ANNALES
DE L'INSTITUT
HENRI
POINCARÉ
PROBABILITÉS
ET STATISTIQUES

A reverse Aldous–Broder algorithm

Yiping Hu[a,1], Russell Lyons[b,2] and Pengfei Tang[b,c,3]

[a]*Department of Mathematics, University of Washington, Seattle, USA. E-mail: huypken@uw.edu*
[b]*Department of Mathematics, Indiana University, Bloomington, USA. E-mail: rdlyons@indiana.edu*
[c]*Current affiliation: Department of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel. E-mail: pengfeitang@mail.tau.ac.il*

Abstract. The Aldous–Broder algorithm provides a way of sampling a uniformly random spanning tree for finite connected graphs using simple random walk. Namely, start a simple random walk on a connected graph and stop at the cover time. The tree formed by all the first-entrance edges has the law of a uniform spanning tree. Here we show that the tree formed by all the last-exit edges also has the law of a uniform spanning tree. This answers a question of Tom Hayes and Cris Moore from 2010. The proof relies on a bijection that is related to the BEST theorem in graph theory. We also give other applications of our results, including new proofs of the reversibility of loop-erased random walk, of the Aldous–Broder algorithm itself, and of Wilson's algorithm.

Résumé. L'algorithme d'Aldous–Broder fournit un moyen d'échantillonner un arbre couvrant aléatoire uniforme pour des graphes connexes finis en utilisant une marche aléatoire simple. Il procède de la façon suivante : commençons une marche aléatoire simple sur un graphe connexe et arrêtons-nous au temps de couverture. L'arbre formé par toutes les arêtes de première entrée a la loi d'un arbre couvrant uniforme. Nous montrons ici que l'arbre formé par toutes les arêtes de dernière sortie a également la loi d'un arbre couvrant uniforme. Cela répond à une question de Tom Hayes et Cris Moore de 2010. La preuve repose sur une bijection qui est liée au théorème BEST de la théorie des graphes. Nous donnons également d'autres preuves de nos résultats, comprenant de nouvelles preuves de la réversibilité de la marche aléatoire à boucles effacées, de l'algorithme d'Aldous–Broder lui-même, et de l'algorithme de Wilson.

1. Introduction

Let $G = (V, E)$ be a finite connected graph and let $\mathcal{T}$ be the set of spanning trees of G. Obviously $\mathcal{T}$ is a finite set. The *uniform spanning tree* is the uniform measure on $\mathcal{T}$, which is denoted by $\mathsf{UST}(G)$.

In the late 1980s, Aldous [1] and Broder [4] found an algorithm to generate a uniform spanning tree using simple random walk on G; both authors thank Persi Diaconis for discussions. The algorithm is now called the Aldous–Broder algorithm. It generates a sample of the uniform spanning tree as follows.

Start a simple random walk $(X_n)_{n \geq 0}$ on G and stop at the cover time, i.e., stop when it first visits all the vertices of G. Collect all the first-entrance edges, i.e., edges (X_n, X_{n+1}) for $n \geq 0$ such that $X_{n+1} \neq X_k$ for all $k \leq n$. These edges form a random spanning tree, T. Then this random tree T has the law $\mathsf{UST}(G)$. The simple random walk can start from any vertex of G, and so X_0 can have any initial distribution.

The proof of this result depends on ideas related to the Markov chain tree theorem. Namely, let $(X_n)_{n \in \mathbb{Z}}$ be a biinfinite stationary random walk on G. Let the last exit from a vertex x before time 0 be $\lambda(x) := \max\{n < 0; X_n = x\}$. Then the random spanning tree $\{(x, X_{\lambda(x)+1}); x \in V \setminus \{X_0\}\}$ of last-exit edges has the uniform distribution. One then proves the validity of the Aldous–Broder algorithm by reversing time and using reversibility of simple random walk; see, e.g., [8,

Section 4.4] for a proof. The algorithmic aspects were studied by [4], while theoretical implications were studied by [1] and, later, Pemantle [10].

In 2010 ("domino" forum email discussion, 9 Sep.; published in [9, p. 645]), Tom Hayes and Cris Moore asked whether the tree formed by all the last-exit edges starting at time 0 and ending at the cover time has the law of the uniform spanning tree. Here we give a positive answer to this question.

Theorem 1.1. *Let G be a finite, connected graph. Start a simple random walk on G and stop at the cover time. Let T be the random tree formed by all the last-exit edges. Then T has the law $\mathsf{UST}(G)$.*

Remark 1.2. As we will show, the corresponding result holds for every finite, connected network and its associated network random walk.

We call this algorithm the *reverse Aldous–Broder* algorithm.

Our proof shows a remarkable strengthening of this equality: Notice that the Aldous–Broder algorithm run up to time n gives a tree T_n on the vertices $\{X_k; k \leq n\}$. The evolution of the rooted tree (T_n, X_n) to (T_{n+1}, X_{n+1}) is given by a Markov chain. For this chain, T_{n+1} contains T_n and may be equal to T_n. Furthermore, if τ_{cov} denotes the cover time, then $T_n = T_{\tau_{\mathrm{cov}}}$ for all $n \geq \tau_{\mathrm{cov}}$. Similarly, the reverse Aldous–Broder algorithm run up to time n gives a tree $\overline{T}_n$ on the vertices $\{X_k; k \leq n\}$. The evolution of $(\overline{T}_n, X_n)$ to $(\overline{T}_{n+1}, X_{n+1})$ is also given by a Markov chain. For this chain, $\overline{T}_{n+1}$ *need not* contain $\overline{T}_n$. In addition, $\overline{T}_n$ need not equal $\overline{T}_{\tau_{\mathrm{cov}}}$ for $n > \tau_{\mathrm{cov}}$. Our theorem above is that the distributions of $T_{\tau_{\mathrm{cov}}}$ and $\overline{T}_{\tau_{\mathrm{cov}}}$ are the same. Our strengthening will show that for every n, the distributions of T_n and $\overline{T}_n$ are the same, even though the two Markov chains have different transition probabilities. Moreover, for every n, the distributions of $T_{n \wedge \tau_{\mathrm{cov}}}$ and $\overline{T}_{n \wedge \tau_{\mathrm{cov}}}$ are the same. These are both proved as Corollary 2.2. It is also true that T_τ has the same distribution as $\overline{T}_\tau$ for some other stopping times τ, such as the hitting time of a vertex; see Proposition 2.6. Using this fact together with the Aldous–Broder algorithm, one obtains the well-known result that the path between two vertices in the $\mathsf{UST}(G)$ has the same distribution as the loop-erased random walk; see Corollary 2.5.

In outline, our proof that T_n and $\overline{T}_n$ have the same distribution proceeds as follows. We fix a starting vertex x and an ending vertex y at time n. For a path γ of length n from x to y, there corresponds a first-entrance tree $F(\gamma)$ and a last-exit tree $L(\gamma)$. We define a probability-preserving permutation Φ of the set of paths of length n from x to y such that $F = L \circ \Phi$. This easily leads to our results. The key, then, is to define Φ. For this purpose, we consider two types of multi-digraphs, one with a linear ordering on each set of incoming edges and the other with a linear ordering on each set of outgoing edges. The former type also determines a spanning tree (of the multi-digraph) directed away from x via first-entrance edges, whereas the latter type determines a spanning tree directed toward y via last-exit edges. A path γ naturally yields a multi-digraph $\hat{F}(\gamma)$ of the first type and a multi-digraph $\hat{L}(\gamma)$ of the second type. Reversing all edges of one type of multi-digraph yields one of the other type, except that the edges on the path from x to y in the spanning tree are not reversed. The correspondence of multi-digraphs to paths then allows us to define Φ.

Our proof is closely related to the BEST theorem in graph theory, named after the initials of de Bruijn and van Aardenne-Ehrenfest [13] and Smith and Tutte [12]; we draw the connection below in Theorem 3.1. Interestingly, by using the BEST theorem, we are able to provide a new proof of the validity of the Aldous–Broder algorithm. This is done in Theorem 3.3: In outline, we note that the BEST theorem shows that given the transition counts of a random walk path of length n, each directed tree is equally likely to correspond to the path. A simple formula connects the transition counts with the likelihood of a tree, and letting $n \to \infty$ yields a spanning tree whose probability is then easily deduced.

The BEST theorem has been used before to study statistics for Markov chains (see the survey by Billingsley [3]) and to characterize Markov exchangeable sequences, as observed by Zaman [15].

2. Proof of the main theorem

We allow weights $w \colon E \to (0, \infty)$ on the edges, so that (G, w) is a network, and the corresponding *weighted uniform spanning tree* measure puts mass proportional to $\prod_{e \in T} w(e)$ on a spanning tree T. We could also allow parallel edges and loops in G, but this would merely require more complicated notation, so we assume that G is a simple graph. Write $w(x)$ for the sum of $w(e)$ over all edges e incident to x.

Let x and y be two vertices of G. If there is an edge in G connecting x and y, then we write $x \sim y$. We call $\gamma = (v_0, \ldots, v_n)$ a *path* (or a walk) on G from v_0 to v_n if $v_{i-1} \sim v_i$ for $i = 1, \ldots, n$, and $|\gamma| := n$ is called the *length* of the path γ.

Let $\mathcal{P}_n^{x,y}$ denote the set of all paths in G from x to y with length n. Simple random walk is the Markov chain on V with transition probabilities $p(x, y) := \frac{1}{\deg(x)} \mathbf{1}_{\{x \sim y\}}$. The network random walk on (G, w) has instead the transition

probabilities $p(x, y) := \frac{w(x,y)}{w(x)} \mathbf{1}_{\{x \sim y\}}$. All results quoted earlier hold for network random walks, not only simple random walks, and for the corresponding random spanning trees. In particular, the Aldous–Broder algorithm holds in this more general context: the network random walk generates the weighted uniform spanning tree by collecting the first-entrance edges up to the cover time [8, Section 4.4]. We henceforth assume that our graphs are weighted and use the corresponding network random walks; the weighted uniform spanning tree measure is still denoted by $\mathsf{UST}(G)$.

Let $\mathscr{T}$ be the set of all subtrees of G, including those that are not necessarily spanning.

For a path $\gamma = (v_0, \ldots, v_n) \in \mathcal{P}_n^{x,y}$, where $v_0 = x$ and $v_n = y$, we write $V(\gamma) = \{v_0, \ldots, v_n\}$ for the set of vertices of γ. For each $u \in V(\gamma) \setminus \{x\}$, there is a smallest index $i \geq 1$ such that $v_i = u$; we call the edge (v_{i-1}, v_i) the *first-entrance edge* to u.

Define the *first-entrance operator* $F: \mathcal{P}_n^{x,y} \to \mathscr{T}$ by setting $F(\gamma)$ to be the tree formed by all the first-entrance edges to vertices in $V(\gamma) \setminus \{x\}$.

Similarly, for each $u \in V(\gamma) \setminus \{y\}$, there is a largest index $i \leq n-1$ such that $v_i = u$, and we call the edge $e = (v_i, v_{i+1})$ the *last-exit edge* of u. Define the *last-exit operator* $L: \mathcal{P}_n^{x,y} \to \mathscr{T}$ by setting $L(\gamma)$ to be the tree formed by all the last-exit edges of vertices in $V(\gamma) \setminus \{y\}$.

For a finite path $\gamma = (v_0, \ldots, v_n)$ in G with length n started at v_0, we write

$$p(\gamma) := \mathbb{P}_{v_0}\big[(X_0, \ldots, X_n) = \gamma\big] = \frac{\prod_{k=0}^{n-1} w(v_k, v_{k+1})}{\prod_{k=0}^{n-1} w(v_k)}, \tag{2.1}$$

where $\mathbb{P}_{v_0}$ denotes the law of the network random walk on (G, w) started from v_0.

The proof will go as follows: In Section 2.1, we state the key lemma (Lemma 2.1) and use it to derive Theorem 1.1, together with other consequences. In Section 2.2, we start the proof of the key lemma by establishing the correspondence between random walk paths and colored multi-digraphs. We finish the proof in Section 2.3.

2.1. *The main lemma and its consequences*

The main ingredient for proving our theorem is the following lemma, which can be viewed as an extension of Proposition 2.1 of [6].

Lemma 2.1. *For every* $x, y \in V(G)$ *and* $n \geq 0$, *there is a bijection* $\Phi: \mathcal{P}_n^{x,y} \to \mathcal{P}_n^{x,y}$ *such that* $F = L \circ \Phi$ *and*

$$\forall \gamma \in \mathcal{P}_n^{x,y} \quad p(\gamma) = p\big(\Phi(\gamma)\big). \tag{2.2}$$

Moreover, Φ *preserves the number of times each vertex is visited as well as the number of times that each* (*unoriented*) *edge is crossed.*

It follows that Φ is also a measure-preserving bijection on each subset of $\mathcal{P}_n^{x,y}$ specified by how many times each edge is crossed or which vertices are visited. For example, such a subset is the set of paths such that every vertex is visited and y is visited only once and only after all other vertices are visited.

Before proving Lemma 2.1, we show how it gives our claims.

Consider the network random walk $(X_n)_{n=0}^{\infty}$ on G with arbitrary initial distribution, and recall that τ_{cov} denotes the cover time of the graph G. Write $U \overset{\mathscr{D}}{=} V$ when U and V have the same distribution.

Corollary 2.2. *We have for all* $n \in \mathbb{N}$,

(i) $F((X_0, \ldots, X_n)) \overset{\mathscr{D}}{=} L((X_0, \ldots, X_n))$ *and*

(ii) $F((X_0, \ldots, X_{\tau_{\mathrm{cov}} \wedge n})) \overset{\mathscr{D}}{=} L((X_0, \ldots, X_{\tau_{\mathrm{cov}} \wedge n}))$.

Proof. This is immediate from Lemma 2.1 and the remarks following it. $\qquad \square$

Proof of Theorem 1.1. Letting $n \to \infty$ in Corollary 2.2(ii) and noting that $\mathbb{P}[\tau_{\mathrm{cov}} \geq n] \to 0$ as $n \to \infty$, one obtains that

$$F\big((X_0, \ldots, X_{\tau_{\mathrm{cov}}})\big) \overset{\mathscr{D}}{=} L\big((X_0, \ldots, X_{\tau_{\mathrm{cov}}})\big). \tag{2.3}$$

By the Aldous–Broder algorithm, $F((X_0, \ldots, X_{\tau_{\mathrm{cov}}}))$ has the law of $\mathsf{UST}(G)$, and thus so does $L((X_0, \ldots, X_{\tau_{\mathrm{cov}}}))$. $\qquad \square$

The equation (2.3) holds for more general stopping times (but not all stopping times). Here we give a sufficient condition.

Let $C_e^n := \#\{j; 0 \le j \le n-1, \{X_j, X_{j+1}\} = e\}$ denote the total number of crossings of the edge e in both directions up to time n. If $I_y := \sum_{e \ni y} C_e^n$, then the number of visits to y by time n is $\lfloor (I_y + 1)/2 \rfloor$. Furthermore, any vertex $y \neq x$ is visited at time n iff I_y is odd, and x is visited at time n iff I_x is even. If τ is a finite stopping time such that

$$\forall n \ge 0 \quad \{\tau = n\} \in \sigma\big(C_e^n; e \in E(G)\big), \tag{2.4}$$

then τ satisfies the corresponding version of equation (2.3):

$$F\big((X_0, \ldots, X_\tau)\big) \overset{\mathscr{D}}{=} L\big((X_0, \ldots, X_\tau)\big). \tag{2.5}$$

For example, if K is a finite subset of $V(G)$ and τ is the hitting time of K, then τ satisfies (2.5).

This implies that the path between two fixed vertices in the uniform spanning tree of a graph has the same law as the loop-erased random walk from one of those vertices to the other. We state this as Corollary 2.5 below, which was first proved by Pemantle [10] using Proposition 2.1 of Lawler [6]. Using Corollary 2.5 and the spatial Markov property of UST(G), one can immediately get Wilson's algorithm [14] for sampling UST(G) (in networks). This was observed by Wilson [14] and Propp and Wilson [11]. We first recall the definition of loop-erased random walk.

Definition 2.3. Let $\gamma = (v_0, v_1, \ldots, v_n)$ be a finite path on G. We define the *loop-erasure* of γ to be the self-avoiding path LE$(\gamma) = (u_0, \ldots, u_m)$ obtained by erasing cycles on γ in the order they appear. Equivalently, we set $u_0 := v_0$ and $l_0 := \max\{j; 0 \le j \le n, v_j = u_0\}$. If $l_0 = n$, we set $m := 0$, LE$(\gamma) := (u_0)$ and terminate. Otherwise, we set $u_1 := v_{l_0+1}$ and $l_1 := \max\{j; l_0 \le j \le n, v_j = u_1\}$. If $l_1 = n$, we set $m := 1$, LE$(\gamma) := (u_0, u_1)$ and terminate. Continue in this way: while $l_k < n$, set $u_{k+1} := v_{l_k+1}$ and $l_{k+1} := \max\{j; l_k \le j \le n, v_j = u_{k+1}\}$. Since γ is a finite path, we must terminate after at most n steps and get the loop-erased path LE(γ).

Definition 2.4. Suppose that u and v are distinct vertices in the graph G. Start a network random walk on G from u and stop at the first visit of v. The loop-erasure of this random walk path is called the *loop-erased random walk* from u to v.

Corollary 2.5. *Let x and y be two distinct vertices in G, and let $\gamma_{x,y}$ be the unique path in the UST(G) from x to y. Then $\gamma_{x,y}$ has the same law as the loop-erased random walk from x to y. In particular, the loop-erased random walk from x to y has the same law as the loop-erased random walk from y to x.*

Proof. Start a random walk on G from x. The stopping time $\tau := \min\{k; X_k = y\}$ satisfies (2.4), whence also (2.5). By the Aldous–Broder algorithm, $\gamma_{x,y}$ has the same law as the unique path in $F((X_0, \ldots, X_\tau))$ from x to y. Thus $\gamma_{x,y}$ has the same law as the unique path in $L((X_0, \ldots, X_\tau))$ from x to y, which is just the loop-erased random walk from x to y.

The last sentence follows from the equality $\gamma_{x,y} = \gamma_{y,x}$. $\square$

Stopping times that rely on more randomness can also satisfy (2.5). One interesting such case is to consider a geometric random variable T independent of the random walk and let K be a finite subset of $V(G)$. Define τ to be the time of the Tth visit to K. Then τ is such a stopping time that satisfies (2.5).

Proposition 2.6. *If T is a random time independent of the random walk, and τ is a finite stopping time such that $\{\tau = n\} \in \sigma(T, C_e^n; e \in E(G))$ for all $n \ge 0$, then τ satisfies (2.5).*

Proof. For each path $\gamma = (v_0, \ldots, v_n)$ in G, we claim that

$$\mathbb{P}\big[(X_0, \ldots, X_n) = \gamma, \tau = n\big] = \mathbb{P}\big[(X_0, \ldots, X_n) = \Phi(\gamma), \tau = n\big]. \tag{2.6}$$

By Lemma 2.1 and the independence between T and the random walk, we have for every $k \ge 0$,

$$\mathbb{P}\big[(X_0, \ldots, X_n) = \gamma, T = k\big] = \mathbb{P}\big[(X_0, \ldots, X_n) = \Phi(\gamma), T = k\big]. \tag{2.7}$$

Let m_e denote the number of crossings of the edge e in γ. Since Φ preserves the number of times each edge is crossed,

$$\big\{(X_0, \ldots, X_n) = \gamma, T = k\big\} \cup \big\{(X_0, \ldots, X_n) = \Phi(\gamma), T = k\big\} \subset \big\{T = k, \forall e \in E \ C_e^n = m_e\big\}.$$

By the assumption on τ,

$$\{T = k, \forall e \in E \; C_e^n = m_e\} \cap \{\tau = n\} = \varnothing \quad \text{or} \quad \{T = k, \forall e \in E \; C_e^n = m_e\} \subset \{\tau = n\}.$$

In the former case,

$$\mathbb{P}\big[(X_0, \ldots, X_n) = \gamma, T = k, \tau = n\big] = \mathbb{P}\big[(X_0, \ldots, X_n) = \Phi(\gamma), T = k, \tau = n\big] = 0. \tag{2.8}$$

In the latter case,

$$
\begin{aligned}
\mathbb{P}\big[(X_0, \ldots, X_n) = \gamma, T = k, \tau = n\big] &= \mathbb{P}\big[(X_0, \ldots, X_n) = \gamma, T = k\big] \\
&\overset{(2.7)}{=} \mathbb{P}\big[(X_0, \ldots, X_n) = \Phi(\gamma), T = k\big] \\
&= \mathbb{P}\big[(X_0, \ldots, X_n) = \Phi(\gamma), T = k, \tau = n\big].
\end{aligned} \tag{2.9}
$$

So we always have

$$\mathbb{P}\big[(X_0, \ldots, X_n) = \gamma, T = k, \tau = n\big] = \mathbb{P}\big[(X_0, \ldots, X_n) = \Phi(\gamma), T = k, \tau = n\big].$$

Summing this equality over k yields (2.6).

Now it is easy to see that (2.5) holds for such stopping times τ: for each subtree t of G, one has that

$$
\begin{aligned}
\mathbb{P}\big[F\big((X_0, \ldots, X_\tau)\big) = t\big] &= \sum_{\gamma : F(\gamma) = t} \mathbb{P}\big[(X_0, \ldots, X_\tau) = \gamma, \tau = |\gamma|\big] \\
&\overset{(2.6)}{=} \sum_{\gamma : F(\gamma) = t} \mathbb{P}\big[(X_0, \ldots, X_\tau) = \Phi(\gamma), \tau = |\gamma|\big] \\
&\overset{\gamma' = \Phi(\gamma)}{=} \sum_{\gamma' : L(\gamma') = t} \mathbb{P}\big[(X_0, \ldots, X_\tau) = \gamma', \tau = |\gamma'|\big] \\
&= \mathbb{P}\big[L\big((X_0, \ldots, X_\tau)\big) = t\big]. \qquad \square
\end{aligned}
$$

2.2. *Augmented operators $\hat{L}$ and $\hat{F}$ and the reversing operator R*

Note that if $x = y$, then Lemma 2.1 is trivial: take Φ to reverse the path. Thus, we assume from now on that $x \neq y$.

We encode every path by a colored multi-digraph. We say a multi-digraph marked with a *start* vertex x and an *end* vertex y is *balanced* if the indegree of each vertex $u \notin \{x, y\}$ equals the outdegree of u, the outdegree of x is larger than its indegree by one, and the indegree of y is larger than its outdegree by one. The balance property is necessary for the existence of an Eulerian path in a multi-digraph. The coloring is either of the following two types.

By an *exit coloring* of a multi-digraph, we mean an assignment to each vertex v of a linear ordering on the set $\mathsf{Out}(v)$ of its outgoing edges. We call an edge *lighter* or *darker* if it is smaller or larger in the ordering. The maximal (darkest) edge in $\mathsf{Out}(v)$ is regarded as *black*, except for $v = y$. For the *entrance coloring*, the ordering is on the edges $\mathsf{In}(v)$ incoming to v instead, and there is no black edge leading into x.

Let $\mathcal{L}_n^{x,y}$ (resp., $\mathcal{F}_n^{x,y}$) denote the set of all colored multi-digraphs marked with the start x and end y satisfying the following five properties:

 (i) the vertex set of the multi-graph contains the start x and end y and is a subset of the vertex set of the original graph G;
 (ii) there are n directed edges in total, each one being an orientation of an edge from G;
(iii) the multi-digraph is balanced;
(iv) the coloring is an exit coloring (resp., an entrance coloring);
 (v) all black edges form a directed spanning tree of the multi-digraph with all edges going from leaves to the root y (resp., from the root x to leaves).

We define the augmented last-exit operator $\hat{L} : \mathcal{P}_n^{x,y} \to \mathcal{L}_n^{x,y}$ that, given a path, returns a colored multi-digraph in the most obvious way: the vertex set consists of all the visited vertices, and we draw one directed edge for each step of the path. This multi-digraph is clearly balanced. The exit coloring is naturally given by the order in which these outgoing edges are traversed. Also recall that there is no black edge leading out of y. Note that the tree formed by all black

edges, ignoring their orientation, is exactly the output of the last-exit operator L defined above, so $\hat{L}$ can be viewed as an augmentation of L. All conditions can be checked easily, so the map is well defined.

Similarly, we can define the augmented first-entrance operator $\hat{F}: \mathcal{P}_n^{x,y} \to \mathcal{F}_n^{x,y}$. The only difference is that we consider the entrance coloring on incoming edges instead of the exit coloring on outgoing edges, and we use the *reverse* of the order in which incoming edges are traversed, so that black edges, in particular, are first-entrance edges.

Clearly $\hat{L}$ and $\hat{F}$ are both injections.

Lemma 2.7. *The maps $\hat{L}$ and $\hat{F}$ are bijections.*

Proof. We describe the inverse operator for $\hat{L}$. Given a colored multi-digraph in $\mathcal{L}_n^{x,y}$, we can associate it with a path by starting from x and traversing the multi-digraph according to the following instruction:

always exit from the lightest unused edge until you run out of edges to use.

Notice that by the balance property, the algorithm always terminates at y. It remains to show that the path is Eulerian (i.e., uses all n edges exactly once), so that the operator is well defined. Given this, since the above rule respects the exit coloring, we may deduce that this traversal algorithm gives the inverse, $\hat{L}^{-1}$.

If the path is not Eulerian, i.e., if some edge has not been used, then some outgoing edge from some vertex u has not been used. According to our instructions, it follows that the black edge leaving u has not been used. Let that black edge be (u, v). By the balance condition, some outgoing edge from v has not been used, whence the black edge leaving v has not been used (unless $v = y$). By repeating this argument and using condition (v), we arrive at the conclusion that some edge leaving y has not been used, whence the path cannot have terminated, a contradiction.

The proof for $\hat{F}$ is similar; it also follows from (2.10) below and the fact that the other functions there are bijections. $\square$

The bijection $\hat{L}$ was also used in the proof of the BEST theorem; see Theorem 3.1 and its proof for details, where we use $\hat{F}$ instead.

Define a reversing operator $R: \mathcal{F}_n^{x,y} \to \mathcal{L}_n^{x,y}$ that maintains the vertex set but reverses all edges, except that the directions of the black edges on the unique path from x to y in the spanning tree remain unchanged. The coloring is also maintained: for $H \in \mathcal{F}_n^{x,y}$ and a vertex u, the set $\mathsf{In}(u)$ in H is mapped to $\mathsf{Out}(u)$ in $R(H)$, and so this set of edges can maintain its linear ordering, except that if $u \notin \{x, y\}$ is on the path from x to y, then one black edge is replaced by another, whereas x gains an outgoing black edge in $R(H)$ compared to the incoming edges to x in H, and y loses an outgoing black edge in $R(H)$ compared to the incoming edges to y in H. It is not hard to see that the operator R does have codomain $\mathcal{L}_n^{x,y}$ and is bijective.

2.3. Proof of Lemma 2.1

Proof of Lemma 2.1. We may now define the main construction, illustrated in Figure 1:

$$\Phi := \hat{L}^{-1} \circ R \circ \hat{F}. \tag{2.10}$$

Since R maintains the coloring, the black edges that formed the first-entrance spanning tree in $\mathcal{F}_n^{x,y}$ now form the last-exit version in $\mathcal{L}_n^{x,y}$ after reversal, that is, $F = L \circ \Phi$. Because $\hat{F}$, R, and $\hat{L}^{-1}$ are injections, so is Φ, which forces Φ to be
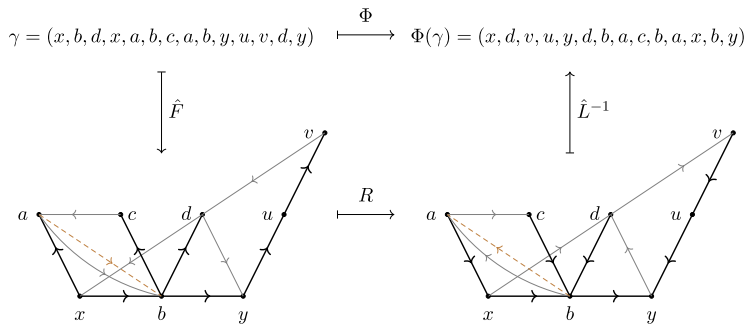

Fig. 1. An example of the bijection R, where we use the order that brown, dashed edges are lighter than gray edges and gray edges are lighter than black edges. In addition, black edges are drawn thick; these form the spanning tree. Note that R reverses the orientation of every edge except for the black edges that form a path from x to y.

a bijection. The map Φ preserves the random walk measure because all edges and vertices are visited same number of times: see (2.1). □

3. Concluding remarks

3.1. *The BEST theorem and the Aldous–Broder algorithm*

We will make explicit the relationship of our proof to the BEST theorem, and use the latter to give an alternative proof of the validity of the Aldous–Broder algorithm. In addition, because of their possible interest, we show how similar ideas lead to new proofs of the Markov chain tree theorem and the directed version of Wilson's algorithm, although these are not the shortest known proofs.

Suppose D is a multi-digraph marked with a start vertex x and an end vertex y. An *Eulerian path* in D is a directed path that starts at x, ends at y, and uses each edge exactly once. An *arborescence* in D is a directed spanning tree of D with each edge belonging to a directed path from the root x to a leaf. Each Eulerian path gives rise to the arborescence formed by all the first-entrance edges of vertices in D except for that of x.

The original BEST theorem [13, Theorem 5b] was concerned with *Eulerian circuits*, which are Eulerian paths that start and end at the same vertex. Depending on the context, they are viewed as loops either with or without a distinguished start or end. Here we state a version of the theorem that holds more generally for Eulerian paths. We denote the indegree of a vertex v by $\operatorname{indeg}(v)$.

Theorem 3.1 (BEST). *Suppose D is a balanced multi-digraph marked with a start x and an end y. For each arborescence Λ in D, there are exactly*

$$\operatorname{indeg}(x) \prod_{v \in V} \big(\operatorname{indeg}(v) - 1\big)! \tag{3.1}$$

Eulerian paths in D whose first-entrance edges coincide with the arborescence Λ. In particular, the total number of Eulerian paths in D is given by

$$t_x(D) \operatorname{indeg}(x) \prod_{v \in V} \big(\operatorname{indeg}(v) - 1\big)!,$$

where $t_x(D)$ is the number of arborescences in D directed away from the root x.

Proof. It suffices to show the formula in (3.1). The proof is a direct consequence of the bijection $\hat{F}$ (see Lemma 2.7): the number of Eulerian paths with first-entrance edges being Λ is equal to the number of entrance colorings of D whose black edges correspond to Λ, while the latter is exactly (3.1). □

Remark 3.2. It is perhaps more natural to reverse the Eulerian paths and formulate Theorem 3.1 in terms of last-exit edges. However, the first-entrance version will be useful for us to study the Aldous–Broder algorithm in Theorem 3.3.

Theorem 3.3 (Aldous–Broder). *Let G be a finite, connected network. Start a network random walk $(X_n)_{n=0}^{\infty}$ on G from the vertex x and stop at the cover time. Let T be the random spanning tree formed by all the first-entrance edges. Then T has the law $\mathsf{UST}(G)$.*

Proof. Recall that the Aldous–Broder algorithm run up to time n gives a tree on the vertices $\{X_k; k \leq n\}$. We denote this tree by T_n. It suffices to show that for every spanning tree t,

$$\lim_{n \to \infty} \mathbb{P}_x[T_n = t] = \prod_{e \in t} w(e) \Big/ \sum_{t'} \prod_{e \in t'} w(e), \tag{3.2}$$

where the summation is over all spanning trees of G.

For any pair of vertices (u, v), let

$$N_{u,v}(n) := \#\{j; 0 \leq j \leq n - 1, X_j = u, X_{j+1} = v\} \tag{3.3}$$

denote the transition counts from u to v up to time n. Let $N(n): (u, v) \mapsto N_{u,v}(n)$ be the collection of such transition counts. Note that $N(n)$ uniquely determines the end vertex X_n. Based on $N(n)$, we define a balanced multi-digraph D_n

with start $X_0 = x$ and end X_n that has the same vertex set as G and contains exactly $N_{u,v}(n)$ directed edges from u to v for all pairs (u, v). By definition, all random walk paths of length n with the same transition counts $N(n)$ occur with equal probability. Therefore, the random walk path up to time n, conditioned on the transition counts $N(n)$, has the same distribution as the law $\mathbb{Q}_{D_n}$ of a uniformly random Eulerian path in the multi-digraph D_n. Let $\overrightarrow{G}$ be the directed graph obtained from G by replacing each edge with two oppositely oriented edges. For any subdigraph H of D_n, define $\overline{H}$ to be the subdigraph of $\overrightarrow{G}$ formed from H by identifying each edge of D_n with its corresponding edge in $\overrightarrow{G}$. Thus we have

$$\mathbb{P}_x\big[T_n = t \mid N(n)\big] = \mathbb{Q}_{D_n}\big[\overline{\overline{\Lambda_n}} = \Lambda(t)\big], \tag{3.4}$$

where Λ_n represents the arborescence generated by the random Eulerian path, and $\Lambda(t)$ is the directed spanning tree in $\overrightarrow{G}$ if we view t as directed away from the root x.

By Theorem 3.1, each arborescence is equally likely to occur under $\mathbb{Q}_{D_n}$. Moreover, given a spanning tree t, there are exactly $\prod_{(u,v)\in\Lambda(t)} N_{u,v}(n)$ arborescences Λ in D_n with $\overline{\overline{\Lambda}} = \Lambda(t)$. Thus, the right-hand side of (3.4) is given by

$$\prod_{(u,v)\in\Lambda(t)} N_{u,v}(n) \Big/ \sum_{t'} \prod_{(u,v)\in\Lambda(t')} N_{u,v}(n). \tag{3.5}$$

Now we take expectation on both sides of (3.4) and let $n \to \infty$ in (3.5). This completes the proof of (3.2), since

$$\lim_{n\to\infty} \frac{N_{u,v}(n)}{n} = \lim_{n\to\infty} \frac{N_u(n)}{n} \cdot \frac{N_{u,v}(n)}{N_u(n)} = \pi(u)p(u,v) = \frac{w(u,v)}{\sum_{u\in V} w(u)} \quad \mathbb{P}_x\text{-a.s.,}$$

where $N_u(n) := \sum_{v\in V} N_{u,v}(n)$, and $\pi(u) := w(u)/\sum_{u\in V} w(u)$ represents the stationary distribution of the random walk. $\qquad\square$

A notable feature of our proof is that we never reversed time. Thus, essentially the same proof yields the following, where $\overleftarrow{p}$ denotes the probability transitions of the reversed Markov chain and $-e$ denotes the reversal of e:

Theorem 3.4 (Directed Aldous–Broder). *Let X be an irreducible Markov chain on a finite state space. Let G be the corresponding directed graph. Let $X_0 = x$ and stop X at the cover time. Let T be the arborescence formed by all the first-entrance edges. Then there is a constant c such that for every arborescence t of G rooted at x, $\mathbb{P}[T = t] = c \prod_{e\in t} \overleftarrow{p}(-e)$.*

Proof. It remains to note that $\prod_{(u,v)\in t} \pi(u)p(u,v) = \prod_{(u,v)\in t} \pi(v)\overleftarrow{p}(v,u) = \prod_{v\neq x} \pi(v) \cdot \prod_{e\in t} \overleftarrow{p}(-e)$. $\qquad\square$

It turns out that one can also deduce the Markov chain tree theorem in a similar fashion. To this end, we introduce a random walk measure μ_n on the loop space. Let X be an irreducible Markov chain on a finite state space with stationary probability measure π. Let G be the corresponding directed graph. Recall that $\mathcal{P}_n^{x,y}$ denotes the set of all paths in G from x to y with length n. Let

$$\mathcal{L}_n := \bigcup_{x\in V(G)} \mathcal{P}_n^{x,x}$$

be the space of all loops of length n in G. If r is the period of the Markov chain X, then $\mathcal{L}_n$ is nonempty iff $n \in r\mathbb{N}$. For simplicity in what follows, we will assume that X is an aperiodic chain, i.e., $r = 1$, although the proof is similar for $r > 1$. Assuming $r = 1$, we define a probability measure μ_n on $\mathcal{L}_n$ by assigning to each path $\gamma = (v_0, \ldots, v_n = v_0) \in \mathcal{L}_n$ the probability

$$\mu_n(\gamma) := \frac{1}{Z_n}p(\gamma) = \frac{1}{Z_n}\prod_{k=0}^{n-1} p(v_k, v_{k+1}),$$

where Z_n is the normalizing constant. As n goes to infinity, the ergodic theorem implies that $Z_n\mu_n(\mathcal{P}_n^{x,x}) \to \pi(x)$. Thus, we have $Z_n \to 1$ and $\mu_n(\mathcal{P}_n^{x,x}) \to \pi(x)$.

Corollary 3.5 (Markov Chain Tree Theorem). *Let X be an irreducible Markov chain on a finite state space with stationary probability measure π. Let G be the corresponding directed graph. Then there is a constant c such that for every x, $\pi(x) = c \sum_{\text{root}(t)=x} \prod_{e\in t} p(e)$, where the sum is over arborescences t of G directed* toward *the root x.*

Proof. The proof here follows closely that of Theorem 3.3. The main difference, however, is that we sample the path $(X_0, \ldots, X_n)$ not from the network random walk, but from the random walk measure μ_n on the loop space $\mathcal{L}_n$. Notice that the definitions of $N(n)$ and D_n below (3.3) still make sense as random variables on the loop space $\mathcal{L}_n$. In this case, the start of D_n coincides with its end, and the indegree of each vertex in D_n is equal to its outdegree. Let D_n^{o} be the same balanced multi-digraph as D_n, except that it does *not* distinguish any start or end. All Eulerian paths in D_n^{o} are Eulerian circuits, which we consider as loops *with* starts: different Eulerian paths may have different starts in $V(D_n^{\mathrm{o}})$.

Now note that the law of random loops μ_n, conditioned on the transition counts $N(n)$, has the same distribution as the law $\mathbb{Q}_{D_n^{\mathrm{o}}}$ of a uniformly random Eulerian path in D_n^{o}. Also note that the probability under $\mathbb{Q}_{D_n^{\mathrm{o}}}$ for a certain arborescence of D_n^{o} to occur is no longer uniform, but proportional to the indegree of its root, according to (3.1). Therefore, for any arborescence t of G directed *away* from its root, we have

$$\mu_n\big[T_n = t \mid N(n)\big] = N_{\mathrm{root}(t)}(n) \prod_{(u,v)\in t} N_{u,v}(n) \Big/ \sum_{t'} N_{\mathrm{root}(t')}(n) \prod_{(u,v)\in t'} N_{u,v}(n),$$

where T_n is the arborescence formed by all the first-entrance edges. By a similar calculation as in the proof of Theorem 3.3 and 3.4, we conclude that there exists a constant c such that

$$\lim_{n\to\infty} \mu_n[T_n = t] = c \prod_{e\in t} \overleftarrow{p}(-e).$$

Thus

$$\pi(x) = \lim_{n\to\infty} \mu_n\big(\mathcal{P}_n^{x,x}\big) = \lim_{n\to\infty} \sum_{\mathrm{root}(t)=x} \mu_n[T_n = t] = c \sum_{\mathrm{root}(t)=x} \prod_{e\in t} \overleftarrow{p}(-e).$$

We complete the proof by reversing the process. $\square$

Another consequence of Theorem 3.4 is the directed Wilson's algorithm. Due to the spatial Markov property of UST for digraphs, it suffices to show the following analogue of Corollary 2.5.

Corollary 3.6. *Let X be an irreducible Markov chain on a finite state space. Let G be the corresponding directed graph. Let x be a vertex in the digraph G and $\mathcal{T}_x$ be the collection of spanning trees of G such that all edges on the trees are directed toward x. For $t \in \mathcal{T}_x$, write $\Psi(t) = \prod_{e\in t} p(e)$. Let $\mathsf{UST} = \mathsf{UST}(G, x)$ be the uniform spanning tree measure, i.e., the probability measure on $\mathcal{T}_x$ such that $\mathsf{UST}(\{t\}) \propto \Psi(t)$. For a vertex $y \neq x$, and let $\gamma_{y,x}$ be the unique directed path in the UST from y to x. Then $\gamma_{y,x}$ has the same law as the loop-erased random walk from y to x.*

Proof. Let $\overleftarrow{\mathbb{P}}_x$ denote the law of the reversed chain started at x. Let $r = (r_0, r_1, \ldots, r_n)$ be a directed path in G and let $-r = (r_n, r_{n-1}, \ldots, r_0)$ be the reversed path in the reversed graph. Similar to (2.1), we let $\overleftarrow{p}(-r) = \prod_{j=1}^n \overleftarrow{p}(r_j, r_{j-1})$.

For a directed self-avoiding path w from y to x in G, let $\mathcal{R}_{y,x}(w)$ be the set of directed paths $r = (r_0, r_1, \ldots, r_n)$ on G from y to x such that

- $r_0 = y$, $r_n = x$, and $\forall i \geq 1$ $r_i \neq y$,
- the loop-erasure of the path r is w.

Thus, $-\mathcal{R}_{y,x}(w) := \{-r; r \in \mathcal{R}_{y,x}(w)\}$ is just the set of all possible walks in the reversed chain from x to the first hit of y such that the path from x to y in the tree formed by first-entrance edges of the walk is $-w$.

By Theorem 3.4, for a directed self-avoiding path $w = (w_0, w_1, \ldots, w_k)$ from y to x in G,

$$\mathbb{P}[\gamma_{y,x} = w] = \sum_{r\in\mathcal{R}_{y,x}(w)} \overleftarrow{p}(-r) = \sum_{r\in\mathcal{R}_{y,x}(w)} \frac{\pi(y)}{\pi(x)} p(r)$$

$$= \frac{\pi(y)}{\pi(x)} \prod_{j=1}^k p(w_{j-1}, w_j) \prod_{j=1}^k \mathscr{G}_{A_{j-1}}(w_j, w_j), \tag{3.6}$$

where $A_j := \{w_0, \ldots, w_j\}$ for $j = 0, \ldots, k$ and $\mathscr{G}_Z(a, a)$ is the expected number of visits to a strictly before hitting Z by a random walk on G started from a. The last step of (3.6) can be seen by decomposing the path in a unique way as the concatenation of edges (w_{j-1}, w_j) and cycles at $w_1, \ldots, w_n$, with the cycle at w_j avoiding A_{j-1}.

Let $l_{y \to x}$ denote the loop-erased random walk from y to x. Using the same technique of decomposing the path (or using formula (12.2.2) of [7]), one has that

$$\mathbb{P}[l_{y \to x} = w] = \prod_{j=1}^{k} p(w_{j-1}, w_j) \prod_{j=0}^{k-1} \mathscr{G}_{A_{j-1} \cup \{x\}}(w_j, w_j), \tag{3.7}$$

where $A_{-1} := \varnothing$.

Comparing (3.6) with (3.7), we see that it suffices to show the following identity: for any simple directed path $w = (w_0, \dots, w_k)$ from y to x,

$$\frac{\pi(y)}{\pi(x)} \prod_{j=1}^{k} \mathscr{G}_{A_{j-1}}(w_j, w_j) = \prod_{j=0}^{k-1} \mathscr{G}_{A_{j-1} \cup \{x\}}(w_j, w_j). \tag{3.8}$$

By Cramer's rule, $\mathscr{G}_Z(a, a) = \frac{\det(I-P)[Z \cup \{a\}]}{\det(I-P)[Z]}$, where P is the transition matrix and for a matrix A indexed by $V(G)$ and $Z \subset V(G)$, the matrix $A[Z]$ is the matrix obtained from A by deleting all rows and columns indexed by an element of Z. Using this, we may simplify (3.8) to

$$\frac{\pi(y)}{\det(I - P)[y]} = \frac{\pi(x)}{\det(I - P)[x]}. \tag{3.9}$$

Since (3.9) is easy to show by applying Cramer's rule to the system of equations determining the stationary probability measure π, we are done. □

Remark 3.7. In [7], Lawler proved another very interesting identity and used it to prove Wilson's algorithm:

$$\forall a, b \notin Z \quad \mathscr{G}_Z(b, b)\mathscr{G}_{Z \cup \{b\}}(a, a) = \mathscr{G}_Z(a, a)\mathscr{G}_{Z \cup \{a\}}(b, b).$$

From this identity one can easily deduce that the function in (12.2.3) on page 200 of [7] is a symmetric function. The directed Wilson's algorithm then follows from this symmetry easily; see (12.7.1) on page 212 of [7].

3.2. *Infinite networks*

Suppose G is a locally finite, infinite connected graph. An *exhaustion* of G is a sequence of finite connected subgraphs $G_n = (V_n, E_n)$ of G such that $G_n \subset G_{n+1}$ and $G = \bigcup G_n$. Suppose that V_n induces G_n, i.e., G_n is the maximal subgraph of G with vertex set V_n. Let G_n^* be the graph formed from G by contracting all vertices outside V_n to a new vertex, ∂_n. Let T_n be a sample of $\mathsf{UST}(G_n^*)$. Then the *wired uniform spanning forest* is the weak limit of T_n. If we orient T_n toward ∂_n and then take the weak limit, we get the *oriented wired uniform spanning forest* of G. For details, see [2] or [8, Chapter 10].

Wilson's algorithm [14] is another efficient way of sampling a uniform spanning tree for a finite connected graph using loop erasure of random walks. It can be applied to recurrent networks directly. For transient networks, Wilson's algorithm can also be applied with a simple modification. This is called Wilson's method rooted at infinity; see [2] for details.

The Aldous–Broder algorithm can also be used to sample the wired uniform spanning forest on recurrent networks directly. However, the extension of the Aldous–Broder algorithm for sampling the wired uniform spanning forest on transient networks was found much later by Hutchcroft [5] using the interlacement process.

Here we simply state a similar interlacement using last-exit edges. It is not directly related to our reverse Aldous–Broder algorithm. Instead, it relates to the process created from the stationary random walk on the nonpositive integers that we recalled in our introduction. The interested reader can refer to [5] for details on the interlacement process and the interlacement Aldous–Broder algorithm.

Theorem 3.8. *Let G be a transient, connected, locally finite network, let $\mathscr{I}$ be the interlacement process on G, and let $t \in \mathbb{R}$. For each vertex v of G, let $\lambda_t(v)$ be the largest time before t such that there exists a trajectory $(W_{\lambda_t(v)}, \lambda_t(v)) \in \mathscr{I}$ passing through v, and let $e_t(v)$ be the oriented edge of G that is traversed by the trajectory $W_{\lambda_t(v)}$ as it leaves v for the last time before t. Then $\{e_t(v); v \in V\}$ has the law of the oriented wired uniform spanning forest of G.*

The proof of Theorem 3.8 is simply an analogue to that of [5, Theorem 1.1]. This version can be used in place of that used by Hutchcroft and is perhaps more natural.

References

[1] D. J. Aldous. The random walk construction of uniform spanning trees and uniform labelled trees. *SIAM J. Discrete Math.* **3** (4) (1990) 450–465. MR1069105 https://doi.org/10.1137/0403039

[2] I. Benjamini, R. Lyons, Y. Peres and O. Schramm. Uniform spanning forests. *Ann. Probab.* **29** (1) (2001) 1–65. MR1825141 https://doi.org/10.1214/aop/1008956321

[3] P. Billingsley. Statistical methods in Markov chains. *Ann. Math. Stat.* **32** (1961) 12–40. MR0123420 https://doi.org/10.1214/aoms/1177705136

[4] A. Broder. Generating random spanning trees. In *30th Annual Symposium on Foundations of Computer Science (Research Triangle Park, North Carolina, 30 Oct.–1 Nov. 1989)* 442–447. IEEE, New York, 1989.

[5] T. Hutchcroft. Interlacements and the wired uniform spanning forest. *Ann. Probab.* **46** (2) (2018) 1170–1200. MR3773383 https://doi.org/10.1214/17-AOP1203

[6] G. F. Lawler. A connective constant for loop-erased self-avoiding random walk. *J. Appl. Probab.* **20** (2) (1983) 264–276. MR0698530 https://doi.org/10.1017/s0021900200023421

[7] G. F. Lawler. Loop-erased random walk. In *Perplexing Problems in Probability* 197–217. *Progr. Probab.* **44**. Birkhäuser Boston, Boston, MA, 1999. MR1703133

[8] R. Lyons and Y. Peres. *Probability on Trees and Networks*. *Cambridge Series in Statistical and Probabilistic Mathematics* **42**. Cambridge University Press, New York, 2016. Available at http://rdlyons.pages.iu.edu. MR3616205 https://doi.org/10.1017/9781316672815

[9] C. Moore and S. Mertens. *The Nature of Computation*. Oxford University Press, Oxford, 2011. MR2849868 https://doi.org/10.1093/acprof:oso/9780199233212.001.0001

[10] R. Pemantle. Choosing a spanning tree for the integer lattice uniformly. *Ann. Probab.* **19** (4) (1991) 1559–1574. MR1127715

[11] J. G. Propp and D. B. Wilson. How to get a perfectly random sample from a generic Markov chain and generate a random spanning tree of a directed graph. *J. Algorithms* **27** (2) (1998) 170–217. MR1622393 https://doi.org/10.1006/jagm.1997.0917

[12] C. A. B. Smith and W. T. Tutte. On unicursal paths in a network of degree 4. *Amer. Math. Monthly* **48** (4) (1941) 233–237. MR1525117 https://doi.org/10.2307/2302716

[13] T. van Aardenne-Ehrenfest and N. G. de Bruijn. Circuits and trees in oriented linear graphs. *Simon Stevin* **28** (1951) 203–217. MR0047311

[14] D. B. Wilson. Generating random spanning trees more quickly than the cover time. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing (Philadelphia, PA, 1996)* 296–303. ACM, New York, 1996. MR1427525 https://doi.org/10.1145/237814.237880

[15] A. Zaman. Urn models for Markov exchangeability. *Ann. Probab.* **12** (1) (1984) 223–229. MR0723741