implications of this range on potential definitions of fairness.

2 Methodology

2.1 Preliminaries

The Political Districting Problem (PDP)

The basic input to the PDP is a set of geographic polygons and their respective populations. Each polygon is represented as a node in a planar graph $G = (B, E)$ where B is the set of atomic geographic blocks used to construct the districts and $(i, j) \in E$ whenever b_i and b_j are geographically adjacent. We let p_i represent the population of b_i and let d_{ij} be the Euclidean distance between the geographic centroids of b_i and b_j . In this setting, the atomic blocks are typically precincts, census tracts, census blocks, or an aggregation thereof. See Figure 1 for an example.

A feasible solution to the PDP for a state with k districts is a k -partition of $B = D_1 \cup \dots \cup D_k$ (and $D_i \cap D_j = \emptyset$, for each pair $i, j \in \{1, \dots, k\}, i \neq j$) such that each district D_j is population balanced, geographically contiguous, and compact. That is, each district D_i must have roughly equal population, the subgraph of G with nodes in D_i must be one fully connected component, and the district should have a geographic shape resembling a fairly simple convex polygon. A solution is called a districting or a district plan. Part of the challenge of this application domain is that these constraints are usually under-specified, with each state offering different levels of specificity and few national standards to fill the gaps. For Congressional districts, the Supreme Court ruled that the districts must be balanced “as nearly as is practicable” [44] but state legislature districts only require “substantial equality of population” [32]. In practice this means adding the constraint that for each district D_j

$$\hat{p}(1 - \epsilon_p) \leq \sum_{i \in D_j} p_i \leq \hat{p}(1 + \epsilon_p), \quad j = 1, \dots, k$$

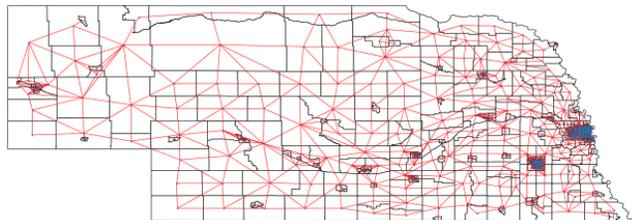


Figure 1: Adjacency graph of Nebraska’s 532 census tracts.

where $\hat{p} = \sum p_i/k$ is the ideal district population, with $\epsilon_p < .01$ for Congressional districts, and $\epsilon_p < .05$ for state legislatures.

There are many proposed mathematical measures of compactness in the redistricting literature [45], but each has particular shortcomings, and none are universally agreed upon. Contiguity is also not entirely unambiguous, because of water features and states that have multiple connected components (e.g., Michigan). Any ambiguity must eventually be resolved in the adjacency graph G by the practitioner.

Column Generation Most integer programming formulations for solving the PDP, and regionalization methods more broadly, choose a set of blocks that act as centers of a region and assign every block to exactly one center, all in one step. They associate decision variables for each pair of nodes [33] for the assignment of blocks to centers and an additional variable for each block to indicate whether or not it is a center. In this design paradigm, the generation of districts and the optimization are performed jointly.

Our approach is based on the idea that we can decouple the district generation from optimization by enumerating feasible districts as an input to the final optimization problem [15, 29, 28]. An optimal solution could theoretically be found by enumerating every legal district D_1, \dots, D_N and solving a set partitioning problem to select the optimal k districts using binary decision variables [15]. Formally, if $A = (a_{ij})$ is the block-district matrix where

$$a_{ij} = \begin{cases} 1 & \text{if } b_i \in D_j \quad (\text{block } i \text{ is in district } j) \\ 0 & \text{else} \end{cases}$$

and x is a binary decision vector where x_j indicates whether or not D_j is in the final plan, the set of feasible solutions is

$$F = \{x : Ax = 1, \|x\|_1 = k, x \in \mathbb{Z}_2^N\}.$$

An optimal plan corresponds to the $\hat{x} \in F$ that minimizes (or maximizes) some linear objective cx . While viable for small problem sizes ($|B| < 100$), the number of feasible districts grows exponentially and quickly becomes intractable to enumerate. However, every feasible district plan must have exactly k non-zero variables, implying that the vast majority of the column space is not useful, and therefore unnecessary to generate.

Implicit column generation is a well-studied technique to efficiently solve huge set partitioning problems [2] such as the PDP. These algorithms involve iteratively solving a relaxed master problem and transferring dual information to a column generation subproblem to determine if there are columns with negative reduced costs

that should be included in the master problem [28]. However, we found when we scaled this approach to larger problem instances, the degeneracy of the master problem rendered the dual values meaningless, resulting in the generation of low quality columns. The core issue with traditional one-at-a-time column generation is that a single arbitrary district is unlikely to have $k - 1$ other districts with which it can be composed to exactly cover the whole state. This observation motivates our heuristic that generates explicitly complimentary columns [17] that can be composed to express a huge number of legal plans.

In addition to the degeneracy of the master problem, there are a number of features of fairness-optimized political redistricting that make “optimal” column generation the wrong approach. When optimizing for expected electoral outcomes, there is significant uncertainty in the objective function. Furthermore, there are many desirable and undesirable properties of feasible maps that should be balanced and analyzed in aggregate. Lastly, the difficulty of solving the master set partitioning problem scales with the number of districts (columns); however, depending on the construction of the districts, the feasible set of plans F , can scale much faster than the set of districts D . For these reasons, instead of focusing on certifying that our columns are “optimal,” a better goal is to generate *efficient* columns, those that maximize $|F|/|D|$.

2.2 Stochastic Hierarchical Partitioning (SHP)

Sample Tree What is the fastest way to generate one trillion unique district plans for a state with 16 districts? One way is to first partition the state into two and generate one million district plans for each half each with 8 districts. The fastest way to generate two sets of one million district plans? Split each half into quarters and generate one thousand district plans for each quarter. And to generate a thousand district plans for each quarter? The attentive reader guesses generating 32 plans for each eighth; partition each eighth 32 times and return.

By construction, the above procedure admits over one trillion district plans while generating only 512 districts by solving only 263 partitioning problems. In practice, none of the trillion plans might be any good because they are all similar, in that they respect the structure imposed by nodes higher in the tree. Therefore, in our method, rather than select a single split, we sample each level repeatedly; this exponential leverage of composing hierarchical partitions is the key property we exploit to generate efficient columns.

To formalize this notion, we introduce the concept of a SHP sample tree, a tree of compact and contiguous

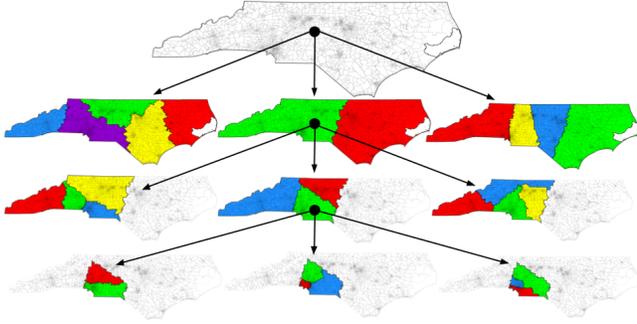


Figure 2: One branch of a sample tree with fan-out width $w = 3$ for North Carolina.

regions that maintain a compositional compatibility invariant (see Figure 2 for an example branch of a sample tree). A sample tree node (R, s) is a collection of contiguous blocks $R \subseteq B$ such that the population of region R is sufficient for the construction of s districts that respect the population constraint. We call s the *capacity* of the region. The root node of the sample tree is (B, k) . The compatibility invariant requires that for any node (R, s) , a valid district plan with s districts can be constructed by combining any valid district plan from each of its children from a single sampled partition. Leaf nodes have $s = 1$; they constitute the set of generated districts and correspond to the columns of the block-district matrix A .

The shape of the tree dictates the properties of the column space. The two main parameters that we use to control the shape are the sample fan-out width w , and the split size z . The former is the number of partitions we sample at each node, and the latter is the size of the partition (binary, ternary, etc.). Parameters w and z control the inherent trade-off in the efficiency and diversity of the columns generated. Sampling more at higher nodes in the tree increases diversity but decreases efficiency. Sampling more at lower nodes has the opposite effect. Increasing the average split size z leads to a shorter tree which yields less exponential leverage but more diversity.

We build the tree by sampling multiple random partitions of each node starting with the root. For computational reasons, we implement this iteratively, rather than recursively, using a queue initialized with the root (B, k) . We continue to partition regions into smaller regions and add those regions to the queue unless $s = 1$, in which case we add this region to the set of generated districts. The *root partition* imposes significant structure on the remaining partitions, so in practice, for sake of diversity, we typically sample hundreds or thousands of root partitions; in contrast,

each nonleaf child node is sampled only a few times (between 2 and 10). The problem then becomes creating an expressive enough column space by sampling a diverse enough set of region partitions.

Partition As depicted in Figure 3, each random partition of a node follows multiple steps: sample the split size, sample the capacity of each center, sample the position of the centers, and finally solve the partitioning integer program to perform the final assignment of blocks to regions.

In more detail, for a node (R, s) , encoding a region $R \subseteq B$ with capacity for s districts, we first sample the split size $z \in [z_{min}, \min(s, z_{max})]$ corresponding to the target number of children and then uniformly at random sample the capacity of each child node s_1, \dots, s_z such that $\sum_i s_i = s$. To prevent significant imbalance in the tree, we usually add a constraint on the maximum difference between the maximum and minimum subregion capacity. One could imagine also tuning z_{min} and z_{max} depending on the region capacity s and the overall size of the state k , but for simplicity, in our experiments we simply set $z_{min} = 2$ and $z_{max} = 5$; this works well in practice.

The next step is to sample the set of centers $C \subset R$ that will act as the seeds for each region of the partition. We detail a number of ways we do this in Appendix B, but the goal is to let the randomization enable a selection of diverse centers that are well-spaced enough to not sacrifice feasibility or create contorted and noncompact regions. Each center is then matched to a size s_i based on an idealized capacity given by the region's Voronoi diagram with respect to the set of centers. See Appendices B and D for explanations of different center selection and capacity matching methods with detailed empirical results for each proposed method.

Finally, using these centers and capacities, we solve an integer program to assign blocks to the z subregions, each centered at a center c_i with capacity s_i that maximize compactness and satisfy the population balance and contiguity constraints. Enforcing contiguity is one of the reasons political districting problems are so difficult, since it requires an exponential number of constraints to do exactly. Instead of explicitly enforcing contiguity, we use an approach first presented in [28] that enforces each partition be a sub-tree of the shortest path tree rooted at the district center. This requires computing the shortest path distance from each center to all nodes in the subgraph of G with nodes in region R , denoted G_R with edgeset E_R . Let d_{ij}^R be the shortest path distance between b_i and b_j in the subgraph G_R . For a center i and block j , let

$$S_{ij} = \{\ell : (\ell, j) \in E_R, d_{i\ell}^R < d_{ij}^R\};$$

for a block j to be assigned to a center i , then one element of S_{ij} , the set of neighbors of j closer to i , must also be assigned to center i .

Formally, the partition integer program (PIP) with inputs R and C is as follows:

$$\begin{aligned}
(2.1) \quad & \text{minimize} \quad \sum_{i \in C} \sum_{j \in R} (d_{ij})^\alpha p_j x_{ij} \\
(2.2) \quad & \text{s.t.} \quad \sum_{i \in C} x_{ij} = 1, \quad \forall j \in R \\
(2.3) \quad & \sum_{j \in R} p_j x_{ij} \leq \hat{p}(s_i + \epsilon_p) \quad \forall i \in C \\
(2.4) \quad & \sum_{j \in R} p_j x_{ij} \geq \hat{p}(s_i - \epsilon_p) \quad \forall i \in C \\
(2.5) \quad & \sum_{\ell \in S_{ij}} x_{i\ell} \geq x_{ij}, \quad \forall i \in C, \quad \forall j \in R \\
(2.6) \quad & x_{ij} \in \{0, 1\}, \quad \forall i \in C, \quad \forall j \in R
\end{aligned}$$

where constraint (2.2) enforces that each block is assigned to exactly one subregion, constraints (2.3) and (2.4) enforce population balance, and constraint (2.5) enforces contiguity. The objective function is based on the dispersion, or moment of inertia measure of compactness [45]. We apply a random exponent to the distance cost term drawn uniformly at $\alpha \in [1, 2]$ to promote more diverse region shapes. For states with abnormal geometric features (e.g. those with multiple connected components or with highly nonconvex shapes such as NY, MA, and MI), it is preferable to optimize for network distance rather than Euclidean distance. We choose this method of enforcing contiguity because it is efficiently computable, but it is not without its drawbacks. While it prevents the formation of noncompact districts, it also discounts many districts that would pass an eyeball test and generally reduces the space of feasible solutions. One item of future work is to profile different methods of enforcing contiguity [41].

The capacitated partition program shares many similarities with the capacitated transportation problem, which is solvable in polynomial time with flow networks [14]. However, instead of assigning units of supply to units of demand, we are assigning units of population to districts with population requirements. We include the additional contiguity constraint [5], but given that most compact solutions are contiguous, this does not drastically increase the difficulty. For a region with n nodes and z centers, the partition IP has nz variables and $n + 2z + nz$ constraints. These problems rarely take more than a few seconds to construct and solve.

Tree Complexity The first key property of our approach is the leverage we gain from our tree formula-

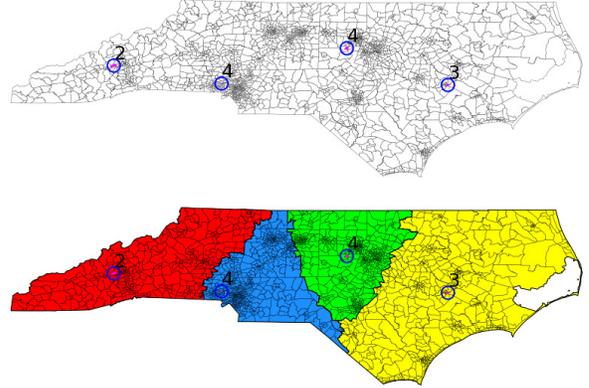


Figure 3: An example partition step of the sample tree root node with split size $z = 4$. First, we sample center population capacities and their geographic locations. Second, we solve the partition integer program to partition the region into contiguous subregions of appropriate population.

tion: districts with shared ancestors are always compatible. With a deeper tree, the number of feasible district plans increases rapidly. Importantly, this makes our method more effective when k is larger, whereas most optimization approaches become intractable with large k .

Let $P(R, s)$ denote the number of feasible s -district plans of region R given by the sample tree. If $w(R)$ gives the sample width for region R , and $z(i)$ gives the number of subregions for sample i , the number of plans has a recursive formula

$$P(R, s) = \begin{cases} \sum_{i=1}^{w(R)} \prod_{j=1}^{z(i)} P(R_{ij}, s_j) & \text{for } s > 1 \\ 1 & \text{for } s = 1 \end{cases}$$

where R_{ij} is subregion j of sample i . Intuitively this means summing over sibling samples and multiplying over children.

Theorem 1. Consider a set of blocks B to be partitioned into k districts. For a sample tree with root node (B, k) and with nodes corresponding to distinct partitions, with constant sample width w , and arbitrary split sizes $z' \in [2, z]$, the tree admits $P(B, k)$ total distinct partitions where

$$w^{\frac{k-1}{z-1}} \leq P(B, k) \leq w^{k-1}.$$

The theorem follows from an inductive counting argument (see Appendix C). Note that the upper bound is tight only when we sample binary partitions (regardless of balance) and the lower bound is tight when we only sample z -partitions and $k = z^n$ for some integer n .

In practice, we have nodes that do not successfully find w samples and generate duplicate regions. Duplicate leaves decrease the number of distinct district plans, whereas duplicate internal nodes can increase the number because they effectively increase the sample width.

To successfully achieve the full w^{k-1} distinct plans for $k = 2^n$, one must solve $w \sum_{d=0}^{n-1} (2w)^d$ partition problems involving center selection and the PIP. This is less work than it appears because as d increases, the size of the PIP decreases exponentially. In particular, where the root node has to partition $|B|$ blocks, a node at depth d will only have to partition $|B|/2^d$ blocks. Again using the binary case, this process will generate $(2w)^{\log_2(k)}/2$ districts which admit $w^{(k-1)}$ unique district plans. This exponential ratio between the number of districts and plans is the targeted column efficiency property.

We call the log of the ratio between the number of generated plans and districts, $\log\left(\frac{|P|}{|D|}\right)$, the *leverage* of the ensemble. To our knowledge, all existing ensemble approaches actually have negative leverage; that is, they generate more districts than plans. For instance, techniques that generate a fully new plan each iteration (such as greedy agglomerative methods [6]) have leverage $\ell = \log\left(\frac{1}{k}\right)$ and Markov chain techniques [7, 9] generate two districts per plan, $\ell = \log\left(\frac{1}{2}\right)$. Especially for large k , having low leverage implies that a technique spends significant computational effort generating districts but that effort does not translate into an efficient exploration of the feasible space.

2.3 Optimization

Objective Function The second key property of our decoupled design is the flexibility it grants in specifying the objective function. Whereas standard approaches for one-step optimization require the objective be a piecewise-linear function of blocks, our two-step method enables piecewise-linear functions of district-level metrics that can be arbitrary functions of blocks. This property enables our fairness objective. A fair district plan, in our view, is one that, *on average*, calibrates electoral outcomes with the partisan affiliation of a state to minimize the number of *wasted* votes.

A wasted vote is a vote cast for a losing candidate or a surplus vote cast for the winner over what was needed to win. The efficiency gap (EG) [39], a popular fairness metric, measures the difference between wasted votes for the two parties. The *responsiveness* of a plan is the number r such that a 1% increase in a party's vote-share is accompanied by a $r\%$ increase in a party's seat-share. Assuming uniform turnout, minimizing the EG coincides with finding a plan with a constant responsiveness of 2 [43] (i.e., a 1% increase in vote-

share should always yield a 2% average increase in seat-share). This is not always possible and scholars debate what the ideal responsiveness should be [27]. However, our method is flexible enough to support optimizing any function of votes to seats so we can accommodate alternative definitions of fairness.

To make such probabilistic statements, we require a probabilistic model of partisan affiliation as an input. We use historical precinct returns from statewide elections to estimate the mean and standard deviation of the two-party vote-share for an arbitrary district (see Appendix A for the details of our data sources and preparation). Let $\hat{\nu}$ be the average statewide partisan vote-share, ν_i be a random variable for the partisan vote-share of district i , and ψ_i be a Bernoulli random variable indicating an electoral win or loss for district i . We assume

$$\nu_i \sim \mathcal{N}(\mu_i, \sigma_i^2) \quad \psi_i \sim \mathcal{B}(P(\nu_i > .5))$$

where μ_i and σ_i^2 are given by the historical statewide returns of blocks in district i . By linearity of expectation, the expected difference between the statewide seat-share and statewide vote-share is the sum of the differences between the expected district-level seat-share and statewide vote-share:

$$E \left[\frac{1}{k} \sum_{i=1}^k \hat{\nu} - \psi_i \right] = \frac{1}{k} \sum_{i=1}^k \hat{\nu} - \left(1 - \Phi \left(\frac{\mu_i - .5}{\sigma_i} \right) \right),$$

where Φ is the unit Gaussian cumulative distribution function (CDF). Given the infrequency of elections, in our experiments we use a t random variable CDF, but the model is flexible to support other affiliation models with richer features. Importantly, we can optimize for $h(\hat{\nu}) - \psi_i$ where h gives a desired mapping from votes to seats enabling optimizing to the shape of arbitrary seat-vote curves. For the efficiency gap, $h(\hat{\nu}) = 2\hat{\nu} - .5$, assuming uniform turnout.

Our model is simple by design, making estimation and optimization tractable for millions of arbitrary districts. We do not claim that this is the right model, and in many cases it is the wrong model when incumbency or other specific information is known for a district. However, given the application domain, we believe simplicity is important for adoption, since a more opaque or harder to reason about model would likely be politically more controversial. Regardless, a promising direction for future work is developing more robust affiliation models suitable for use in optimization settings that can generalize to multiple parties or different voting paradigms.

Tree-based Dynamic Programming When the following conditions hold, the sample tree admits an efficient dynamic programming solution to find the optimal objective.

- C1. There are no additional constraints on the composition of districts (e.g., maintaining a specified number of competitive or majority-minority districts) other than those implicitly encoded by the structure of the tree.
- C2. The objective function is linear (e.g., compactness, competitiveness, but not seat-vote difference)

The intuition underlying the dynamic program is that for a particular partition of a region, the optimal objective (solution) is the sum (composition) of the optimal objectives (solutions) for each of the subregions in the partition. Written recursively, the optimal objective value of region R with capacity s is

$$f^*(R, s) = \begin{cases} \operatorname{argmin}_{i \in w(R)} \sum_{j=1}^{z(i)} f^*(R_{ij}, s_{ij}) & \text{for } s > 1 \\ c_{ij} & \text{for } s = 1 \end{cases}$$

where c_{ij} is the cost coefficient of the district R_{ij} , $z(i)$ is the split size of region partition i , and $w(R)$ is the number of partitions sampled.

While this approach is not typically helpful for finding a specific *fair* plan, it is convenient for quickly evaluating the range of the most extreme solutions generated for arbitrary linear metrics. This is especially useful in ensemble studies to evaluate how different inputs or constraints effect the range of possible partisan outcomes.

Master Selection Problem When either condition C1 or C2 above does not hold (most notably when some definition of fairness is the objective), we use the machinery of integer programming to find a family of solutions. This is accomplished by solving a set partitioning master selection problem (MSP) to choose the final set of districts. We solve a separate MSP for each set of leaf nodes that share a parent root partition. In this standard formulation [28], we have a binary decision variable for every generated district x_j (indexed by D) and wish to

$$(2.1) \quad \text{minimize} \quad \sum_{j \in D} c_j x_j$$

$$(2.2) \quad \text{s.t.} \quad \sum_{j \in D} a_{ij} x_j = 1, \quad \forall i \in B;$$

$$(2.3) \quad \sum_{j \in D} x_j = k;$$

$$(2.4) \quad x_j \in \{0, 1\}, \quad \forall j \in D,$$

where $c_j = \mathbf{E}(h(\hat{\nu}) - \psi_j)$ as estimated by our affiliation model and A is the binary block-district matrix with

$a_{ij} = 1$ if block b_i is in district d_j . Constraint (2.2) enforces that each block is assigned to exactly one district and constraint (2.3) enforces that there are exactly k districts. Constraint (2.3) is strictly necessary only when $k\epsilon_p > 1$, by virtue of the generation process. Any additional linear constraints placed on the composition of districts can be added seamlessly to the formulation (e.g., requiring a minimum number of districts meet a threshold level of competitiveness). Finally, we make the objective linear by introducing and minimizing a new variable w , such that $\sum_{j \in D} c_j x_j \leq w$ and $\sum_{j \in D} c_j x_j \geq -w$.

We solve this integer program for the columns generated from each root partition separately for three reasons. By sharding the tree by root partition, the whole pipeline becomes arbitrarily parallelizable. One could set up an arbitrary number of computational workers where each worker runs the SHP generation routine and then solves the MSP for each generated root partition. One line of future work is to use optimal transport [1] to characterize the convergence behavior of our method to understand when it is unlikely to improve upon the existing solution pool so we can employ more principled terminating conditions.

One concern that might be raised is that by separating subproblems by the root partition, we limit the possible district plans; however, unless an internal node (R, s) has an identical copy in another subtree, it is unlikely that two columns generated from different partitions will be compatible. That is, with high probability there does not exist $k - 2$ other generated districts that can be included to satisfy constraint (2) of the MSP. One could search for duplicates across the entire tree and augment the column set to include all descendants of all duplicates. However, duplicates are fairly rare and limits the inherent parallelism of our approach.

Finally, in a state where a fair solution exists, there are typically a large number of alternative fair solutions. Since there are many other desirable properties of districts plans to trade off, one can filter the range of candidate plans to those that meet a requisite level of fairness, and then consider other properties such as competitiveness, maintaining political boundaries, or not separating communities of interest. Solving one MSP per root partition is a very simple way to get a large number of fair solutions, which are also substantially different from each other. One could use a more sophisticated method [37] to find many near optimal solutions for each root partition, but these are more likely to be very similar given the structure imposed by the shared root partition.

3 Results

We provide two types of experimental results. The first set of experiments is designed to tune the algorithm’s parameters and understand its behavior. We study the performance of different configurations of the SHP generation routine and the impact of varying the PDP parameters. We further provide empirical results of convergence behavior and runtime performance. These results are included in Appendices D and F¹

In our main experiment, we generate a large number of districts for every multi-district state to understand the range of possible outcomes using only reasonably shaped districts and compare this to a recombination ensemble baseline. Given the exponential nature of our column space, this constitutes the largest district plan ensemble ever (implicitly) constructed. However, this also makes computing some ensemble-level metrics intractable. Therefore, for some results, we use a pruned sample tree to make enumeration feasible. See Appendix E for all experiment details.

Compactness Compactness has long been used as the objective function in political redistricting research, rationalized by the claim that a compact map drawn with no political input is inherently fair. Although highly noncompact human drawn districts can signal manipulation, there is no *a priori* reason to assume that a randomly drawn compact map will be more fair than a randomly drawn noncompact map. To verify this, we compute the Spearman correlation coefficient [36] for each state between the magnitude of the expected efficiency gap and three different measures of compactness: centralization, Roeck, and cut-edges. Centralization is the average distance a constituent has to walk to the center of their district, averaged over all districts (similar to the dispersion measure of compactness [45]); Roeck is the district area divided by the area of the smallest circumscribing circle averaged over all districts; and cut-edges measures the average number of edges that need to be cut in the adjacency graph to form the districts of a plan.

Figure 4 and Table 1 show that compactness and fairness are orthogonal attributes of a district plan. Any correlation between compactness and fairness is noise stemming from the political geography of a state. For instance in Nebraska, where the correlation is the strongest, the only redistricting decision that affects the expected efficiency gap is whether Omaha is cracked and diluted into two reliably Republican districts (not compact and not fair) or packed into one toss-up district

	mean ρ	k -weighted mean ρ
centralization	0.074756	0.038859
roeck	-0.000355	-0.008970
cut_edges	0.105347	0.066784

Table 1: Average (and k -weighted average) Spearman correlation coefficient between the magnitude of expected efficiency gap and three different compactness metrics over all multi-district states.

(compact and fair). In many swing states where redistricting decisions can have the most impact, such as Pennsylvania, Wisconsin, and Georgia, the correlation is actually negative. This is because the most compact plans pack the major cities into landslide Democratic districts, while the surrounding suburbs and rural areas have a more efficient placement of Republican voters. Lastly, in the largest states, California and Texas, where no single geographic feature can dominate the correlation, the relationship is almost perfectly independent.

We do not claim that compactness is not important or desirable; compact plans are more likely to represent more coherent communities and make administering elections marginally easier. However, fairness is a far more important consideration and therefore we want to encourage practitioners to treat compactness as a constraint rather than an objective.

Fairness Unfortunately, fairness, like compactness, does not have a universally agreed-upon characterization; there are different and often competing aspects of fairness, which are difficult to reconcile with the distribution of partisan voters. The political geography of a state, typically defined by a few urban concentrations of more liberal voters punctuating larger and sparser regions of more conservative voters, does not guarantee a fair solution for any reasonable definition of fair. This greatly complicates creating a unified definition of fairness when the number of districts and distribution of voters greatly constraints the space of expected outcomes [11]. We use the efficiency gap [39] as our primary metric because it has gained traction as a standard metric in redistricting lawsuits, has an intuitive and compelling interpretation, and is simple to embed in an integer program as we have shown in Section 2.3.

Figure 5 shows the range of expected outcomes found by our generation algorithm and includes the point which would minimize the efficiency gap as well as the actual average seat-share over the last redistricting round ordered by partisan affiliation (see Appendix E for experiment details). By construction, our approach generates only relatively compact districts which we call *natural* districts, so this range can be thought of as

¹The full paper with appendices is available at <https://arxiv.org/abs/2103.11469>; all references to appendices within the text are in reference to this full version.

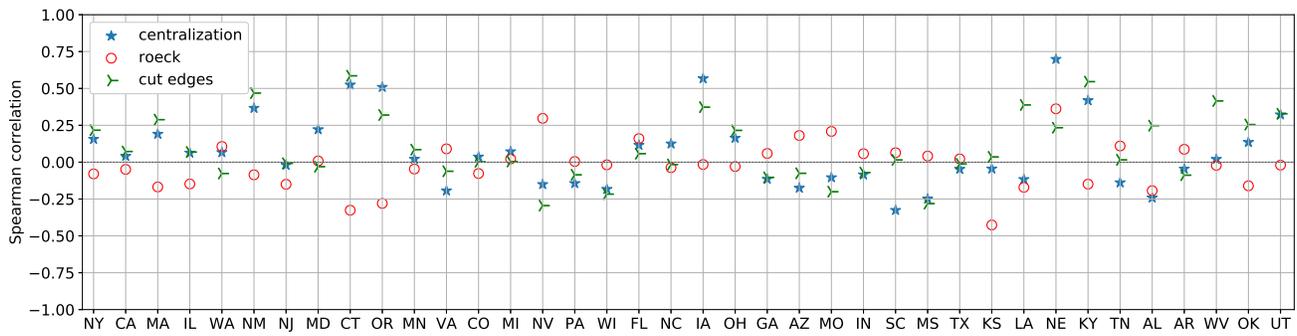


Figure 4: Spearman correlation coefficient between the magnitude of the expected efficiency gap and three different measures of compactness for the 38 states with three or more congressional districts ordered by partisan affiliation.

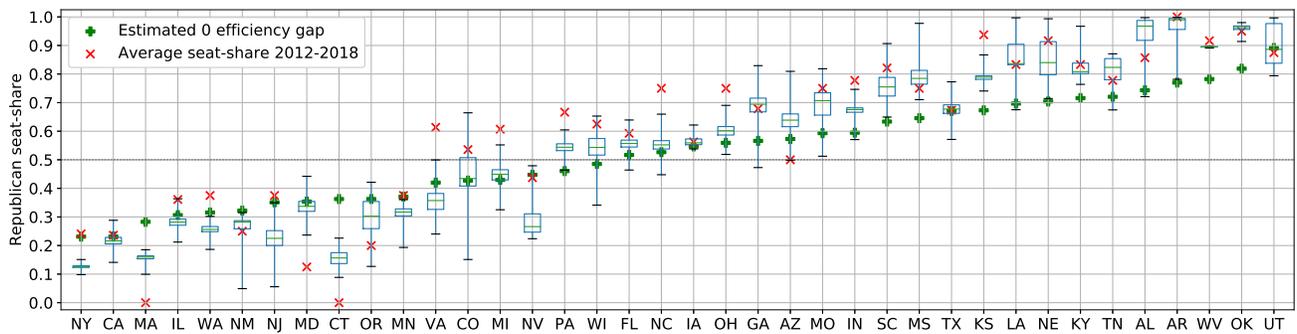


Figure 5: Distribution of expected Republican seat-share of generated plan ensemble for each state with three or more congressional districts ordered by partisan affiliation. The box center indicates the median, with the top and bottom of the box indicating the 75th and 25th percentiles respectively of the down-sampled ensemble. The whiskers denote the minimum and maximum value of any plan found using the tree based dynamic programming optimization algorithm.

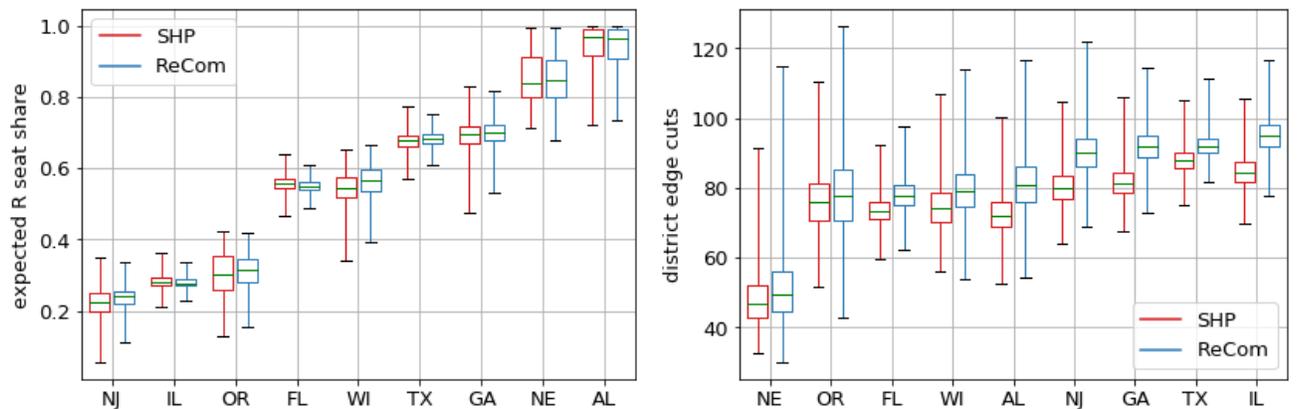


Figure 6: Comparison of expected seat-share (left) and average district edge cuts (right) on ensembles generated with Stochastic Hierarchical Partitioning (SHP) and recombination Markov chains.

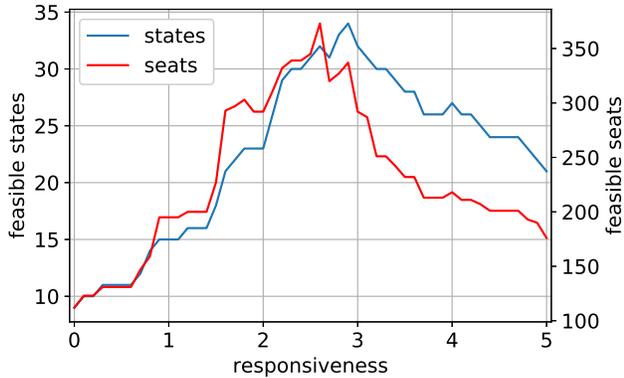


Figure 7: The number of states and total seats that admit a district plan in our ensemble for an expected level of responsiveness.

the geographic tolerance for natural or unintentional gerrymandering [6]: reasonably shaped plans that pass the eyeball test but are actually unrepresentative. In every state, we find a plan that is more compact than the current one; in just 7 states the enacted plan is more compact than our median plan, and in 8 states (most of which are well known for gerrymandering) the enacted plan is less compact than any plan we generate (see Appendix E).

The headline result is that using only natural district boundaries, we can change the expected composition of Democrats in the House of Representatives from 43% to 62%. A 20% swing in the partisan composition of the House is exceptionally consequential, since a body with full control of all congressional boundaries could theoretically create a permanent majority or permanent minority for either party, barring any extreme shifts in partisan affiliation along geographic lines. We run the same experiment with competitiveness (see Appendix E) and in aggregate achieve plans with natural districts that have between 32 and 106 expected total seat swaps per election.

However, the expected partisan seat range is highly variable. Some states like Colorado admit reasonable district plans that can swing by up to 50% of total seats, whereas others like West Virginia or Iowa offer very little room to alter the outcome using only natural districts. Consequently, any definition of fairness that relies on an ideal level of responsiveness (e.g., the efficiency gap) will not be feasible in all states without relaxing the compactness or contiguity constraints. Figure 7 shows the number of feasible states (and the corresponding number of seats) that have a solution that achieves a target level of responsiveness. In particular, only 15 states admit a proportional plan where the expected

seat-share equals the expected vote-share. Additionally, we found plans with 0 expected efficiency gap for only half of the multi-district states.

Recombination Baseline To test our claims about more effective sampling of extreme solutions, we compare ensembles generated with our stochastic hierarchical partitioning (SHP) algorithm and recombination Markov chains [9], the predominant tool in quantitative redistricting analyses. Recombination chains generate ensembles by iteratively merging two random adjacent districts, sampling a random spanning tree, and choosing a tree cut to balance the populations of the two connected components to create two new districts.

In Figure 6, we compare the expected seat-share and the edge cuts between ensembles generated with the two methods for a representative sample of states. To make the comparison as meaningful as possible, we generate the same number of distinct districts with each method for each state. In all states but Nebraska, SHP finds a larger range of expected seat-shares (while also being significantly more compact on average). This is expected since small states have very short sample trees and therefore do not benefit as much from more efficient columns as larger states do. Furthermore, by using random spanning trees instead of an inner partition integer program that optimizes for region compactness, recombination chains can sample less compact partitions, which may achieve a more extreme outcome than a natural district would.

4 Conclusion

Part methodological exposition, part civic plea, this paper introduces a scalable new method for creating fair political districts. Using the ensembling capability of our algorithm, we performed the largest ever district ensemble study to show the spurious relationship between fairness and compactness. We also illustrated the range of possible outcomes with reasonably shaped districts to better frame the constraints of fairness. We want to emphasize the overall framework more than any specific design decision, as we see the main contribution of this work as offering an alternate paradigm for solving supervised regionalization problems, most notably congressional redistricting, and potentially a more general heuristic for solving other hard computational problems.

Most importantly, we want our work to actually move the needle on redistricting reform. Our core capability is quickly, cheaply, and transparently generating a huge number of legal and fair district plans. We extend an open invitation to collaborate with states in the upcoming redistricting cycle and hope that the maps of the next decade are more representative than those of the last.

References

- [1] Tara Abrishami et al. “Geometry of graph partitions via optimal transport”. In: *SIAM Journal on Scientific Computing* 42.5 (2020), A3340–A3366.
- [2] Cynthia Barnhart et al. “Branch-and-price: Column generation for solving huge integer programs”. In: *Operations research* 46.3 (1998), pp. 316–329.
- [3] Mitchell N Berman. “Managing Gerrymandering”. In: *Tex L. Rev.* 83 (2004), p. 781.
- [4] Bruce E Cain et al. “A reasonable bias approach to gerrymandering: Using automated plan generation to evaluate redistricting proposals”. In: *Wm. & Mary L. Rev.* 59 (2017), p. 1521.
- [5] Tanima Chatterjee and Bhaskar DasGupta. “On partisan bias in redistricting: computational complexity meets the science of gerrymandering”. In: *arXiv preprint arXiv:1910.01565* (2019).
- [6] Jowei Chen and Jonathan Rodden. “Unintentional gerrymandering: Political geography and electoral bias in legislatures”. In: *Quarterly Journal of Political Science* 8.3 (2013), pp. 239–269.
- [7] Maria Chikina, Alan Frieze, and Wesley Pegden. “Assessing significance in a Markov chain without mixing”. In: *Proceedings of the National Academy of Sciences* 114.11 (2017), pp. 2860–2864.
- [8] Vincent Cohen-Addad, Philip N Klein, and Neal E Young. “Balanced centroidal power diagrams for redistricting”. In: *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. 2018, pp. 389–396.
- [9] Daryl DeFord, Moon Duchin, and Justin Solomon. “Recombination: A family of Markov chains for redistricting”. In: *arXiv preprint arXiv:1911.05725* (2019).
- [10] Moon Duchin. *Outlier analysis for Pennsylvania congressional redistricting*. 2018.
- [11] Moon Duchin et al. “Locating the representational baseline: Republicans in Massachusetts”. In: *Election Law Journal: Rules, Politics, and Policy* 18.4 (2019), pp. 388–401.
- [12] Juan Carlos Duque, Raúl Ramos, and Jordi Suriñach. “Supervised regionalization methods: A survey”. In: *International Regional Science Review* 30.3 (2007), pp. 195–220.
- [13] Robert S Erikson. “Malapportionment, gerrymandering, and party fortunes in congressional elections”. In: *The American Political Science Review* (1972), pp. 1234–1245.
- [14] Lester Randolph Ford Jr and Delbert Ray Fulkerson. “Solving the transportation problem”. In: *Management Science* 3.1 (1956), pp. 24–32.
- [15] Robert S Garfinkel and George L Nemhauser. “Optimal political districting by implicit enumeration techniques”. In: *Management Science* 16.8 (1970), B–495.
- [16] Andrew Gelman and Gary King. “Estimating the electoral consequences of legislative redistricting”. In: *Journal of the American Statistical Association* 85.410 (1990), pp. 274–282.
- [17] Ahmed Ghoniem and Hanif D Sherali. “Complementary column generation and bounding approaches for set partitioning formulations”. In: *Optimization Letters* 3.1 (2009), p. 123.
- [18] Gregory Herschlag, Robert Ravier, and Jonathan C Mattingly. “Evaluating partisan gerrymandering in Wisconsin”. In: *arXiv preprint arXiv:1709.01596* (2017).
- [19] Sidney Wayne Hess et al. “Nonpartisan political redistricting by computer”. In: *Operations Research* 13.6 (1965), pp. 998–1006.
- [20] Douglas M King, Sheldon H Jacobson, and Edward C Sewell. “Efficient geo-graph contiguity and hole algorithms for geographic zoning and dynamic plane graph partitioning”. In: *Mathematical Programming* 149.1-2 (2015), pp. 425–457.
- [21] Richard Kueng, Dustin G Mixon, and Soledad Villar. “Fair redistricting is hard”. In: *Theoretical Computer Science* 791 (2019), pp. 28–35.
- [22] Harry A Levin and Sorelle A Friedler. “Automated congressional redistricting”. In: *Journal of Experimental Algorithmics (JEA)* 24.1 (2019), pp. 1–24.
- [23] Justin Levitt. “A citizen’s guide to redistricting”. In: *Available at SSRN 1647221* (2008).
- [24] Zhenping Li, Rui-Sheng Wang, and Yong Wang. “A quadratic programming model for political districting problem”. In: *Proceedings of the first international symposium on optimization and system biology (OSB)*. Beijing, China. 2007.
- [25] Yan Y Liu, Wendy K Tam Cho, and Shaowen Wang. “PEAR: a massively parallel evolutionary computation approach for political redistricting optimization and analysis”. In: *Swarm and Evolutionary Computation* 30 (2016), pp. 78–92.
- [26] Nolan McCarty, Keith T Poole, and Howard Rosenthal. “Does gerrymandering cause polarization?” In: *American Journal of Political Science* 53.3 (2009), pp. 666–680.

- [27] Anthony J McGann et al. *Gerrymandering in America: The House of Representatives, the Supreme Court, and the future of popular sovereignty*. Cambridge University Press, 2016.
- [28] Anuj Mehrotra, Ellis L Johnson, and George L Nemhauser. “An optimization based heuristic for political districting”. In: *Management Science* 44.8 (1998), pp. 1100–1114.
- [29] Bjørn Nygreen. “European assembly constituencies for Wales-comparing of methods for solving a political districting problem”. In: *Mathematical Programming* 42.1-3 (1988), pp. 159–169.
- [30] Daniel D Polsby and Robert D Popper. “The third criterion: Compactness as a procedural safeguard against partisan gerrymandering”. In: *Yale Law & Policy Review* 9.2 (1991), pp. 301–353.
- [31] Clemens Puppe and Attila Tasnádi. “A computational approach to unbiased districting”. In: *Mathematical and Computer Modelling* 48.9-10 (2008), pp. 1455–1460.
- [32] *Reynolds v. Sims*. 1964.
- [33] Federica Ricca, Andrea Scozzari, and Bruno Simeone. “Political districting: from classical models to recent approaches”. In: *Annals of Operations Research* 204.1 (2013), pp. 271–299.
- [34] Federica Ricca, Andrea Scozzari, and Bruno Simeone. “Weighted Voronoi region algorithms for political districting”. In: *Mathematical and Computer Modelling* 48.9-10 (2008), pp. 1468–1477.
- [35] Federica Ricca and Bruno Simeone. “Local search algorithms for political districting”. In: *European Journal of Operational Research* 189.3 (2008), pp. 1409–1426.
- [36] Patrick Schober, Christa Boer, and Lothar A Schwarte. “Correlation coefficients: appropriate use and interpretation”. In: *Anesthesia & Analgesia* 126.5 (2018), pp. 1763–1768.
- [37] Thiago Serra and JN Hooker. “Compact representation of near-optimal integer programming solutions”. In: *Mathematical Programming* (2019), pp. 1–34.
- [38] Nicholas O Stephanopoulos. “The causes and consequences of gerrymandering”. In: *Wm. & Mary L. Rev.* 59 (2017), p. 2115.
- [39] Nicholas O Stephanopoulos and Eric M McGhee. “Partisan gerrymandering and the efficiency gap”. In: *U. Chi. L. Rev.* 82 (2015), p. 831.
- [40] Rahul Swamy, Douglas King, and Sheldon Jacobson. “Multi-Objective Optimization for Political Districting: A Scalable Multilevel Approach”. In: (2019).
- [41] Hamidreza Validi, Austin Buchanan, and Eugene Lykhovyd. “Imposing contiguity constraints in political districting models”. In: (2020).
- [42] William Vickrey. “On the prevention of gerrymandering”. In: *Political Science Quarterly* 76.1 (1961), pp. 105–110.
- [43] Gregory S Warrington. “Quantifying gerrymandering using the vote distribution”. In: *Election Law Journal* 17.1 (2018), pp. 39–57.
- [44] *Wesberry v. Sanders*. 1964.
- [45] H Peyton Young. “Measuring the compactness of legislative districts”. In: *Legislative Studies Quarterly* 13.1 (1988), pp. 105–115.