Sampling Balanced Forests of Grids in Polynomial Time

Sarah Cannon
Claremont McKenna College
Claremont, USA
scannon@cmc.edu

Wesley Pegden
Carnegie Mellon University
Pittsburgh, USA
wes@math.cmu.edu

Jamie Tucker-Foltz
Harvard University
Boston, USA
jtuckerfoltz@g.harvard.edu

ABSTRACT

We prove that a polynomial fraction of the set of k-component forests in the $m \times n$ grid graph have equal numbers of vertices in each component, for any constant k. This resolves a conjecture of Charikar, Liu, Liu, and Vuong, and establishes the first provably polynomial-time algorithm for (exactly or approximately) sampling balanced grid graph partitions according to the spanning tree distribution, which weights each k-partition according to the product, across its k pieces, of the number of spanning trees of each piece. Our result follows from a careful analysis of the probability a uniformly random spanning tree of the grid can be cut into balanced pieces.

Beyond grids, we show that for a broad family of lattice-like graphs, we achieve balance up to any multiplicative $(1 \pm \varepsilon)$ constant with constant probability. More generally, we show that, with constant probability, components derived from uniform spanning trees can approximate any given partition of a planar region specified by Jordan curves. This implies polynomial-time algorithms for sampling approximately balanced tree-weighted partitions for lattice-like graphs.

Our results have applications to understanding political districtings, where there is an underlying graph of indivisible geographic units that must be partitioned into k population-balanced connected subgraphs. In this setting, tree-weighted partitions have interesting geometric properties, and this has stimulated significant effort to develop methods to sample them.

CCS CONCEPTS

• **Theory of computation → Random walks and Markov chains**.

KEYWORDS

Spanning trees, Grid graphs, Redistricting, Markov chains, Random walks, Wilson's algorithm

ACM Reference Format:
Sarah Cannon, Wesley Pegden, and Jamie Tucker-Foltz. 2024. Sampling Balanced Forests of Grids in Polynomial Time. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3618260.3649699

STOC '24, June 24–28, 2024, Vancouver, BC, Canada

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649699

1 INTRODUCTION

We consider the following question: given a graph G and an integer constant k, how can one randomly sample partitions of G into k connected pieces, each of equal size? We address this question in the context of the *spanning tree distribution* on partitions, under which the weight of a partition is proportional to the product of the numbers of spanning trees in each partition class. This distribution has been the subject of intense research in the context of mathematical approaches to the analysis of political districtings [8, 9, 12, 16, 25, 27, 29]. While efficient algorithms exist to sample from this distribution when there are no size constraints on the partition classes, there is no general recipe for converting such a sampler to an efficient sampler for the *balanced* spanning tree distribution, where we condition the spanning tree distribution on the event that the partition classes are equal in size. For the prototypical case of grid graphs, the following conjecture of Charikar, Liu, Liu, and Vuong asserted that rejection sampling would suffice:

Conjecture 1.1 (Charikar, Liu, Liu, and Vuong [9]). *For the $m \times n$ grid graph, the proportion of balanced k-partitions under the spanning tree distribution is at least $1/\mathrm{poly}(m, n)$, when $k = O(1)$.*

We confirm this conjecture as follows:

Theorem 1.2. *Let G be an $m \times n$ grid graph where $m \geq n$ and $k|m$. The probability that a k-partition from the spanning tree distribution is balanced is at least*

$$\frac{1}{\beta^{k^2} n^{5k-5} m^{3k-3}} \tag{1}$$

for a fixed constant β.

We note that the assumption that k divides the longer dimension is mostly for ease of exposition. With some more effort (and worse constant factors) one could require just $k|nm$, with essentially the same proof techniques. Theorem 1.2 will follow from Theorem 3.5, which will assert that, for a uniformly random spanning tree of the $m \times n$ grid graph ($m \geq n, k|m$), there is a $1/\mathrm{poly}(mn)$ chance that there are $k-1$ edges whose removal divides the tree into equal-size components. Section 3 is devoted to proving Theorem 3.5, along with stronger bounds for the special case of $k = 2$.

The relative frequency of balanced partitions under the spanning tree distribution is particularly salient given the significant progress made in sampling algorithms for this distribution. For example, in 2020, leveraging recent breakthroughs in the polynomial-method approach to Markov chain mixing, Anari, Liu, Gharan, Vinzant, and Vuong gave an $O(N \log^2 N)$ approximate sampler based on the 'down-up' walk on the complement of k-component forests of an N-vertex graph [1]. In Section 2.5, we discuss the use of our results in the context of an additional rejection step for approximate samplers based on Markov chains, and also show how to exactly sample from the spanning tree distribution on balanced k-partitions in expected

time $O(N^{3k-2} \log N)$ for a grid graph with N vertices. These are the first provably polynomial-time algorithms for (perfectly or approximately) sampling from the spanning tree distribution on balanced partitions.

In Section 4, we turn to analyze partitions in grid-like graphs under a looser balance constraint. If we are interested in dividing a random spanning tree into components that are only approximately balanced (up to a $(1 \pm \varepsilon)$ multiplicative error), we show on lattice-like graphs (including grids) that this is possible with *constant* probability; Corollary 4.3 gives the precise statement for grids. In fact, we prove a more general result, which is that a uniform spanning tree on a sufficiently refined lattice-like graph will, with probability bounded below by a constant, be splittable into components that approximately match any partition of a region of the plane given by a collection of Jordan curves (Figure 1). In particular, suppose Λ_n is a sequence of infinite planar graphs of decreasing scale embedded in $\mathbb{R}^2$ which are lattice-like (see Definition 4.1). For example, our definition of "lattice-like" is broad enough to apply almost surely to the sequence where Λ_n is the Delauney triangulation of a Poisson point cloud in $\mathbb{R}^2$ of rate n. If D is a fixed plane graph, and Ω_{D,Λ_n} denotes a region of Λ_n whose boundary approximates the boundary of the outer face of D, we have that:

Theorem 1.3 (Informal version of Theorem 4.2). *Given any plane graph D with $k+1$ faces, let $\phi_1, \ldots, \phi_k \subseteq \mathbb{R}^2$ denote its inner faces. For any $\varepsilon > 0$, as $n \to \infty$, there is a constant lower bound, depending only on the plane graph and ε, on the probability that a random spanning tree T of Ω_{D,Λ_n} contains $k-1$ edges whose removal disconnects T into components $C_1, \ldots, C_k$, where each C_i is at Hausdorff distance $< \varepsilon$ from a corresponding face ϕ_i of D.*

Again, combining these results with known algorithms and rejection sampling gives corresponding polynomial-time sampling algorithms in these more general settings.

1.1 Random Sampling of Political Districting Plans

In the context of the analysis of districting plans, sampling algorithms enable the generation of large *ensembles* of plans, which are useful for several purposes (detecting outliers, understanding the impacts of rules, evaluating the stated intentions of map-drawers, and more). Ensemble analysis has been used in many academic studies, including [2, 3, 5–8, 11, 13, 15, 16, 18, 19, 21, 22, 25, 32], as well as in mathematicians' expert reports in court cases [4, 10, 17, 24].

Randomly sampling political districting plans is equivalent to a sampling problem for suitable partitions of a graph, with vertices representing small geographic regions such as precincts or census blocks and edges representing adjacencies. Because they represent physical geography, these graphs are typically planar or nearly planar. While they are not usually perfect grids (except at times in cities), there is general consensus that grids are the logical simplified setting to first consider. By going beyond grids to lattice-like graphs, we move to a much more expressive graph class that can describe significant additional real-world geography.

A districting plan with k districts is a partition of this graph into k pieces, which are generally required to be connected. Throughout, we will call a partition of a graph into k connected pieces a k-*partition*, and we will refer to the k partition classes of a partition as *districts*.

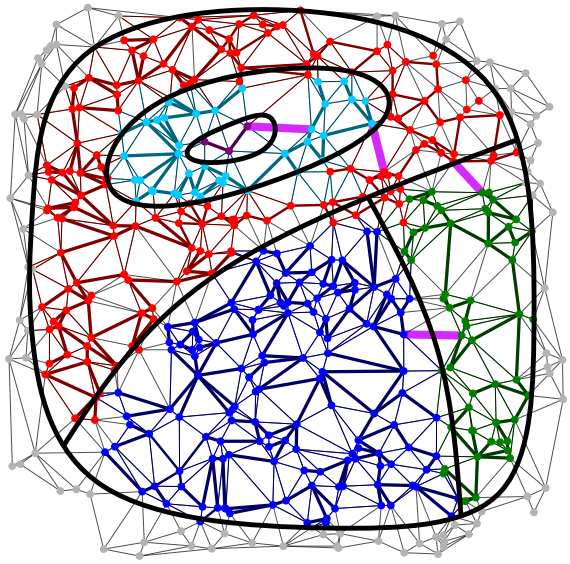

Figure 1: A partition of a region of a lattice-like graph approximating a division of the plane given by Jordan curves, and induced by the components remaining after deleting the four bright purple edges from a spanning tree of the region. Theorem 4.2 shows that given a division of the plane by curves, a random spanning tree of a sufficiently refined lattice-like graph can, with probability bounded below by a constant, be cut into components inducing a partition whose classes each has small Hausdorff distance from the corresponding face of the drawing.

In the context of redistricting, there are other constraints on partitions one must consider, including those related to population and shape. Our interest in balanced partitions stems from common requirements that districts have equal or near-equal populations. While our first main result resolves a conjecture about exactly balanced partitions, in practice most processes for sampling political districting plans do not aim for exact population balance but instead aim to keep the population to within a tolerance of 1-2%. This naturally corresponds to the setting of Theorem 1.3, where district sizes may vary by a multiplicative $1 \pm \varepsilon$ factor. Related to district shape, the spanning-tree distribution we analyze is targeted by several sampling algorithms designed for redistricting analysis [8, 16, 25], and has been shown to strongly correlate with geometric properties intended to capture legal requirements for 'compactness' of districts [12, 27, 29].

Unlike Markov chains such as the up-down walk, which operating in a context without balance constraints, we know that the approaches cited above such as *recombination Markov chains* can have exponential mixing time for some special families of graphs (including carefully chosen subgraphs of the grid) [9]. Even on rectangular grids, recombination chains with strict balance constraints

can fail to be ergodic if there are many small districts [30]. Positive mixing time results for any reasonable class of graphs are not available. However, by giving the first polynomial lower bounds in grid and grid-like graphs on the probability of finding edges that cut random spanning trees in balanced ways, our approach also addresses a crucial factor for Markov chains like those in [8, 16, 25] that aim to achieve balance by preserving it at every step, by only using such balanced cuts in transitions.

Other Markov chains employed in the redistricting context include Glauber dynamics for contiguous partitions, which exchange individual vertices between districts. Here, without any additional constraints or weighting, stationary distributions are uniform on partitions with connected districts. Mixing time can again be exponential for some classes of graphs [20]. In fact, even in the absence of balance, it is not known whether the Glauber dynamics has polynomial mixing time for partitions of grid graphs into k connected pieces, or indeed whether any polynomial time algorithm to uniformly sample partitions of grid graphs uniformly randomly into k connected pieces exists, even for $k = 2$.

1.2 Approach

Rather than working with the tree distribution on partitions, we work with the uniform distribution on spanning trees. As we prove in Lemma 2.4, if there is a polynomial lower bound on the probability a random spanning tree can be split into k equal-sized components, there is a polynomial lower bound on the probability a random tree-weighted forest with k components is balanced. The majority of our work therefore focuses on uniformly random spanning trees and the probability they can be split into components with desired properties.

Spanning trees of planar graphs are in bijective correspondence with the spanning trees of their dual graphs: If T is a spanning tree of G, its dual spanning tree T^* contains all edges in G^* whose corresponding edges are missing from T. The first key idea behind our approach is to study the structure of T^* rather than T. If T is a spanning tree of G with dual tree T^*, then the k connected components of $T \setminus e_1, \ldots, e_{k-1}$ are bounded by k cycles in $T^* \cup e_1 \cup \cdots \cup e_{k-1}$. In particular, to show that components with certain sizes or structure can be created by removing edges in T, it suffices to show that suitable boundary cycles almost already exist in T^*.

The second key idea is to study the probability of such suitable near-cycles occurring in T^* by analyzing the steps of Wilson's algorithm on the dual graph. For an arbitrary root vertex, Wilson's algorithm builds a uniformly random spanning tree by running a series of loop-erased random walks from arbitrary starting points to the component containing the root [31]. By choosing the root to be the dual graph vertex corresponding to the exterior face and carefully choosing the starting points of each random walk, we are able to show the algorithm is sufficiently likely to produce paths in T^* that have the properties we desire.

For our results on general lattice sequences, we will use a particular implementation of Wilson's algorithm described in Section 4.3 in which, having completed one loop erased random walk, we (sometimes) choose the next starting point for a new loop-erased random walk as the exit vertex of simple random walk within the induced subgraph of the already-built tree itself. This allows us to analyze the progress of the algorithm in long phases that may include many separate loop-erased random walks, but for which these separate loop-erased random walks can all be seen as being generated using a single random walk on the graph.

1.3 Note About Full Version of Paper

This is the conference version of this paper. Several things have been omitted due to space considerations. The full version of this paper is available at https://arxiv.org/abs/2310.15152 and additionally contains:

- All missing proofs, and some additional supporting lemmas.
- A short section on empirical experiments on grid graphs, where we estimate the probabilities of certain edges being contained in a uniformly random spanning tree and splitting it into parts of various sizes.
- A new theoretical result about lattice-like graphs which we did not have ready in time for the conference submission deadline. It is similar to Corollary 4.3, except that we obtain an *additive* approximation to balance with *polynomial* probability (rather than a multiplicative approximation with polynomial probability), assuming some mild additional axioms about the family of graphs.

2 PRELIMINARIES

2.1 Notation

For a positive integer n, we denote $[n] := \{1, 2, \ldots, n\}$. Unless otherwise specified, all graphs we consider are undirected with no self-loops, but multiple edges may be allowed between any pair of vertices. The $m \times n$ grid graph is the graph with vertex set $[m] \times [n]$, with an edge between (i, j) and (i', j') whenever $|i' - i| + |j' - j| = 1$. We always draw grid graphs in a Cartesian coordinate system, with m being the horizontal dimension and n being the vertical dimension. We denote by $\mathbb{Z}^2$ the infinite grid graph, where the vertex set is $\mathbb{Z} \times \mathbb{Z}$ and the edge relation is the same as in finite grids.

A forest is a graph with no cycles, and a tree is a connected forest. A *k-forest* is a forest with k connected components. A forest is *balanced* if every connected component has exactly the same number of vertices. If T is a tree and $S \subseteq E(T)$, we define $T \setminus S$ to be the forest F with vertex set $V(F) := V(T)$ and edge set $E(F) := E(T) \setminus S$. Thus, a tree T is *k-splittable* if there is some set $S \subseteq E(T)$ of size $k - 1$ such that $T \setminus S$ is a balanced k-forest.

For a graph H, we let $\mathrm{sp}(H)$ denote the number of spanning trees of H. For a k-partition P of G with districts $P_1, \ldots, P_k$, we denote by π_{sp} the spanning tree distribution, given by

$$\pi_{\mathrm{sp}}(P) = \frac{\prod_{i=1}^{k} \mathrm{sp}(P_i)}{Z},$$

where Z is the normalizing constant, also called the partition function, given by

$$Z = \sum_{k\text{-partitions } P} \prod_{i=1}^{k} \mathrm{sp}(P_i).$$

Note that the uniform distribution over k-forests of G is equivalent to the spanning tree distribution over k-partitions of G when a forest is identified with its connected components.

We write $d(x, y)$ to denote the Euclidean distance between two points $x, y \in \mathbb{R}^2$. The *Hausdorff distance* between two subsets $X, Y \subseteq \mathbb{R}^2$, written $d(X, Y)$, is defined as

$$d(X, Y) := \max\{\sup_{x\in X} \inf_{y\in Y} d(x, y), \sup_{y\in Y} \inf_{x\in X} d(x, y)\}.$$

2.2 Duality

Let G be a connected, planar graph, and fix an embedding of G in the plane with no edges crossing. The *dual graph* of G (with respect to the embedding) is the graph G^* whose vertices are faces of G, with an edge between two faces $a^* \in V(G^*)$ and $b^* \in V(G^*)$ whenever the two faces share a common boundary edge. Note that we count the outer face of G as a vertex of G^* as well.

For any edge $e \in E(G)$, let $e^* \in E(G^*)$ be the edge between the faces it bounds. For any set of edges $S \subseteq E(G)$, we analogously define $S^* := \{e^* \mid e \in S\} \subseteq E(G^*)$. The following lemma is a standard result.

Lemma 2.1. *Assume that G is connected and embedded in the plane such that no edge of G has the same face on both sides. Then $e \mapsto e^*$ is a bijection between edges of G and edges of G^*, and $T \mapsto T^* := (V(G^*), E(G^*) \setminus E(T)^*)$ is a bijection between spanning trees of G and spanning trees of G^*.*

Note that T^* does not contain the edges e^* for each $e \in T$, but rather those edges that are *not* in this set. The hypotheses of Lemma 2.1 hold for all $m \times n$ grid graphs with $m, n > 1$.

2.3 Wilson's Algorithm

Wilson's algorithm [31] is important for us not just because it samples uniformly random trees efficiently, thus serving as a key subroutine in our perfect sampling algorithm (See Section 2.5.1), but also because our proofs rely on running Wilson's algorithm in a specific way.

For an input graph G, the steps of Wilson's algorithm are as follows:

(1) Set $T \leftarrow \{r\}$ for an arbitrary "root" vertex $r \in V(G)$
(2) While T does not connect all vertices of G:
   (a) Do a loop-erased random walk[1] starting at an arbitrary vertex $v \notin T$ until it reaches a vertex of T
   (b) Add all vertices and edges along this loop-erased random walk to T
(3) Return T

Importantly, it does not matter which vertex is initially chosen as the root, and in each iteration of the while loop, it does not matter at which vertex not in T the next loop-erased random walk begins. Regardless of what arbitrary choices are made at these steps, one can prove the end result is a perfectly uniformly random spanning tree of G. We use this crucial fact in our proofs, analyzing the process of Wilson's algorithm (in the dual graph G^*) from carefully-chosen starting vertices.

Recall that the hitting time $\tau_u(v)$ of u from v is the expected time before a simple random walk reaches v from u, and the commute time between u and v is $\tau_u(v) + \tau_v(u)$. A *π-random* vertex of G is a vertex chosen according to the stationary distribution of the simple random walk on G, $\pi(v) = deg(v)/2m$. Wilson characterizes the expected running time of his algorithm (measured by the number of times we need to find a random neighbor of a vertex) in terms of the commute time as follows:

Proposition 2.2 (Wilson). *The expected number of times we generate a random neighbor for a vertex in the course of running Wilson's algorithm on a graph G with root r is precisely the expected commute time between r and a π-random vertex v.* □

For general graphs with N vertices and M edges, it is well-known that the hitting time and thus the commute time between any pair of vertices is at most $O(NM)$ [23]; this implies that Wilson's algorithm runs in time $O(N^2)$ for any planar graph on N vertices. However, this can be improved for grid graphs by considering the dual graph and a carefully-chosen root:

Proposition 2.3. *Wilson's algorithm has expected running time $O(N \log N)$ on the dual of any grid graph on N vertices, when the root is chosen to be the dual vertex corresponding to the outer face of the grid graph.*

This is easily proved using the characterization of the commute time in terms of effective resistance; we include a proof in the full version of this paper.

2.4 Splittability and the Spanning Tree Distribution

Here we explicitly connect the uniform distribution over spanning trees of a graph G with the uniform distribution over k-forests of G. This enables us to analyze the likelihood of obtaining a balanced partition when sampling from the spanning tree distribution over forests, as the up-down walk of [9] (approximately) does; see Section 2.5.2.

Lemma 2.4. *If the probability a uniformly random spanning tree of G with N vertices and M edges is k-splittable is at least α, then the probability a uniformly random k-forest of P is balanced is at least*

$$\frac{\alpha}{N^{k-1}(M - N + 1)^{k-1}}.$$ □

The (short) proof of this lemma can be found in the full version. We now use it to prove Theorem 1.2

Proof that Theorem 3.5 implies Theorem 1.2. Sampling a k-partition P of G according to the spanning tree distribution is the same as sampling a k-forest F of G uniformly at random and then considering its connected components. By Theorem 3.5, the probability that a uniformly random spanning tree T of G is k-splittable is a least $\frac{1}{\beta^{k^2} n^{3k-3} m^{k-1}}$ for some fixed constant β. By Lemma 2.4, as G has nm vertices and strictly less than $2nm$ edges, the probability a uniformly random k-forest is balanced is therefore at most $\frac{1}{\beta^{k^2} n^{5k-5} m^{3k-3}}$. □

There is hope this bound could be improved by studying random forests directly, rather than studying spanning trees and then considering cutting them to obtain forests.

[1] That is, every time the random walk revisits a node u, erase the cycle and resume the random walk from u.

2.5 Algorithms for Sampling Balanced Tree-Weighted Partitions

Our theorems imply that known approaches sampling (not necessarily balanced) k-partitions according to the spanning tree distribution in polynomial time can be combined with a rejection sampling step to obtain an expected polynomial-time algorithm for sampling balanced k-partitions. Here we present two methods by which this could be done, for the case of sampling exactly balanced k-partitions.

2.5.1 Perfectly Sampling Balanced k-Partitions with Wilson's Algorithm. Wilson's algorithm generates a perfectly uniform random spanning tree of a graph G. We can use it to randomly sample a balanced k-partition as follows.

(1) Uniformly sample a random spanning tree T of G using Wilson's algorithm.
(2) Check if T has $k-1$ edges whose removal disconnects T into k components of equal size. If no, reject and return to step 1.
(3) If yes, create a k-partition P of G comprised of the connected components when these $k-1$ edges are removed from T.
(4) Create a graph G/P which contracts each district of P into a single point and retains all edges between components with the appropriate multiplicity.
(5) Compute the number s of spanning trees of G/P.
(6) Return P with probability $1/s$. With the remaining probability $(s-1)/s$ reject and return to step 1.

Theorem 2.5. *For N-vertex grid graphs, this algorithm produces a balanced k-partition drawn perfectly from the spanning tree distribution in expected running time $O(N^{3k-2}\log N)$.*

See the full version for a proof of this theorem. Briefly, the expected run time bounds are because it takes expected time $O(N\log N)$ steps to sample a random spanning tree and check if it is k-splittable, $O(N^{2k-2})$ attempts in expectation to see a k-splittable tree, and $O(N^{k-1})$ attempts to be successful in the final rejection of Step 5, by Theorem 3.5.

2.5.2 Approximately Sampling Balanced k-Partitions with the Up-Down Walk. An alternate method using the up-down Markov chain described in Charikar et al. can produce an approximately uniformly random k-forest [1, 9]. We briefly motivate and describe this approach here.

On any graph G, the spanning forests with at least k components form a matroid whose bases are exactly the k-component spanning forests of G. The well-known down-up chain on bases of a matroid mixes in time $O(r(\log r+\log\log n))$ when bases have r elements and the matroid has n total elements [14]; when run for longer than its mixing time, this chain produces an approximately uniformly random basis. For k-component forests, this down-up chain randomly removes an edge of the forest (to produce $k+1$ components), and then randomly adds back in an edge connecting two different components. It's mixing time is $O((N-k)(\log(N-k)+\log\log M))$ for graphs with N vertices and M edges; for constant k, this becomes $O(N\log N)$. However, naively implementing one down-up step requires $O(M)$ time, making the overall time for this chain to produce an approximate sample $O(NM\log N)$.

This was improved in [1] by considering the up-down walk instead. This walk randomly adds an edge to the forest. If adding this edge creates a cycle, a random edge of the cycle is removed. If adding this edge did not create a cycle (e.g. the edge connected two components of the forest) then a random edge of the forest is removed. This chain has mixing time $O(M\log M)$ for graphs with M edges, as the up-down walk can also be viewed as the down-up walk on the dual matroid whose bases are the complements of k-component forests. Using a link-cut tree data structure, each up-down step can be implemented in amortized quasi-constant time, resulting in an overall runtime of $O(M\log^2 M)$ to produce one random sample. For planar graphs where $M=O(N)$, this mixing time is $O(N\log^2 N)$. It is this up-down chain, rather than the usual down-up chain, that we will use.

The following is our algorithm for approximately randomly sampling a balanced k-forest of a N-vertex graph.

(1) Run the up-down Markov chain on k-forests for some fixed amount of time longer than its mixing time.
(2) If the current state of the chain is a balanced k-forest, return the partition P consisting of the connected components of the forest. Else, return to step 1 and repeat.

Theorem 2.6. *For N-vertex grid graphs, this algorithm produces a balanced k-partition drawn approximately from the spanning tree distribution in expected running time $O(N^{4k-3}\log^2 N)$.* □

See the full version for a proof of this theorem. Briefly, the expected running time is because it takes $O(N\log^2 N)$ steps to approximately sample a random k-forest and $O(N^{4k-4})$ attempts in expectation to see a balanced one, by Theorem 1.2. Note that, with the relatively crude estimates we employ to deduce Theorem 1.2 from Theorem 3.5 (in Section 2.4), the runtime we prove for this approximate sampling approach is actually worse than for the exact sampler above. There is little reason to believe this to be the truth, however.

3 EXACT BALANCE ON GRID GRAPHS

3.1 Exactly Balanced Bipartitions

In this section we prove the following:

Theorem 3.1. *Let G be a grid graph with N vertices, where N is even. The probability that a uniformly random spanning tree T of G is 2-splittable is at least $1/N^2$.*

In fact, we prove a stronger result, namely that specific edges near the center of the grid have a decent probability of being the edge that splits the tree. Formally, If G is an $m\times n$ grid graph, we define a *horizontal central edge* of G to be an edge of the form $\{(i,j),(i,j+1)\}$ that is as close to the center of G as possible. Note that there may be 1, 2, or 4 horizontal central edges depending on the parities of m and n.

Lemma 3.2. *Let G be an $m\times n$ grid graph where $m\geq n$, and mn is even, and let $e\in E(G)$ be any horizontal central edge of G. Then the probability that a uniformly random spanning tree T of G contains e, and $T\setminus\{e\}$ is a balanced 2-forest, is at least*

$$\begin{cases}\frac{1}{mn^3} & \text{if } m \text{ is even}\\ \frac{1}{4mn^3} & \text{if } m \text{ is odd}\end{cases}.$$

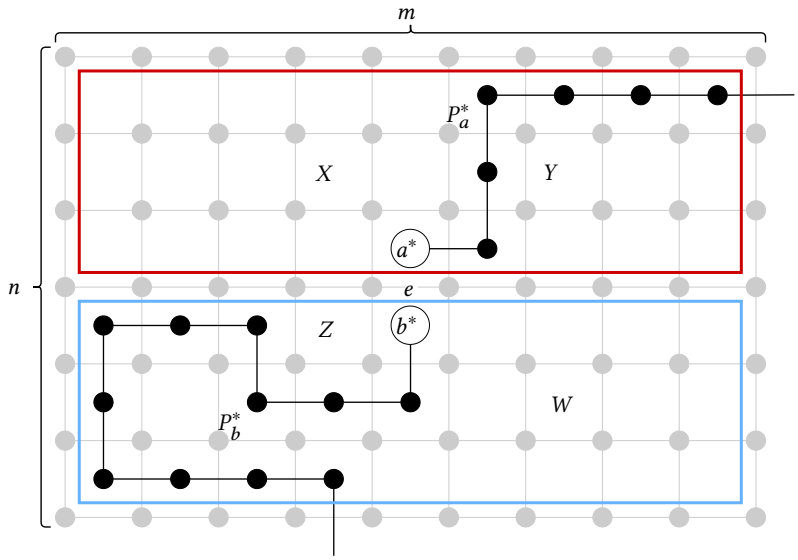

Figure 2: A possible run of the dual graph spanning tree sampling algorithm in the proof of Lemma 3.2 when m is odd. In this example, $m = 10$ and $n = 7$. The primal graph G is depicted in gray, and the first two random walks in the dual graph G^* are depicted in black.

To prove Lemma 3.2, we will require two further lemmas.

Lemma 3.3. *Let G be the $m \times n$ grid graph induced by the subset $[m] \times [n]$ of the grid $\mathbb{Z}^2$, and $(i_0, j_0) \in V(G)$. The probability that a random walk from (i_0, j_0) in $\mathbb{Z}^2$ exits G for the first time to a vertex (i', j') with $j' > 0$ is at least $\frac{j_0}{n+1}$.*

Proof. Let $(X_0, Y_0) = (i_0, j_0), (X_1, Y_1), (X_2, Y_2), \ldots$ be an infinite random walk in $\mathbb{Z}^2$. Let $\widetilde{Y}_0 = j_0, \widetilde{Y}_1, \widetilde{Y}_2, \ldots$ be the subsequence of vertical coordinates $Y_0, Y_1, Y_2, \ldots$ where we delete terms that repeat the previous value (because the horizontal coordinate changed instead). A sufficient condition for the original two-dimensional walk to first exit G to a vertex (i', j') with $j' > 0$ is for $\widetilde{Y}_0, \widetilde{Y}_1, \widetilde{Y}_2, \ldots$ to reach $n + 1$ before it reaches zero, as reaching $n + 1$ necessitates exiting G. Since $\widetilde{Y}_0, \widetilde{Y}_1, \widetilde{Y}_2, \ldots$ is just an unbiased one-dimensional random walk from j_0, it is a well-known fact[2] that this probability is precisely $\frac{j_0}{n+1}$. □

Lemma 3.4. *Let X be a discrete probability distribution supported on a set of size k. Then*

$$\Pr_{x_1, x_2 \sim X \times X}(x_1 = x_2) \geq \frac{1}{k}.$$

Proof. Suppose the probabilities of each element in the support of X are $p_1, p_2, \ldots, p_k$, where these values sum to one. Then the probability that two independent samples are the same is given by $\sum_{i=1}^k p_i^2$. Let $\mathbf{p} = (p_1, p_2, \ldots, p_k)$ and let $\mathbf{v} = \left(\frac{1}{k}, \frac{1}{k}, \ldots, \frac{1}{k}\right)$ be length k vectors. We see $||\mathbf{p}||^2 = \sum_{i=1}^k p_i^2$, $||\mathbf{v}||^2 = 1/k$, and $|\langle \mathbf{p}, \mathbf{v} \rangle|^2 = 1/k^2$, so the lemma follows immediately from the Cauchy-Schwarz inequality. □

[2]This is an instance of the classic "Gambler's Ruin" problem. This proof can be found in [26, Section 7.2.1], where $\ell_1 := j_0$ and $\ell_2 := (n + 1) - j_0$.

Proof of Lemma 3.2. Assume $n > 1$ (otherwise there is nothing to show; the probability is one). Then note that Lemma 2.1 applies to G. We first consider the case where n is odd (so m must be even). In this case there is a unique horizontal central edge e, connecting the vertices $(m/2, (n + 1)/2)$ and $(m/2 + 1, (n + 1)/2)$. Let G^* be the dual graph of G in the plane, and denote the outer face by $r^* \in V(G^*)$. Let $a^* \in V(G^*)$ be the face above e and let $b^* \in V(G^*)$ be the face below e, as in Figure 2.

Consider the following algorithm for generating a uniformly random spanning tree T of G. Run Wilson's algorithm on G^* with r^* as the root, starting the first loop-erased random walk from a^* and the second random walk from b^* (if it is not already added to the tree in the first random walk). The remaining random walks in Wilson's algorithm can be executed from arbitrary starting points. This gives us spanning tree T^* of G^*. We then output the primal tree T whose dual is T^*. Since Wilson's algorithm gives a uniformly random sample from the set of spanning trees of G^*, and those dual trees are in bijection with the primal spanning trees of G (Lemma 2.1), this algorithm gives us a uniformly random sample from the set of spanning trees of G.

We apply Lemma 3.3 to the $(m - 1) \times \frac{n-1}{2}$ dual sub-grid outlined in the top (red) rectangle in Figure 2, with initial vertex a^*. Note that $j_0 = 1$ because the coordinate system is shifted so that a^* is in the bottom row. Lemma 3.3 says that a random walk from a^* will first exit the sub-grid above, to the left, or to the right (just not below) with probability at least

$$\frac{1}{\frac{n-1}{2} + 1} > \frac{1}{(n - 1) + 1} = \frac{1}{n}.$$

This clearly applies to our loop-erased random walk as well: The probability that the first walk in Wilson's algorithm, which starts from a^*, makes it to the outer face r^* without ever entering the

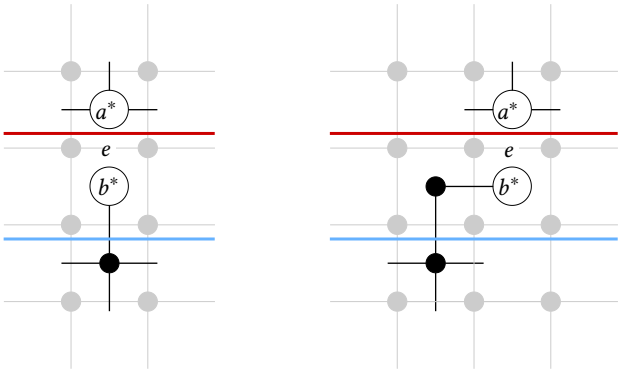

Figure 3: The cases in the proof of Lemma 3.2 when m is even, in which we must assume that the initial steps of the random walk from b^* takes a specific path into the blue rectangle, from which it never leaves until hitting the outer face.

bottom half of the grid is at least $\frac{1}{n}$. Assuming this happens, we may then apply the same argument to the bottom (blue) rectangle, for the next random walk starting from b^*. By independence, with probability at least $\frac{1}{n^2}$, both paths will have made it to r^* without crossing the horizontal midline.

Assume that this happens, as it does in Figure 2. Let P_a^* be the path from a^* to the boundary, and let P_b^* be the path from b^* to the boundary. Since both P_a^* and P_b^* will be included in T^*, we know that T cannot cross these paths. This means e must be included in T. Moreover, P_a^* and P_b^* completely determine the number of vertices on each side of e in T, as follows. Suppose there are X vertices in the top-half of the grid to the left of P_a^*, Y vertices in the top-half of the grid to the right of P_a^*, Z vertices in the bottom-half of the grid to the left of P_b^*, and W vertices in the bottom-half of the grid to the right of P_b^*. Then the subtree of T to the left of e will have $X + Z + \frac{m}{2}$ vertices, and the subtree to the right of e will have $Y + W + \frac{m}{2}$ vertices (the $\frac{m}{2}$ terms come from the vertices on the horizontal midline). Observe that the distribution, over the random path P_a^*, of the possible values of $X - Y$ is independent of and identical to the distribution, over the random path P_b^*, of the possible values of $W - Z$. Both distributions can take any integral value from $-\frac{m-1}{2}n$ to $\frac{m-1}{2}n$. Thus, applying Lemma 3.4, we know that, with probability at least

$$\frac{1}{(m\frac{n-1}{2}) - (-m\frac{n-1}{2}) + 1} = \frac{1}{mn - m + 1} > \frac{1}{mn},$$

we have $X - Y = W - Z$, which implies

$$X + Z + \frac{m}{2} = Y + W + \frac{m}{2},$$

i.e., the subtrees are balanced.

Thus, we have shown that the probability e is included in a uniformly random spanning tree T of G and splits it into a balanced 2-forest is at least

$$\frac{1}{n^2} \cdot \frac{1}{mn} = \frac{1}{mn^3}.$$

The remaining cases, where n is even, are almost the same. There are just a few minor additional assumptions we must impose about what happens to the random walks at the very beginning, as illustrated in Figure 3.

If m is even as well, there are two horizontal central edges bordering the unique central face of the grid. Without loss of generality, take e to be the top one, then define a^* and b^* as before. We suppose that the random walk from b^* first steps directly downward, as in Figure 3 (left). This happens with probability $\frac{1}{4}$. From there, by the same arguments as before, noting that the two subgrids are now each $m \times (n/2 - 1)$, the probabilities that the paths leave their respective red and blue rectangles at the boundary of the grid are both at least

$$\frac{1}{(\frac{n}{2} - 1) + 1} = \frac{2}{n}.$$

The probabilities that the number of vertices on each side are the same is at least

$$\frac{1}{(m\frac{n-2}{2}) - (-m\frac{n-2}{2}) + 1} = \frac{1}{mn - 2m + 1} > \frac{1}{mn}.$$

Thus, the probability that e splits a uniformly random spanning tree T into a balanced 2-forest is at least

$$\frac{1}{4} \cdot \left(\frac{2}{n}\right)^2 \cdot \frac{1}{mn} = \frac{1}{mn^3}.$$

Finally, consider the case where m is odd and n is even. Now there are four horizontal central edges, of which we pick the top-right one without loss of generality. With probability $\frac{1}{16}$, the random walk from b^* first steps to the left and then down into the blue rectangle, as in Figure 3 (right). Now we can again apply the same arguments as above to the subgrids of dimensions $m \times (n/2 - 1)$ showing the probability the remaining paths leave their subgrids at the boundary of G are both at least $\frac{2}{m}$. While the random walk in the top grid no longer begins exactly in the center of the top grid (it can't, because this grid is now of even width), the top and bottom grids are rotationally symmetric, with the top walk beginning just one unit left of center and the bottom walk beginning one unit right of center. As before, the distributions of difference of the number of vertices on each side of the path are identical and are supported on sets of size at most mn, so by Lemma 3.4, the probability these differences are identical is at least $\frac{1}{mn}$. Thus, the probability that e splits a uniformly random spanning tree T into a balanced 2-forest is at least

$$\frac{1}{16} \cdot \left(\frac{2}{n}\right)^2 \cdot \frac{1}{mn} = \frac{1}{4mn^3}.$$ □

We now use this lemma to prove Theorem 3.1.

Proof of Thoerem 3.1. Recall that $N = nm$ is the total number of vertices. Assuming $m \geq n$, we know that $mn^3 \leq m^2n^2 = N^2$. Thus, in the case where m is even, we simply choose one of the horizontal central edges, which, by Lemma 3.2, splits a random tree into a balanced 2-forest with probability at least $\frac{1}{mn^3} \geq \frac{1}{N^2}$. In the case where m is odd (and so n must be even) there are 4 horizontal central edges, each of which will split a random tree into a balanced 2-forest with probability at least $\frac{1}{4mn^3}$. Since these 4 events are mutually exclusive, one of these four will give a balanced split with probability at least $\frac{1}{mn^3} \geq \frac{1}{N^2}$. □

3.2 Exactly Balanced k-Partitions

With some additional effort, we can generalize the proof of Theorem 3.1 to handle values of $k > 2$:

Theorem 3.5. *For $m \geq n$, let G be an $m \times n$ grid graph, and let k be a positive integer dividing m. There exists a set $S \subseteq E(G)$ of size $k-1$ such that the probability a uniformly random spanning tree T of G contains each edge in S, and $T \setminus S$ is a balanced k-forest, at least*

$$\frac{1}{\beta^{k^2} n^{3k-3} m^{k-1}} \tag{2}$$

for a fixed constant β.

The proof proceeds along similar lines as the proof of Lemma 3.2, and is contained in the full version of this paper; here we just give a sketch. We require the following stronger lemma about random walks on grids. This is similar to Lemma 3.3, except that now we are also not allowed to hit the left or right sides, which makes the proof significantly more involved.

Lemma 3.6. *Suppose there is a constant $\varepsilon > 0$ such that $m > \varepsilon n$. Let G be the $(m+1) \times (n+1)$ grid graph induced by the subset $\{0, \dots, m\} \times \{0, \dots, n\}$ of the grid $\mathbb{Z}^2$, where $m+1$ is odd, and let $(i_0, j_0) = (\frac{m}{2}, 0)$. The probability that a random walk from (i_0, j_0) in $\mathbb{Z}^2$ exits G for the first time to a vertex (i', j') with $j' = n+1$ is at least $\frac{1}{Ane^{A/\varepsilon}}$, for a fixed constant A.* □

The proof approach is to first handle the case where $m = n$. For this we decompose the 2-dimensional random walk into two 1-dimensional random walks and show that, with decent probability, the vertical walk makes it to the far side of the square quickly *and* the horizontal walk does not stray to either side too fast. The technical tools we require are the reflection principle and Bertrand's ballot theorem.

To handle the more difficult case where $m < n$ (but n is only greater than m by some constant factor), we lower-bound the probability that the random walk leaves a sequence of squares in specific directions, as depicted in Figure 4. This is an argument we reuse later in our results for lattice-like graphs.

With Lemma 3.6 in hand, Theorem 3.5 follows from considering an implementation of Wilson's algorithm as depicted in Figure 5, where we start the first $2(k-1)$ random walks from the dual vertices $a_1^*, b_1^*, a_2^*, b_2^*, \dots, a_{k-1}^*, b_{k-1}^*$, hoping each one makes it to the respective far side.

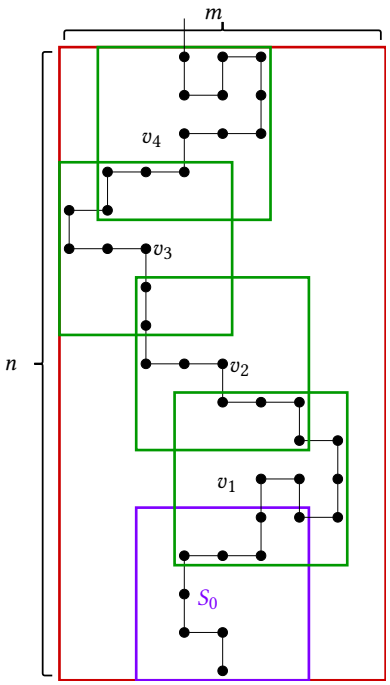

Figure 4: Illustration of the proof of Lemma 3.6, which is included in the full version. A random walk from the bottom of the red rectangle that first exits at the very top because it exits a sequence of squares at the top, and on the side pulling it closer to the center. Most of the work goes into showing that the walk leaves the very first square (in purple, at the bottom), since it must cross to the far side. This happens with inverse polynomial probability. For the subsequent green squares, the walk starts at the center, so the probability is constant. And there are a constant number of squares since n is larger than m by a constant factor.

4 APPROXIMATING PARTITIONS OF LATTICE-LIKE GRAPHS

In this section we move beyond grid graphs to a more general class of lattice-like graphs. In Section 4.1, we begin by defining what we mean by lattice-like graphs. In Section 4.2 we state our main result, along with a corollary that random trees on grid graphs are approximately balanced with constant probability. Section 4.3 gives a sketch of the proof; the details are included in the full version of the paper.

4.1 Lattice Sequences

Recall that $d(x, y)$ and $d(X, Y)$ denote Euclidean distances between points and Hausdorff distance between sets, respectively. Recall that a *curve* γ is a continuous function $\gamma : [a, b] \to \mathbb{R}^2$ for $a < b$. Except where specified otherwise, we take $a = 0, b = 1$. A plane graph $D = (V, \Gamma)$ is a drawing of a (planar) graph in the plane without intersections. In particular, $V = V(D)$ is a finite set of points in the plane $\mathbb{R}^2$. Γ is a finite collection of curves given by continuous functions $\gamma_i : [0, 1] \to \mathbb{R}^2$ such that no two such curves intersect except possibly at their endpoints, and such that if E is the set of pairs of endpoints:

$$E = \{\{\gamma_i(0), \gamma_i(1)\} \mid i = 1, \dots, |\Gamma|\},$$

then (V, E) is a graph. The faces of D are the connected components of $\mathbb{R}^2 \setminus \bigcup_{i=1}^{|\Gamma|} \mathrm{im}(\gamma_i)$ (here $\mathrm{im}(\gamma)$ is the image of the function γ), and we refer to the unique unbounded face as the *outer* face.

We state our results in terms of sequences of infinite plane graphs that get finer and finer with properties defined as follows.

Definition 4.1. A *lattice sequence* is a pair $(\{\Lambda_n\}, \rho)$, where $\{\Lambda_n\}$ is sequence of plane graphs with vertex sets $V(\Lambda_n) \subseteq \mathbb{R}^2$ for which there are corresponding dual plane graphs Λ_n^*, with each vertex $v \in V(\Lambda_n^*)$ a point in the corresponding face of Λ_n, such that for all $\varepsilon > 0$, there exists N such that for all $n > N$,

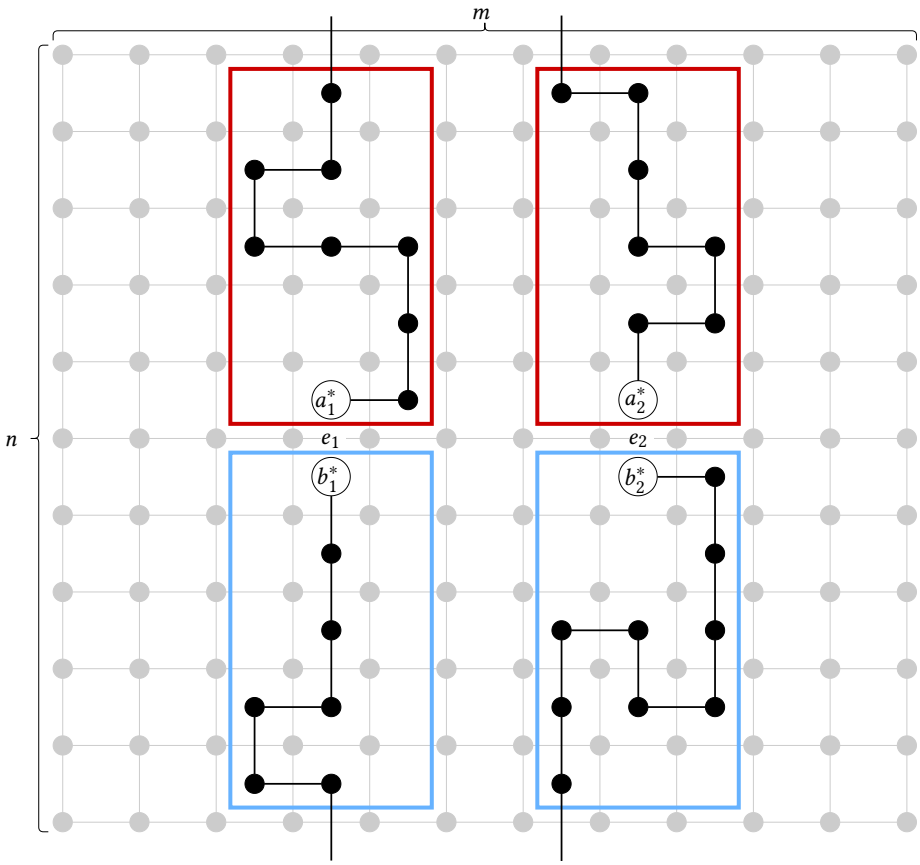

Figure 5: A possible run of the dual graph spanning tree sampling algorithm in the proof of Theorem 3.5 when m is odd. In this example, $m = 12$, $n = 11$, and $k = 3$. The primal graph G is depicted in gray, and the first four random walks in the dual graph G^* are depicted in black.

(a) For any adjacent pair $x, y \in V(\Lambda_n^*)$, $d(x, y) < \varepsilon$,
(b) For any $p \in \mathbb{R}^2$, the ball $B_\varepsilon(p)$ contains a vertex of Λ_n^*,
(c) For any dual vertex v and $\varepsilon > 0$ there is a division of the circle $C_{\varepsilon,v}$ of radius ε into arcs $A_1, \ldots, A_s$ each of length at most $\frac{1}{8}2\pi\varepsilon$, such that the following holds for each $i \in [s]$: In a simple random walk $v = v_0, v_1, v_2, \ldots$, with probability at least $\rho > 0$, letting j the first index where $d(v_0, v_j) > \varepsilon$, the straight line segment joining v_{j-1} with v_j passes through $C_{\varepsilon,v}$ in A_i.

This definition generalizes sequences of finer and finer grids. Specifically, the family of plane graphs $\left\{\mathbb{Z}^2, \frac{1}{2}\mathbb{Z}^2, \frac{1}{3}\mathbb{Z}^2, \ldots\right\}$ is a lattice sequence with $\rho = \frac{1}{8}$. Properties (a) and (b) say that, as we increase n, neighboring vertices in $\frac{1}{n}\mathbb{Z}^2$ become arbitrarily close, and the vertex set becomes arbitrarily dense. Property (c) holds using the partition of $C_{\varepsilon,v}$ obtained by drawing horizontal, vertical, and both 45 degree diagonal lines through v. By symmetry, a random walk from v is equally likely to exit through each of these 8 arcs.

The definition also applies to scalings of lattices like the triangular or hexagonal lattice, since, for example, property (c) holds for any sequence of lattices for which the scaling limit of random walk on the dual is Brownian motion. For the same reason, it applies a.s., for example, to the sequence where Λ_n denotes the Delauney triangulation of a Poisson cloud in $\mathbb{R}^2$ of rate n, see [28]. The choice of $\frac{1}{8}$ in the definition is made just for convenience; replacing it with any constant $< \frac{1}{2}$ would give exactly the same definition.

4.2 Statement of Results

In addition to lattice sequences, we will also consider a fixed bounded plane graph D that gives the partition structure we are looking to approximate. In doing this, we will need to restrict the infinite graphs in the lattice sequence to a reasonable bounded subgraph that falls inside D, and we do this as follows. Let D be a bounded plane graph, and fix $\delta > 0$ that will be chosen later in terms of D (in Lemma 4.5). Given plane graph Λ_n with dual Λ_n^* from a lattice sequence, and a cycle C^* in Λ^* at Hausdorff distance $< \delta$ from the outer face boundary of D, we let Ω_{D,Λ_n} be the subgraph of of Λ_n lying inside C^*. We let Ω^*_{D,Λ_n} be the subgraph of Λ_n^* induced by all vertices of C^* along with the vertices of Λ_n^* lying inside C^*. In this way, we can consider the planar dual of Ω_{D,Λ_n} to be Ω^*_{D,Λ_n} with *wired boundary condition*, where the entire cycle C^* (rather than a single dual vertex) corresponds to the outer face of Ω_{D,Λ_n}. (In

particular, for our proofs, we will run Wilson's algorithm on Ω^*_{D,Λ_n} with the cycle C^* identified as a single root.)

Given this plane graph D describing the partition structure we are looking to approximate, let $k+1$ be the number of faces and denote the k bounded faces by $\phi_1, \dots, \phi_k$. We say a partition of the graph $\Omega_{D,\Lambda}$ into connected components $C_1, \dots, C_k$ is *ε-compatible* with D if for all i and vertices $v \in \Omega_{D,\Lambda}$, the implication

$$v \in C_i \implies d(v, \phi_i) \le \varepsilon \tag{3}$$

holds. By subdividing edges if necessary, we will assume that D has no loops, so that $\gamma(0) \neq \gamma(1)$ for all $\gamma \in \Gamma(D)$.

For a lattice sequence $(\{\Lambda_n\}, \rho)$ and a probability space on the set of spanning trees of Ω_{D,Λ_n} and given $\varepsilon > 0$, we define the event $\mathcal{E}_{D,\Lambda_n,\varepsilon}$, which holds whenever there are $k-1$ edges whose removal from T results in a forest with components $C_1, \dots, C_k$ that is ε-compatible with D.

Theorem 4.2. *Let $(\{\Lambda_n\}, \rho)$ be a lattice sequence, let D be a plane graph with $k+1$ faces, and let Ω_{D,Λ_n} be as above. For the uniform probability space on the set of spanning trees of a graph Ω_{D,Λ_n}, we have that as $n \to \infty$, $\Pr(\mathcal{E}_{D,\Lambda_n,\varepsilon})$ is bounded below by a constant depending only on D and ε.*

As a consequence, if we draw the partition so that the parts contain approximately equal numbers of vertices, we can conclude that random trees are splittable into approximately balanced pieces with constant probability. This is possible so long as $\{\Lambda_n\}$ has the property that for any $\delta > 0$ and R, there is an $\varepsilon > 0$ so that every ε ball $B_\varepsilon(p)$ satisfies $|B_\varepsilon(p) \cap V(\Lambda_n)| \le \delta |B_R(0) \cap V(\Lambda_n)|$. In the case of grid graphs, for instance, we obtain the following corollary.

Corollary 4.3. *Fix $\varepsilon \ge 0$ and k a positive integer. Let m, n be positive integers such that $n \le m$, $k \mid m$, and $20/n \le \varepsilon \le 1/(3k)$. Let G be an $m \times n$ grid graph. There is a constant $C(k, \varepsilon)$ such that the probability a uniformly random spanning tree of G is (k, ε)-approximately splittable is at least $C(k, \varepsilon)$.*

4.3 Proof Sketch

We begin with a lemma generalizing the argument described in Figure 4 to lattice sequences.

Lemma 4.4. *Let $(\{\Lambda_n\}, \rho)$ be a lattice sequence, let γ_1, γ_2 be non-trivial curves in the plane, where γ_1 has length T and $\gamma_2(0) = \gamma_1(1)$. For all sufficiently small $\varepsilon > 0$, for any $v_0 \in \Lambda_n$ with $d(v_0, \gamma_1(0)) \le \varepsilon/2$, and for a random walk in Λ_n started from v_0, let $\mathcal{E}' = \mathcal{E}'_{n,v_0,\gamma_1,\gamma_2}$ be the event that the walk traverses an edge which intersects the curve γ_2 at a point within $\varepsilon/2$ of $\gamma_1(1)$ before ever reaching a vertex at distance $> \varepsilon$ from the curve γ_1. For all sufficiently large n,*

$$\Pr(\mathcal{E}'_{n,v_0,\gamma_1,\gamma_2}) \ge \rho^{100T/\varepsilon + 200}. \qquad \square$$

Fixing a plane graph $D = (V, \Gamma)$ as in Theorem 4.2 we have the following.

Lemma 4.5. *For $0 < \delta < \varepsilon$ sufficiently small, we have that:*

(A1) The distance between any two vertices $u, v \in V$ is at least 3ε.

(A2) If a_γ denotes the last time t at which $\gamma(t)$ is in the closed ball of radius ε about $\gamma(0)$, and b_γ denotes the first time t at which $\gamma(t)$ is in the closed ball of radius ε about $\gamma(1)$, the restricted curves $\bar{\gamma} = \gamma_{|[a_\gamma, b_\gamma]}$ are all at distance at least 3ε from each other.

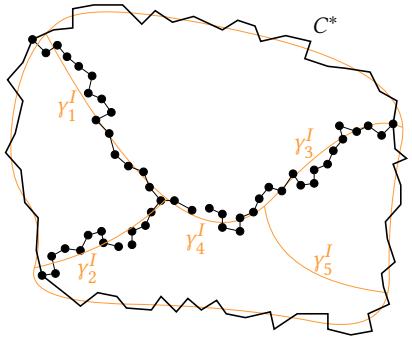

Figure 6: A run of the first four phases of Wilson's algorithm approximating the first four inner curves $\gamma_1^I, \gamma_2^I, \gamma_3^I, \gamma_4^I$. Note that paths corresponding to γ_2^I and γ_4^I each are missing an edge, and in particular, the edges present do not disconnect the interior of C^*. These missing edges are dual to the edges in the corresponding primal spanning tree whose removal would disconnect the tree into components approximating the faces of this drawing.

(A3) For any two points p, q in a common face of D and both at distance at least ε from any curve of D, there is a curve γ (not a curve of D) joining p to q whose distance to every curve of D is at least δ. $\square$

From here on, we let ε, δ be as promised by Lemma 4.5. We call a curve of D an *outer curve* if every point of the curve lies on the boundary of the outer face, and an *inner curve* if no point does, other than possibly its endpoints. Note that every curve must be one of these two types. We let Γ^I and Γ^O denote set of inner curves and outer curves, respectively. Since D is connected, we can order the inner curves of D as $\gamma_1^I, \dots, \gamma_{m_i}^I$ such that for all ℓ, the plane graph D_ℓ of D with edges $\Gamma_\ell = \Gamma^O \cup \{\gamma_1^I, \dots, \gamma_\ell^I\}$ and vertex set $V_\ell = \{\{\gamma(x) \mid \gamma \in \Gamma_\ell, x \in \{0,1\}\}$ is connected (see Figure 6). Moreover, without loss of generality we assume that the orientation of each curve is such that $\gamma_\ell^I(1)$ is a vertex of $\Gamma_{\ell-1}$ for all $\ell = 1, \dots, m_i$ (the curves are oriented "towards the outer face").

We prove the theorem by analyzing how Wilson's algorithm constructs spanning trees of Ω^*_{D,Λ_n} with wired boundary conditions, where the whole boundary cycle C^* is used as the root of Wilson's algorithm. In particular, it is equivalent to view Wilson's algorithm as building a unicylic graph, initialized with the boundary cycle of Ω^*_{D,Λ_n}. The freedom to choose the starts of loop-erased random walks in Wilson's algorithm allows us to use the following implementation, which determines starts by using additional random walks. We proceed in phases, creating a sequence of trees T_i^j where the subscript i denotes the phase and the superscript j denotes the step within phase i. Within phase i, we alternate a loop-erased random walk from a vertex outside T_i^j to a vertex in T_i^j that gets added to the tree to obtain T_i^{j+1} (one step of Wilson's algorithm) with a random walk among the vertices in T_i^{j+1} until a vertex outside T_i^{j+1} is reached (choosing the starting point for the next step of Wilson's algorithm). Here we describe the procedure for one phase of this

process; the particular choices of source vertex and target subsets that will be useful for our purposes will be specified below.

(1) At the beginning of each phase i, we have an existing tree T_{i-1} that has already been built. (For $i = 1$, T_0 consists of just the root vertex). We choose a source vertex v and target subset $U_i \subseteq T_{i-1}$ for this phase, and initialize $T_{i-1}^0 = T_{i-1}$.
(2) We begin each step of this phase with T_{i-1}^{j-1} (at the beginning of the phase, for $j = 1$) and a source vertex. We do one of two things according to whether the source belongs to the tree T_{i-1}^{j-1}:
   (a) If the source vertex is not in T_{i-1}^{j-1}, we conduct a loop-erased random walk from the source until it hits T_{i-1}^{j-1} at a vertex u. This loop-erased random walk is added to T_{i-1}^{j-1} to create T_{i-1}^{j}. If $u \in U_i$ this phase ends, and we set $T_i = T_{i-1}^{j}$. Otherwise, we increment j, and continue this phase with u as the new source vertex (we will be in case (b) next).
   (b) If the source vertex is in T_{i-1}^{j-1}, we take a random walk from the source until we reach a vertex u outside of T_{i-1}, and then increment j and restart this step from the vertex u as the new source vertex, and $T_{i-1}^{j} = T_{i-1}^{j-1}$ (we will be in case (a) next).
(3) The previous loop continues until either the target is eventually hit by an instance of the loop-erased random walk, or the entire spanning tree is completed.

Note that we can use a single random walk W_i from $v \notin T_{i-1}$ to implement each phase of the algorithm (with loop erasure while in case (2a), and without loop erasure while in case (2b)). In particular, with this implementation, we have the following observation for general graphs:

Observation 1. *Suppose that we run the implementation of Wilson's algorithm above on a graph G, and have built the tree T_ℓ after the first ℓ phases.*

(a) For a connected set of vertices S, if the walk $W_{\ell+1}$ begins from a vertex $v \in S$ and ends phase $\ell+1$ by hitting T_ℓ for the first time at the target $U_{\ell+1} \subseteq S$ and without leaving S, then after this phase, there is a path P in $T_{\ell+1}$ joining v to $U_{\ell+1}$.
(b) For a connected set of vertices S, suppose the walk $W_{\ell+1}$ begins from a vertex in S that is adjacent to $v \in T_\ell \cap S$ and ends phase $\ell+1$ by hitting the target $U_{\ell+1} \subseteq S$ without leaving S. Then after this phase, there is a path P from a vertex $v' \in (S \cap T_\ell) \setminus U_{\ell+1}$ to $U_{\ell+1}$, all of whose vertices belong to S, and all but one of whose edges belong to $T_{\ell+1}$. □

In particular, applied to our situation, using as S the set of vertices close to a given curve, we obtain the following:

Observation 2. *Suppose that we run the implementation of Wilson's algorithm above on the dual graph Ω^*_{D,Λ_n}, and have built the tree T_ℓ after the first ℓ phases. Then:*

(a) If the walk $W_{\ell+1}$ begins from a vertex v and ends phase $\ell+1$ by hitting T_ℓ for the first time at the target $U_{\ell+1} = T_\ell \cap B(\delta, \gamma_{\ell+1}(1))$ while also staying within distance δ of the curve $\gamma_{\ell+1}$, then after this phase, there is a path P in $T_{\ell+1}$ joining v to $U_{\ell+1}$ whose vertices are all within distance δ from the curve $\gamma_{\ell+1}$.
(b) Suppose the walk $W_{\ell+1}$ begins from a vertex adjacent to a vertex $v \in T_\ell \cap B(\delta, \gamma_{\ell+1}(0))$ and ends phase $\ell+1$ by hitting the target $U_{\ell+1} = T_\ell \cap B(\delta, \gamma_{\ell+1}(1))$ while also staying within distance δ of the curve $\gamma_{\ell+1}$ throughout the phase. Suppose all vertices of T_ℓ that are within δ of the curve $\gamma_{\ell+1}$ are in either $B(\delta, \gamma_{\ell+1}(0))$ or $B(\delta, \gamma_{\ell+1}(1))$. Then after this phase, there is a path P in the dual graph joining some vertex in $T_\ell \cap B(\delta, \gamma_{\ell+1}(0))$ to $U_{\ell+1}$ whose vertices are all within distance δ from the curve $\gamma_{\ell+1}$, and such that all but at most one edge of P belongs to the tree $T_{\ell+1}$.

Proof. This follows from Observation 1, with $G = \Omega^*_{D,\Lambda_n}$ and S to be all the vertices of Ω^*_{D,Λ_n} within distance δ of the curve $\gamma_{\ell+1}$. □

Using Observation 2, we complete the proof as follows. For a plane graph D, let $f(D)$ be the number of interior faces of D. We say a (not-necessarily-spanning) tree T in the dual graph Ω^*_{D,Λ_n} *δ-corresponds* to a plane graph D with inner curves $\gamma_1^I, \ldots, \gamma_\ell^I$ if there are paths $P_1, \ldots, P_\ell$ in Ω^*_{D,Λ_n}, such that for each i:

(a) For $f(D) - 1$ of the paths, all but one edge of P_i belongs to the tree T, while for the rest of the paths, the whole path belongs to T. Intuitively, T is the dual tree that partitions the lattice, and the P_i are obtained by adding the "missing" edges whose duals are the split edges of the tree in the primal graph, such as the gap in the center of Figure 6.
(b) $P_i \cap P_j \neq \emptyset$ if and only if γ_i^I and γ_j^I share a common endpoint.
(c) For any point p, if $\gamma_{i_1}^I, \ldots, \gamma_{i_s}^I$ are all the curves of D that have p as an endpoint, then the union of the paths $P_{i_1}, \ldots, P_{i_s}$ is a tree.
(d) Every vertex of P_i is within distance δ of the curve γ_i^I.

This definition then allows the following lemma, which applies to each plane graph D_ℓ in the sequence constructed above.

Lemma 4.6. *Let T be a tree in the dual graph Ω^*_{D,Λ_n}, and let H_T be the spanning subgraph of the primal graph $\Omega_{D,h}$ obtained by removing from Ω_{D,Λ_n} all the edges e for which $e^* \in T$. If T δ-corresponds to D_ℓ, then there are $f(D_\ell) - 1$ edges of H whose removal results in connected components $C_1, \ldots, C_k$ whose induced partition of Ω_{D,Λ_n} is ε-compatible with D_ℓ.* □

Let us now use Lemma 4.6 and induction on the sequence of plane graphs D_ℓ to prove the Theorem. Initially, tree T_0 consisting of just the root vertex of Ω^*_{D,Λ_n} (equivalently, of just the wired boundary cycle of Ω^*_{D,Λ_n}) trivially δ-corresponds to the plane graph D_0 with no interior curves. Having already constructed a tree $T_{\ell-1}$ that δ-corresponds to $D_{\ell-1}$, we begin another phase of Wilson's algorithm from a vertex $v \in \Omega_{D,\Lambda_n}$. If $T_{\ell-1} \cap B(\delta, \gamma_\ell^I(0)) = \emptyset$, we begin from a vertex v at minimum distance from $\gamma_\ell^I(0)$ (call this Case A), and note v will not be in $T_{\ell-1}$. Otherwise, if $T_{\ell-1} \cap B(\delta, \gamma_\ell^I(0)) = \emptyset$, we begin at a vertex v adjacent to any vertex in this set (Case B). In both cases, we use the target $U_\ell = T_{\ell-1} \cap B(\delta, \gamma_\ell^I(1))$. Note that when $\gamma_\ell^I(1)$ is incident on the outer face of D, the target U_ℓ contains a portion of the boundary cycle of Ω^*_{D,Λ_n} within distance δ of $\gamma_\ell^I(1)$. By Lemma 4.4, with constant probability, the walk W_ℓ for this phase will hit the target U_ℓ while staying within distance δ of γ_ℓ^I. Thus, by Observation 2, and the fact that we are in Case A instead of Case B if and only if $f(D_\ell) = f(D_{\ell-1})$ (as in Euler's formula), at the end of the phase, the tree T_ℓ δ-corresponds to D_ℓ.

In particular, after phase $\ell = m_i$ (recall m_i is the total number of interior curves in D) we have that with constant probability, our tree T_{m_i} consists only of the root and vertices within distance δ of the internal curves of D, and that there is a collection of paths $P_1, \ldots, P_{m_i}$ satisfying properties (a), (b), (c), and (d) above, and additionally that all but precisely $f(D) - 1 = k - 2$ of the paths belong entirely to three T_{m_i}.

From here, we complete Wilson's algorithm with arbitrary choices for starting vertices to produce a final tree T, which still contains all but at most one edge of each path P_i, and the whole path in all but precisely $k - 1$ cases. In particular, in the primal graph, the tree T corresponds to a tree from which $k - 1$ edges can be deleted, to produce a partition that is ε-compatible with the drawing D. □

Corollary 4.3 follows with some additional observations about how the parameters of Theorem 4.2 applies to grids.

ACKNOWLEDGEMENTS

We thank Jacob Calvert for pointing out Lemma 3.4 could be proved more simply using the Cauchy-Schwarz inequality. This material is based upon work supported by the National Science Foundation under Grant No. DMS-1928930 and by the Alfred P. Sloan Foundation under grant G-2021-16778, while the authors were in residence at the Simons Laufer Mathematical Sciences Institute (formerly MSRI) in Berkeley, California, during the Fall 2023 semester. S. Cannon is supported in part by NSF CCF-2104795. W. Pegden is supported in part by NSF DMS-2054503. J. Tucker-Foltz is supported in part by a Google PhD Fellowship.

REFERENCES

[1] Nima Anari, Kuikui Liu, Shayan Oveis Gharan, Cynthia Vinzant, and Thuy-Duong Vuong. 2021. Log-Concave Polynomials IV: Approximate Exchange, Tight Mixing Times, and near-Optimal Sampling of Forests. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing* (Virtual, Italy) *(STOC 2021)*. Association for Computing Machinery, New York, NY, USA, 408–420. https://doi.org/10.1145/3406325.3451091

[2] Eric Autry, Daniel Carter, Gregory Herschlag, Zach Hunter, and Jonathan C. Mattingly. 2023. Metropolized Forest Recombination for Monte Carlo Sampling of Graph Partitions. *SIAM J. Appl. Math.* 83, 4 (2023), 1366–1391. https://doi.org/10.1137/21M1418010

[3] Eric A. Autry, Daniel Carter, Gregory Herschlag, Zach Hunter, and Jonathan C. Mattingly. 2021. Metropolized Multiscale Forest Recombination for Redistricting. *Multiscale Modeling & Simulation* 19, 4 (2021), 1885–1914. https://doi.org/10.1137/21M1406854

[4] Amariah Becker, Daryl R. DeFord, Dara Gold, Sam Hirsch, Mary E. Marshall, and Jessica Ring Amunson. 2022. Brief of Computational Redistricting Experts as *Amici Curiae* in support of Appellees and Respondents. (2022). Merrill v. Milligan; Merrill v. Caster; United States Supreme Court, 2022. Nos. 21-1086, 21-1087.

[5] Amariah Becker, Moon Duchin, Dara Gold, and Sam Hirsch. 2021. Computational Redistricting and the Voting Rights Act. *Election Law Journal: Rules, Politics, and Policy* 20, 4 (2021), 407–441. https://doi.org/10.1089/elj.2020.0704 arXiv:https://doi.org/10.1089/elj.2020.0704

[6] Gerdus Benadè, Ruth Buck, Moon Duchin, Dara Gold, and Thomas Weighill. 2021. Ranked Choice Voting and Proportional Representation. (2021). Working Paper. Available at https://ssrn.com/abstract=3778021.

[7] Sophia Caldera, Daryl DeFord, Moon Duchin, Samuel C. Gutekunst, and Cara Nix. 2020. Mathematics of Nested Districts: The Case of Alaska. *Statistics and Public Policy* (2020), 1–22.

[8] Sarah Cannon, Moon Duchin, Dana Randall, and Parker Rule. 2022. Spanning tree methods for sampling graph partitions. (2022). Preprint. Available at https://arxiv.org/abs/2210.01401.

[9] Moses Charikar, Paul Liu, Tianyu Liu, and Thuy-Duong Vuong. 2022. On the Complexity of Sampling Redistricting Plans. (2022). Preprint. Available at https://arxiv.org/abs/2206.04883.

[10] Jowei Chen, Christopher S. Elmendorf, Ruth Greenwood, Theresa J. Lee, Nicholas O. Stephanolpoulos, and Christopher S. Warshaw. 2022. Brief of *Amici Curiae* Professors Jowei Chen, Christopher S. Elmendorf, Nicholas O. Stephanolpoulos, and Christopher S. Warshaw in support of Appellees/Respondents. (2022). Merrill v. Milligan; Merrill v. Caster; United States Supreme Court, 2022. Nos. 21-1086, 21-1087.

[11] Jowei Chen and Nicholas O. Stephanopoulos. 2021. The Race-Blind Future of Voting Rights. *The Yale Law Journal* 130, 4 (2021).

[12] Jeanne N. Clelland, Nicholas Bossenbroek, Thomas Heckmaster, Adam Nelson, Peter Rock, and Jade VanAusdall. 2021. Compactness statistics for spanning tree recombination. (2021). Preprint. Available at https://arxiv.org/abs/2103.02699.

[13] Aloni Cohen, Moon Duchin, JN Matthews, and Bhushan Suwal. 2021. Census TopDown: The Impacts of Differential Privacy on Redistricting. In *2nd Symposium on Foundations of Responsible Computing (FORC 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 192)*, Katrina Ligett and Swati Gupta (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 5:1–5:22. https://doi.org/10.4230/LIPIcs.FORC.2021.5

[14] Mary Cryan, Heng Guo, and Giorgos Mousa. 2021. Modified log-Sobolev inequalities for strongly log-concave distributions. *The Annals of Probability* 49, 1 (2021), 506 – 525. https://doi.org/10.1214/20-AOP1453

[15] Daryl DeFord, Moon Duchin, and Justin Solomon. 2020. A Computational Approach to Measuring Vote Elasticity and Competitiveness. *Statistics and Public Policy* 7, 1 (2020), 69–86.

[16] Daryl DeFord, Moon Duchin, and Justin Solomon. 2021. Recombination: A Family of Markov Chains for Redistricting. *Harvard Data Science Review* 3, 1 (2021).

[17] Moon Duchin, Jeanne Clelland, Daryl DeFord, Jordan Ellenberg, Tyler Jarvis, Nestor Guillen, Dmitry Morozov, Elchanan Mossel, Dana Randall, Justin Solomon, Ari Stern, Guy-Uriel Charles, Luis Fuentes-Rohwer, Anna Dorman, Dana Paikowsky, and Robin Tholin. 2019. Amicus Brief of Mathematicians, Law Professors, and Students in Support of Appellees and Affirmance. (2019). Rucho v. Common Cause, United States Supreme Court, 2019. Available at https://mggg.org/SCOTUS-MathBrief.pdf.

[18] Moon Duchin and Douglas M. Spencer. 2021. Models, Race, and the Law. *The Yale Law Journal* 130 (2021), 744–797.

[19] Benjamin Fifield, Michael Higgins, Kosuke Imai, and Alexander Tarr. 2020. Automated Redistricting Simulation Using Markov Chain Monte Carlo. *Journal of Computational and Graphical Statistics* 29, 4 (2020), 715–728.

[20] Alan Frieze and Wesley Pegden. 2023. Subexponential mixing for partition chains on grid-like graphs. In *Proceedings of the 2023 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 3317–3329. https://doi.org/10.1137/1.9781611977554.ch127

[21] Gregory Herschlag, Han Sung Kang, Justin Luo, Christy Vaughn Graves, Sachet Bangia, Robert Ravier, and Jonathan C. Mattingly. 2020. Quantifying Gerrymandering in North Carolina. *Statistics and Public Policy* 7, 1 (2020), 452–479. https://doi.org/10.1080/2330443X.2020.1796400

[22] Christopher T. Kenny, Shiro Kuriwaki, Cory McCartan, Evan T. R. Rosenman, Tyler Simko, and Kosuke Imai. 2021. The use of differential privacy for census data and its impact on redistricting: The case of the 2020 U.S. Census. *Science Advances* 7, 41 (2021), eabk3283. https://doi.org/10.1126/sciadv.abk3283 arXiv:https://www.science.org/doi/pdf/10.1126/sciadv.abk3283

[23] David A Levin and Yuval Peres. 2017. *Markov chains and mixing times* (second edition ed.). American Mathematical Society.

[24] Jonathan C. Mattingly. 2021. Expert Report on the North Carolina State Legislature and Congressional Redistricting (Corrected Version). (2021). Harper v. Hall, North Carolina Superior Court, Wake County, No. 21-cvs-500085.

[25] Cory McCartan and Kosuke Imai. 2020. Sequential Monte Carlo for Sampling Balanced and Compact Redistricting Plans. Submitted. Available at https://arxiv.org/abs/2008.06131.

[26] Michael Mitzenmacher and Eli Upfal. 2005. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511813603

[27] Ariel D. Procaccia and Jamie Tucker-Foltz. 2022. Compact Redistricting Plans Have Many Spanning Trees. *ACM-SIAM Symposium on Discrete Algorithms (SODA)* (2022).

[28] Arnaud Rousselle. 2015. Quenched invariance principle for random walks on Delaunay triangulations. *Electron. J. Probab* 20, 33 (2015), 1–32.

[29] Kristopher Tapp. 2021. Spanning tree bounds for grid graphs. *arXiv preprint arXiv:2109.05987* (2021).

[30] Jamie Tucker-Foltz. 2023. Locked Polyomino Tilings. (2023). Preprint. Available at https://arxiv.org/abs/2307.15996.

[31] David Bruce Wilson. 1996. Generating Random Spanning Trees More Quickly than the Cover Time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing (STOC)*. 296–303. https://doi.org/10.1145/237814.237880

[32] Zhanzhan Zhao, Cyrus Hettle, Swati Gupta, Jonathan Christopher Mattingly, Dana Randall, and Gregory Joseph Herschlag. 2022. Mathematically Quantifying Non-Responsiveness of the 2021 Georgia Congressional Districting Plan. In *Equity and Access in Algorithms, Mechanisms, and Optimization (EAAMO '22)*. Association for Computing Machinery, Article 15, 11 pages.

Received 13-NOV-2023; accepted 2024-02-11