Generating random spanning trees

Andrei Broder

DEC - Systems Research Center
130 Lytton Ave., Palo Alto, CA 94301

Extended abstract

Abstract.

This paper describes a probabilistic algorithm that, given a connected, undirected graph G with n vertices, produces a spanning tree of G chosen uniformly at random among the spanning trees of G. The expected running time is $O(n \log n)$ per generated tree for almost all graphs, and $O(n^3)$ for the worst graphs. Previously known deterministic algorithms and much more complicated and require $O(n^3)$ time per generated tree.
A Markov chain is called *rapidly mixing* if it gets close to the limit distribution in time polynomial in the log of the number of states. Starting from the analysis of the algorithm above we show that the Markov chain on the space of all spanning trees of a given a graph where the basic step is an edge swap is rapidly mixing.

1. Introduction.

> Puzzle: A particle moves on a cycle graph. At each step the particle goes from the current vertex to one of the two adjacent vertices chosen with equal probability. Clearly, the first time when the particle has visited all the vertices, it has gone over all the edges in the cycle, except one. The question is, how is the "left-out" edge distributed, relative to the starting point? It is tempting to conjecture that the edge farthest away from the origin is the most likely to be left out. The answer is given at the end of section 3.

Consider a particle that moves on a connected, undirected graph $G = (V, E)$ with n vertices. At each step the particle goes from the current vertex to one of its neighbors, chosen uniformly at random. This stochastic process is a Markov chain; it is called the simple random walk on the graph G. (See [KS69] for a general reference on Markov chains.)

The first result of this paper is that the simulation of the simple random walk on a connected undirected graph G can be used to generate a spanning tree of G uniformly at random (over all the spanning trees of G) by a very simple algorithm:

Algorithm **Generate**.

1. Simulate the simple random walk on the graph G starting at an arbitrary vertex s until every vertex is visited. For each vertex $i \in V - s$ collect the edge $\{j, i\}$ that corresponds to the first entrance to vertex i. Let T be this collection of edges.

2. Output the set T.

∎

The set T is a spanning tree because it contains $|V| - 1$ edges (one for each vertex except s) and no loops. We shall see later that it is indeed uniformly distributed.

The *cover time*, C_v is the first time when the particle has visited all the vertices in the graph starting from a vertex v. Clearly the expected running time of **Generate** per output tree is equal to $\mathbf{E}(C_s)$. It is known [AKLLR79] that for every connected graph $\mathbf{E}(C) = O(n^3)$; however it was recently shown in [BK89], that if the transition probability matrix of the simple random walk has a second largest eigenvalue bounded away from 1, then the expected cover time is only $O(n \log n)$. This condition is satisfied by almost all graphs (in the $G_{n,p}$ model, for every $p > c \log n/n$, in particular $p = 1/2$ [FK81]), and by almost all d-regular graphs (see [BS87], [FKS89]). Also all expander graphs have $\mathbf{E}(C) = O(n \log n)$ [CRRST89]. Hence the expected running time of the **Generate** algorithm is $O(n \log n)$ per generated tree for almost all graphs, and $O(n^3)$ for the worst graphs.

The first previously published algorithm for this problem [Guénoche83] has a running time of roughly n^5. It is based on the fact that the total number of (directed) trees in a graph can be computed exactly (the Matrix-Tree Theorem – see e.g. [Knuth73 – p. 377]) by computing a determinant of size $n \times n$. The algorithm views the edges of the graph as labeled with distinct labels from 1 to m. Each spanning tree is labeled by the set of its edges. This induces a lexicographic order on the set of trees, and the i'th tree can be found by computing at most m determinants. Subsequent improvements [CDN88, CDM89] reduced the number of determinant computations thus decreasing the running time to $O(n^3)$ or $O(L(n))$ where $L(n)$ is the time to multiply $n \times n$ matrices, but the newer algorithms are quite complicated.

The algorithm **Generate** is based on the simulation of a Markov chain on the space of the objects of interest, a technique that had recently seen several very interesting applications to the quasi-uniform generation of combinatorial structures, and via such generation to approximate counting in polynomial time: [Broder86], [JS87], [DLMV88] describe the quasi-uniform generation of matchings in certain classes of graphs and the approximation of the permanent, and [DFK88] describes the quasi-uniform generation of sample points that are useful for approximation of the volume of a given convex body.

In contrast **Generate** applied to an undirected graph produces spanning trees with an exactly uniform distribution. For directed graphs, an algorithm similar to **Generate** can be used to produce quasi-uniform directed trees in time $O(n\,\mathbf{E}(C))$ where $\mathbf{E}(C)$ is the expected time to cover the graph. (See section 4.) This reduces to only $O(\mathbf{E}(C))$ if the graph is out-degree regular. The basic idea is to simulate a certain Markov chain $\{B_t\}$ (called "the backward chain") on the spanning trees of the directed graph and stop after a fixed number of steps.

A more "natural" random walk on the spanning trees of an undirected graph $G(V, E)$ is the following: Let S_t be the current tree; pick an edge $e \in E$ and edge $f \in S_t$ uniformly at random. If $S_t + \{f\} - \{e\}$ is a tree then let $S_{t+1} = S_t + \{f\} - \{e\}$; otherwise $S_{t+1} = S_t$. In other words, swap e and f if possible. The chain $\{S_t\}$ being symmetric, converges to the uniform distribution.

In section 5 we show that the chain $\{S_t\}$ gets close to the uniform distribution in time polynomial in the number of vertices in G. We use the fact that the transition graph associated to $\{S_t\}$ is a supergraph of a refinement of the transition graph associated to $\{B_t\}$.

2. The Markov Chain Tree Theorem.

Let M be a Markov chain with a finite state space $V = \{1, \ldots, n\}$ and transition probability matrix P. With this chain we associate a directed graph $G_M = (V, E)$, where the edge set E is given by $E = \{[i, j] \mid P_{i,j} > 0\}$. For each edge $[i, j] \in E$, its *weight*, $w\big([i, j]\big)$ is $P_{i,j}$. A *directed spanning tree of* G_M *rooted at* i is spanning tree of G_M with all edges pointing towards the root i. We define the weight of a spanning tree T by $w(T) = \prod_{e \in T} w(e)$. The family of all the directed spanning trees of G_M rooted at i is denoted $\mathcal{T}_i(G_M)$ and the family of all rooted directed spanning trees of G_M is denoted $\mathcal{T}(G_M)$.

We shall restrict our discussion to irreducible chains, in which case the graph G_M is strongly connected. All the results can be easily extended to reducible chains.

The Markov Chain Tree Theorem states that the stationary distribution, π_i, of an irreducible chain is proportional to the sum of the weights of all the directed spanning tree rooted at i. This theorem goes apparently back to Kirchhoff [Diaconis88], but it was re-discovered many times ([KV80], [LR82]).

We will not make direct use of this theorem, but the concepts introduced in the proof below are essential to the proof of our main result.

Theorem 1. (The Markov chain tree theorem.) *Let M be an irreducible Markov chain on n states with stationary distribution $\pi_1, \ldots, \pi_n$. Let G_M be the directed graph associated with M. Then*

$$\pi_i = \frac{\sum_{T \in \mathcal{T}_i(G_M)} w(T)}{\sum_{T \in \mathcal{T}(G_M)} w(T)}.$$

Proof: This proof is due to Persi Diaconis. (A similar proof was published in [AT88].)

Let $M = X_0, X_1, \ldots$. It is convenient to view M as the position of a particle moving on G_M with suitable transition probabilities.

Define B_t, the *backward tree at time t*, as follows: Let I be the set of states visited before time $t+1$, that is $I = \bigcup_{0 \le i \le t} \{X_i\}$. For each $i \in I$ let $l(i,t)$ be the last time the state i was visited before time $t+1$. The root of the backward tree B_t is X_t and the edges of B_t are $\left\{[X_{l(i,t)}, X_{l(i,t)+1}] \mid i \in I - X_t\right\}$. In other words, B_t is formed by superposing the (directed) edges corresponding to the last exit from each visited state before time t. Clearly B_t is a rooted directed tree with each edge going from leaves to root. For instance if the sequence of visited states is

t	0 1 2 3 4 5 6 7 8 9 ...
X_t	2 7 1 8 2 8 1 8 2 8 ...

then $B_5 = \{[7,1],[1,8],[2,8]\}$, $B_6 = \{[7,1],[2,8],[8,1]\}$ and so on.

Let C be the *cover time*, that is the first time when all the states are visited. Plainly, for $t \ge C$, the tree B_t is a rooted directed spanning tree of G_M.

Remark that through the definition above, the random walk $\{X_t\}$ on the vertices of G_M induces a Markov chain $\{B_t\}$ on the space of all the directed trees of G_M, called the backward tree chain. In the backward tree chain, non-spanning directed trees of G_M

are transient states because all the states will be eventually visited, the base chain being irreducible. Assume, as we will shall show later, that all the rooted directed spanning trees of G_M form exactly one recurrent class in the backward tree chain. Hence the backward tree chain has a stationary distribution $\sigma(T)$ with $\sigma(T) > 0$ iff T is a rooted directed spanning tree.

Plainly, the stationary distributions of the two chains satisfy

$$\pi_i = \lim_{N\to\infty} \frac{1}{N} \sum_{0\le t\le N} \mathbf{Pr}(X_t = i)$$
$$= \lim_{N\to\infty} \frac{1}{N} \sum_{0\le t\le N} \mathbf{Pr}(B_t \text{ rooted at } i) = \sum_{T\in\mathcal{T}_i} \sigma(T).$$

Hence all we need to show is that $\sigma(T)$ is proportional to $w(T)$, or equivalently that $w(T)$ satisfies the stationarity equations for the backward tree chain.

Let $T^{(i)}$ be a fixed directed spanning tree rooted at i. The only possible directed spanning tree *precursors* of $T^{(i)}$ (that is, spanning trees from which $T^{(i)}$ can be reached in one step) are trees corresponding to adding an edge $[i,j]$ to $T^{(i)}$ and deleting the last edge in the path from j to i in $T^{(i)}$. Call the deleted edge $[l(j),i]$ and note that the probability of the transition to $T^{(i)}$ from this predecessor is $P_{l(j),i}$. Therefore the stationarity equations are

$$\sigma\bigl(T^{(i)}\bigr) = \sum_{[i,j]\in E} \sigma\bigl(T^{(i)} + [i,j] - [l(j),i]\bigr) P_{l(j),i}.$$

These equations are satisfied by $w(T)$ because

$$\sum_{[i,j]\in E} w\bigl(T^{(i)} + [i,j] - [l(j),i]\bigr) P_{l(j),i}$$
$$= \sum_{[i,j]\in E} \frac{w\bigl(T^{(i)}\bigr) P_{i,j}}{P_{l(j),i}} P_{l(j),i} = w\bigl(T^{(i)}\bigr) \sum_j P_{i,j} = w\bigl(T^{(i)}\bigr),$$

as required.

It remains to show that all the rooted directed spanning trees of G_M form one recurrent class in the backward tree chain. This part of the proof is left for the full paper. ∎

3. Reversible chains and uniform generation.

If $M = \{X_t\}$ is a reversible chains it is possible to associate to M a second kind of directed tree in G_M.

Define F_t, the *forward tree at time* t, as follows: Let I be the set of states visited before time $t+1$, that is $I = \bigcup_{0 \le i \le t} \{X_i\}$. For each $i \in I$ let $f(i,t)$ be the first time the state i was visited. The root of the forward tree F_t is X_0 and the edges of F_t are $\left\{[X_{f(i,t)}, X_{f(i,t)-1}] \mid i \in I - X_0\right\}$. In other words, F_t is formed by superposing the edges corresponding to the first entrance to each visited state, with a reverse orientation. Because M is a reversible chains, these reversed edges are guaranteed to exist. Clearly F_t is a directed tree with each edge going from leaves to root. For instance if the sequence of visited states is

t	0 1 2 3 4 5 6 7 8 9 ...
X_t	2 7 1 8 2 8 1 8 2 8 ...

then $F_2 = \{[7,2],[1,7]\}$, $F_3, \ldots, F_9 = \{[7,2],[1,7],[8,1]\}$, and so on.

Note that for $t \ge C$, where C is the cover time, F_t is a directed spanning tree and $F_t = F_C$. For the chain $\{F_t\}$ the non-spanning trees are transient and each spanning tree is an absorbing state. Furthermore, the next theorem shows that F_C is distributed according to the stationary distribution of the backward tree chain.

Theorem 2. *Let M be an irreducible, reversible Markov chain on n states with stationary distribution $\pi_1, \ldots, \pi_n$. Let G_M be the directed graph associated with M. Let C be the cover time for M starting from the stationary distribution. Let F_C be the forward tree at time C. Then for any rooted directed spanning tree, T, of G_M*

$$\mathbf{Pr}(F_C = T) = \frac{\prod_{[i,j]\in T} P_{i,j}}{\sum_{T' \in \mathcal{T}(G_M)} \prod_{[i,j]\in T'} P_{i,j}}.$$

Proof: Fix some integer $k > 0$. Because M is reversible and we start from the stationary distribution,

$$\mathbf{Pr}(X_0 = x_0, X_1 = x_1, \ldots, X_k = x_k)$$
$$= \mathbf{Pr}(X_0 = x_k, X_1 = x_{k-1}, \ldots, X_k = x_0).$$

Hence, for reversible chains, starting from the stationary distribution, for any tree T,

$$\mathbf{Pr}(B_k = T \mid \pi) = \mathbf{Pr}(F_k = T \mid \pi),$$

where the condition "$\mid \pi$" is a shorthand for "$\mid X_0$ distributed according to π."

Let now T be a rooted directed spanning tree. The stationary distribution for the backward tree chain, $\sigma(T)$, satisfies

$$\sigma(T) = \lim_{N\to\infty} \frac{1}{N} \sum_{0\le t\le N} \mathbf{Pr}(B_t = T \mid \pi)$$

$$= \lim_{N\to\infty} \frac{1}{N} \sum_{0\le t\le N} \mathbf{Pr}(F_t = T \mid \pi).$$

But $\lim_{N\to\infty} \frac{1}{N} \sum_{0\le t\le N} \mathbf{Pr}(F_t = T \mid \pi)$ is exactly the probability that the forward chain is absorbed into state T, which is the same as $\mathbf{Pr}(F_C = T \mid \pi)$. ∎

Let M be the simple random walk on a connected, undirected graph $G = (V, E)$. This process is a reversible Markov chain with stationary distribution $\pi_i = d_i / \sum_{j\in V} d_j$ where d_i is the degree of i. The directed graph associated with this chain, $G_M = (V, E')$, is obtained by replacing each undirected edge $\{i, j\} \in E$ by a directed edge $[i, j]$ with weight $1/d_i$, and a directed edge $[j, i]$ with weight $1/d_j$.

Corollary 3. *Let M be the simple random walk on a connected, undirected graph $G = (V, E)$ starting at some vertex i. Let G_M be the directed graph associated with M. Let C be the cover time for G starting from the stationary distribution. Let F_C be the forward tree (in G_M) at time C. Then F_C is uniformly distributed over all directed spanning trees rooted at i.*

Proof: We apply the previous theorem. Notice that the weight of any directed spanning tree T rooted at i is $d_i / \prod_{j\in V} d_j$ because each node $j \neq i$ has outdegree 1 in T and all edges out of vertex j have weight $1/d_j$. Hence all directed spanning trees rooted at i are equally likely. ∎

Corollary 4. (Proof of the algorithm **Generate**) *Let M be the simple random walk on a connected, undirected graph $G = (V, E)$ starting at some vertex i. Let G_M be the directed graph associated with M. Let C be the cover time for G starting from the stationary distribution. Let F_C be the forward tree (in G_M) at time C. Then the undirected version of F_C is a spanning tree of G distributed uniformly at random among the spanning trees of G.*

Proof: Immediate from the previous corollary. (The undirected version of a directed graph is obtained by replacing each directed edge $[i, j]$ by an undirected edge $\{i, j\}$ and removing duplicate edges.) ∎

Returning to the puzzle mentioned in the introduction, the corollary above implies that the tree induced by the simple random walk until cover time is uniformly distributed over the spanning trees of the cycle; but the "left-out" edge is exactly the complement of the induced tree, and therefore it also uniformly distributed.

The brute force approach to proving this fact (path counting) is fairly tedious. However there is a very simple direct proof: consider a fixed edge $\{a, b\}$ on the cycle. Eventually (with probability 1) the particle will visit for the first time an endpoint of $\{a, b\}$, say a. Given this fact, the probability that $\{a, b\}$ is the "left-out" edge, is the probability that the particle will visit b starting from a before it traverses the edge $\{a, b\}$. This is clearly independent of the position of $\{a, b\}$ and therefore all edges are equally likely to be the "left-out" edge. This simple proof was re-discovered for my benefit by Jim Saxe. Previous discoverers include Avrim Blum and Ernesto Ramos.

4. Quasi-uniform generation

In general the algorithm **Generate** can not be applied to directed graphs, because in this case the simple random walk is not reversible. However a very simple coupling argument shows that the backward tree chain get close to the stationary distribution in time $O(\mathbf{E}(C))$. This idea yields an algorithm for quasi-uniform generation in time $O(n\,\mathbf{E}(C))$. The extra factor appears because the random walk needs to be modified in such a way that all transitions have the same probability; thus every tree has the same weight. This is done by adding self-loops to the graph. If the graph is already out-degree regular then the running time is only $O(\mathbf{E}(C))$. On the other hand for directed graphs $\mathbf{E}(C)$ can be as large as 2^n which is not competitive with the deterministic algorithm. The details are left for the full paper.

5. The swap chain

The most "natural" random walk on the spanning trees of an undirected graph G is the following: Let S'_t be the current tree; pick an edge e from $G - S'_t$ uniformly at random. Add e to T thus forming a cycle; pick an edge uniformly at random from the cycle and delete it to obtain S'_{t+1}. The chain $\{S'_t\}$ being symmetric, converges to the uniform distribution. Despite many attempts so far there was no proof that $\{S'_t\}$ gets close to the uniform distribution in time polynomial in n. In this section we prove this fact. (In general, the number of spanning trees is exponential in n; therefore we want to show that $\{S_t\}$ gets close to its stationary distribution in time polynomial in the log of the number of states. Such chains are called rapidly mixing.)

Consider this chain: Let S_t be the current tree; pick an edge $e \in E$ and edge $f \in S_t$ uniformly at random. If $S_t + \{f\} - \{e\}$ is a tree then let $S_{t+1} = S_t + \{f\} - \{e\}$; otherwise $S_{t+1} = S_t$. In other words, swap e and f if possible. The chain $\{S_t\}$ being symmetric, converges to the uniform distribution. Moreover, all transition in $\{S_t\}$ have the same probability, $1/\big((n-1)m\big)$. So $\{S_t\}$ is just a simple random walk on the undirected graph associated to it. We call $\{S_t\}$ the swap chain. Note that $\{S'_t\}$ is just the jump chain of $\{S_t\}$.

To show that $\{S_t\}$ is rapidly mixing, we use the fact that the transition graph associated to $\{S_t\}$ is a supergraph of a refinement of the transition graph associated to $\{B_t\}$.

Let M be an arbitrary irreducible Markov chain with state space V stationary distribution π, and transition probability matrix P. Following [SJ89] define for every non-empty subset $S \subset V$ the *ergodic flow* out of S, as $F_S(P) = \sum_{i \in S, j \notin S} \pi_i P_{ij}$.

Lemma 5. *Let P be the transition probability matrix of an irreducible Markov chain with state space V and stationary distribution π. Assume that P is doubly-stochastic. Then for every non-empty $S \subset V$*

$$F_S(P^t) \leq t F_S(P).$$

Proof: Left for the full paper. ∎

Lemma 6. *Let P be the transition probability matrix of an irreducible Markov chain with state space V and stationary distribution π. Assume that P is doubly-stochastic. Then for every non-empty $S \subset V$*

$$F_S\big(\frac{1}{2}(P + P^T)\big) = F_S(P),$$

where P^T means P transposed.

Proof: Left for the full paper. ∎

Lemma 7. *Let $G(V,E)$ and $G'(V,E')$ be regular graphs of degree d and d'. Assume that $G \subset G'$. (Hence $d' \geq d$.) Let P and P' be the transition probability matrices of G and G'. Then for every non-empty $S \subset V$*

$$F_S(P') \geq \frac{d}{d'} F_S(P).$$

∎

Theorem 8. *The swap chain $\{S_t\}$ gets close to the uniform distribution in time polynomial in the size of the underlying graph.*

Proof: Sketch. We use Lemma 5 and a coupling argument to obtain a lower bound on the ergodic flow for $\{B_t\}$. Then we "transform" the graph associated to $\{B_t\}$ to the graph associated to $\{S_t\}$, without reducing the ergodic flow too much. This gives a bound on the second eigenvalue of the transition probability matrix associated to $\{S_t\}$ which implies that $\{S_t\}$ is rapidly mixing. The details are left for the full paper. ∎

Acknowledgements

I am indebted to Persi Diaconis for the new proof of the Markov chain tree theorem, and to Jim Saxe for the simple proof of the cycle puzzle.
Very recently, David Aldous has independently discovered the same algorithm [Aldous88].

References

[Aldous88] D. Aldous, "A Random Walk Construction of Uniform Labeled Trees and Uniform Spanning Trees," manuscript, 1988.

[Alon86] N. Alon, "Eigenvalues and expanders," *Combinatorica*, **6**(1986), 83–96.

[AKLLR79] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovasz and C. Rackoff, "Random walks, universal traversal sequences, and the complexity of maze problems," *FOCS*, 1979, 218–233.

[AT88] V. Anantharan and P. Tsoucas, "A Proof of the Markov Chain Tree Theorem," Technical Report, Cornell, 1988.

[Broder86] A. Z. Broder, "How hard is it to marry at random? (On the approximation of the permanent)," *STOC*, 1986, 50–58. Erratum: *STOC*, 1988, p. 551.

[BK89] A.Z. Broder and A.R. Karlin, "Bounds on Cover Times," *Journal of Theoretical Probability*, **2**(1989), 101–120.

[BS87] A.Z. Broder and E. Shamir, "On the second eigenvalue of random regular graphs," *FOCS*, 1987, 286–294.

[CDN88] C.J. Colbourn, R.P.J. Day, and L.D. Nel, "Unranking and Ranking Spanning Trees of a Graph," *J. Algorithms*, to appear.

[CDM88] C.J. Colbourn, R.P.J. Day, and W. Myrvold, "Generating Random Spanning Trees Efficiently," manuscript, 1988.

[CRRST89] A.K. Chandra, P. Raghavan, W.L. Ruzzo, R. Smolensky, and P. Tiwari, "The Electrical Resistance of a Graph Captures its Commute and Cover Times," *STOC*, 1989, 574–586.

[DMLV88] P. Dagum, M. Luby, M. Mihail and U. Vazirani, "Polytopes, Permanents, and Graphs with Large Factors," *FOCS*, 1988, 412–421.

[Diaconis88] Personal communication.

[DFK88] M. Dyer, A. Frieze, and R. Kannan, "A random polynomial time algorithm for approximating the volume of convex bodies," *STOC*, 1989, 375–381.

[FK81] Z. Füredi and J. Komlós, "The eigenvalues of random symmetric matrices," *Combinatorica*, **1**(1981), 233–241.

[FKS89] J. Friedman, J. Kahn, and E. Szemeredi, "On the second eigenvalue of random regular graphs," *STOC*, 1989, 587–598.

[Guénoche83] A. Gu'enoche, "Random Spanning Tree," *J. Algorithms*, **4**(1983), 214–220.

[JS87] M. Jerrum and A. Sinclair, "Conductance and the Rapid Mixing Property for Markov Chains: the Approximation of the Permanent Resolved," *STOC*, 1988, 235–244.

[LR82] F.T. Leighton and R.L.Rivest, "The Markov Chain Theorem," MIT, 1983.

[KS69] J.G. Kemeny and J.L. Snell, *Finite Markov Chains*, Van Nostrand, 1969.

[Knuth73] D. E. Knuth, *The Art of Computer Programming*, Vol. 1, Second edition, Addison-Wesley, Reading, Mass., 1973.

[SJ89] A. Sinclair and M. Jerrum, "Approximate Counting, Uniform Generation, and Rapidly Mixing Markov Chains," *Information and Computation*, **82**(1989), 93–133.