SIMULATING SIZE-CONSTRAINED GALTON-WATSON TREES

Luc Devroye

September 5, 2011

ABSTRACT. We discuss various methods for generating random Galton-Watson trees conditional on their sizes being equal to n. A linear expected time algorithm is proposed.

KEYWORDS AND PHRASES. Random variate generation. Cayley trees. Catalan trees. Simulation. Galton-Watson branching process. Expected time analysis. Random trees.

1991 MATHEMATICS SUBJECT CLASSIFICATIONS: Primary 65C10. Secondary 65C05, 11K45, 68U20.

Author's address: Luc Devroye, School of Computer Science, McGill University, 3480 University Street, Montreal, Canada H3A 2K6. Research was supported by NSERC Grant A3456 and FQRNT Grant 90-ER-0291.

Introduction

A Galton-Watson tree (Athreya and Ney, 1972) is an ordered tree in which all nodes independently produce offspring distributed as ξ. They are called subcritical, critical, and supercritical, according to whether $\mathbb{E}\{\xi\}$ is < 1 (subcritical), $= 1$ (critical), or > 1 (supercritical). We exclude the trivial cases $\xi \equiv 1$ and $\xi \equiv 0$. The size of a Galton-Watson tree T is denoted by $|T|$. The purpose of this note is to discuss methods for the generation of T, conditional on $|T| = n$.

The size-conditioned Galton-Watson trees are important in combinatorial analysis, as they correspond (Kennedy, 1975) to the so-called "simply generated trees" of Moon and Meir (1970, 1978) (see also Moon, 1970). A random conditional Galton-Watson tree has the same distribution as a random simply generated tree picked uniformly from a set of such trees. That particular set depends, of course, on the distribution of ξ. For example, the distribution $(1/4, 1/2, 1/4)$ on $\{0, 1, 2\}$ yields a uniform random binary tree, or Catalan tree. The distribution $(1/3, 1/3, 1/3)$ on $\{0, 1, 2\}$ yields a uniform unary-binary tree, or Motzkin tree. The geometric distribution $1/2^{i+1}$, $i \geq 0$, yields the uniform random planted plane tree studied by DeBruijn and Knuth (1972). And the Poisson distribution yields a random rooted labeled tree, or rooted Cayley tree. For simply generated trees, no generally applicable efficient method has been published to our knowledge. The Boltzmann sampler of Duchon, Flajolet, Louchard and Schaeffer (2001, 2002, 2004) is a general linear time procedure that is easy to implement, but it yields random Galton-Watson trees of random size near n. Using rejection, one can generate random trees with their method until the size is right, but then the complexity is superlinear.

For random Catalan trees, several linear-time methods exist. They are often based on equivalences between these trees and other structures such as strings of n balanced parentheses, or simple walks of length $2n$ that remain positive, and start and end at the origin, or Dyck paths. Arnold and Sleep (1980) propose an elegant $O(n)$ algorithm for Catalan trees, which uses an incremental tree construction based on preserving uniformity every step of the way. Their algorithm is genetically linked to the general recursive method of Nijenhuis and Wilf (1978) (see also Flajolet, Zimmerman and van Cutsem, 1994). Another linear time algorithm is described in Alonso, Rémy and Schott (1997a, 1997b), who provide a slightly more general "codeword" method that also covers other trees. Related work can be found in Rémy (1985), Alonso (1992), Alnoso and Schott (1995), Mäkinen and Siltaneva (1999), Siltaneva (2000), Gouyou-Beauchamps (1993, 2003), Hickey and Cohen (1983), Banderier, Bousquet-Mélou, Denise, Flajolet, Gardy and Gouyou-Beauchamps (2002) and Barcucci, Del Lungo and Pergola (1992, 1999). Devroye (1986) provides an early survey of various methods for Catalan trees. See Mäkinen (1999) for another survey. Finally, Luczak and Winkler (2004) give an incremental algorithm for growing random Catalan trees one node at a time such that for each n, the partial trees are uniformly random.

Via the well-known equivalence between an ordered tree on n nodes and a binary tree on $n - 1$ nodes, we can thus also generate a random rooted ordered tree in linear time. Such trees correspond to conditional Galton-Watson processes with ξ geometrically distributed with parameter $1/2$.

The Cayley trees have an equally extensive history. Various representations exist that explicitly explain the number (n^{n-2}) of labeled free trees of size n, typically based on a one-to-one mapping between an $n-2$-vector drawn from $\{1, \ldots, n\}^{n-2}$ and such labeled (unrooted) trees. One of these is Pruefer's code (1918). Linear time algorithms can convert representations to trees and vice versa. For the Pruefer code, this was in the thesis of Klingsberg (1977) at the University of Washington (see also Devroye (1986)).

Random rooted labeled free trees, of which there are n^{n-1}, are in fact nothing but Galton-Watson trees conditioned on size n when the number of children has the Poisson distribution.

The purpose of the present note is to provide a universal linear expected time algorithm for all conditional Galton-Watson trees for which $\mathbb{E}\{\xi^2\} < \infty$.

Preliminaries

The distribution of ξ is determined by the probabilities $p_i = \mathbb{P}\{\xi = i\}$. Consider the family of distributions parametrized by $\theta > 0$, having $q_i = cp_i\theta^i$, $i \geq 0$, where $c = 1/\sum_i p_i\theta^i$ is a normalization constant. The range for θ is $(0, \rho)$, where ρ, possibly infinite, is the radius of convergence. It is well-known (Kennedy, 1975) that conditioning on $|T| = n$ makes all the trees in this parametrized family identically distributed—the value of θ does not matter! For this reason, but also for other reasons, it helps to pick a canonical member with mean one, the critical distribution. Thus, we assume throughout that $\mathbb{E}\{\xi\} = 1$, and that $p_1 \neq 1$.

There is also the thorny issue of the span d of ξ, the greatest common divisor of all $i \geq 1$ for which $p_i > 0$. If $d = 1$, then there exists n_0 such that for all $n \geq n_0$, $\mathbb{P}\{|T| = n\} > 0$, so that size conditioning is possible. If $d > 1$, then there exists n_0 such that for all $n \geq n_0$ such that $n - 1$ is a multiple of d, $\mathbb{P}\{|T| = n\} > 0$. We call $\mathcal{N}$ the set of integers n for which $\mathbb{P}\{|T| = n\} > 0$, and will assume that $n \in \mathcal{N}$.

The algorithms of this paper apply for all distributions of ξ. We are particularly interested in those for which ξ is not monoatomic (to avoid trivialities). A particularly important class is those for which

$$0 < \sigma^2 \stackrel{\text{def}}{=} \mathbb{V}\{\xi\} < \infty.$$

At the end of the paper, we will briefly deal with ξ that do not have finite variance or even finite mean.

Finally, many complexity results depend upon the parameters φ_n and τ_n defined by

$$\varphi_n = \mathbb{P}\{\xi_1 + \cdots + \xi_n = n - 1\}, \tau_n = \mathbb{E}\left\{\max(\xi_1, \ldots, \xi_n)\right\},$$

where $\xi_1, \ldots, \xi_n$, as elsewhere in the paper, are i.i.d. random variables distributed as ξ. It is well-known (see, e.g., Kolchin, 1986, or Petrov, 1975, 1995), that if $0 < \sigma < \infty$,

$$\varphi_n = \frac{(d + o(1))1_{[n\in\mathcal{N}]}}{\sigma\sqrt{2\pi n}}.$$

Assuming a RAM model of computation, and assuming that independent copies of ξ can be generated in expected time 1, the main result of the paper is as follows.

THEOREM 1. *There exists an algorithm for generating T conditional on $|T| = n$ in expected time bounded from above by a constant times*

$$n + \frac{1 + \tau_n}{\varphi_n}, n \in \mathcal{N}.$$

In particular, if $\mathbb{E}\{\xi^2\} < \infty$, then $\tau_n = o(\sqrt{n})$, $\varphi_n = \Theta(1/\sqrt{n})$ and thus, the expected time is $O(n)$.

In this paper, we were tempted to use the terminology “Bienaymé tree” instead of the widely accepted name “Galton-Watson tree”, as it increasingly clear that Bienaymé defined and derived the main properties of these trees almost fifty years before Galton and Watson (see, e.g., Kendall, 1975).

We first review two natural attempts that have superlinear expected complexity. The final algorithm uses ingredients from these simple methods, but adds a key ingredient—the multinomial method—, which permits the problem to be split into one of generating a certain multinomial random vector, and then applying a uniform random permutation. The first term (n) in the expected complexity of Theorem 1 comes from the uniform random permutation—it does not depend upon ξ. The τ_n/φ_n term in Theorem 1 comes from the multinomial part of the algorithm. Interestingly, if $\mathbb{E}\{\xi^2\} < \infty$, it is $o(n)$. Therefore, the speed of execution is basically determined by the uniform random permutation generator.

The naive method

A tree T can be traversed in DFS (depth first search) order. If we do so, we may keep a list of the number of offspring, $\xi_1, \xi_2, \ldots$. Vice versa, a tree T can be constructed from a sequence $\xi_1, \xi_2, \ldots$ in this manner. The latter process can be viewed sequentially: start with a root node and put it in a queue. At the i-th step, grab the first node from the queue, give it ξ_i children, and place these in the queue. The process ends when the queue is empty. It ends with a tree T of correct size n when the queue becomes empty for the first time after the n-th step.

Let T denote the tree generated by this process. From a sequence $\xi_1, \xi_2, \ldots, \xi_n$, we can generate T if $|T| \le n$ and decide that $|T| > n$ otherwise. Thus, stopping after the n-th step or when the queue becomes empty for the first time has complexity $\min(|T|, n)$.

We can repeat the above procedure this until for the first time $|T| = n$. By Wald’s lemma, the expected complexity is the expected number of iterations times $\mathbb{E}\{\min(|T|, n)\}$. The expected number of iterations is $1/\mathbb{P}\{|T| = n\}$. We know (see, e.g., Kolchin, 1986) that

$$\mathbb{P}\{|T| = n\} = \frac{\varphi_n}{n}.$$

And then,

$$\mathbb{E}\{\min(|T|, n)\} = \sum_{i=1}^{n} \varphi_i + n\mathbb{P}\{|T| > n\}.$$

By Wald’s identity, the expected complexity is asymptotic to

$$\frac{\mathbb{E}\{\min(|T|, n)\}}{\mathbb{P}\{|T| = n\}} = n \sum_{i=1}^{n} \frac{\varphi_i}{\varphi_n} + n^2 \sum_{i=n+1}^{\infty} \frac{\varphi_i}{i\varphi_n}.$$

Under the condition $0 < \sigma < \infty$, $n \in \mathcal{N}$, it is easy to see that $\varphi_n = \Theta(1/\sqrt{n})$, $\mathbb{P}\{|T| = n\} = \Theta(n^{-3/2})$, and thus that the expected complexity is $\Theta(n^2)$.

Turning to random walks

The size of the queue after the t-th step above is denoted by S_t. Thus, $S_0 = 1$, and

$$S_t = S_{t-1} + (\xi_t - 1) = 1 + \sum_{i=1}^{t} (\xi_i - 1), t > 0.$$

This provides the well-known random walk construction of Galton-Watson trees. The size of the Galton-Watson tree T generated in this manner is

$$|T| = \min\{t : S_t = 0\}.$$

But T is, of course, an unconditional Galton-Watson tree. For a conditional tree, we are interested in sequences $\xi_1, \xi_2, \ldots$ with $|T| = n$. A necessary condition for this is that

$$\Xi \stackrel{\text{def}}{=} (\xi_1, \ldots, \xi_n)$$

has sum

$$\mathcal{S}(\Xi) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \xi_i = n - 1.$$

This implies that $|T| \le n$. However, there is one and only one rotation of Ξ, i.e., a vector

$$\Xi(\ell) \stackrel{\text{def}}{=} (\xi_\ell, \xi_{\ell+1}, \ldots, \xi_n, \xi_1, \ldots, \xi_{\ell-1})$$

with the required property that $|T| = n$. This follows from the standard rotation argument for partial sums, sometimes referred to as the Dvoretzky-Motzkin cycle lemma (see, e.g., Comtet (1994) or Dershowitz and Zaks (1990)). Following Figure 1, ℓ is the smallest index in $\{1, \ldots, n\}$ at which S_t reaches its minimum:

$$\ell = \arg\min\{S_t : 1 \le t \le n\}.$$

It is easy to verify that $\Xi(\ell)$ has $S_t \ge 1$ for $0 \le t < n$ and $S_n = 0$. Furthermore, $|T| < n$ for all $\Xi(s)$, $s \ne \ell$.

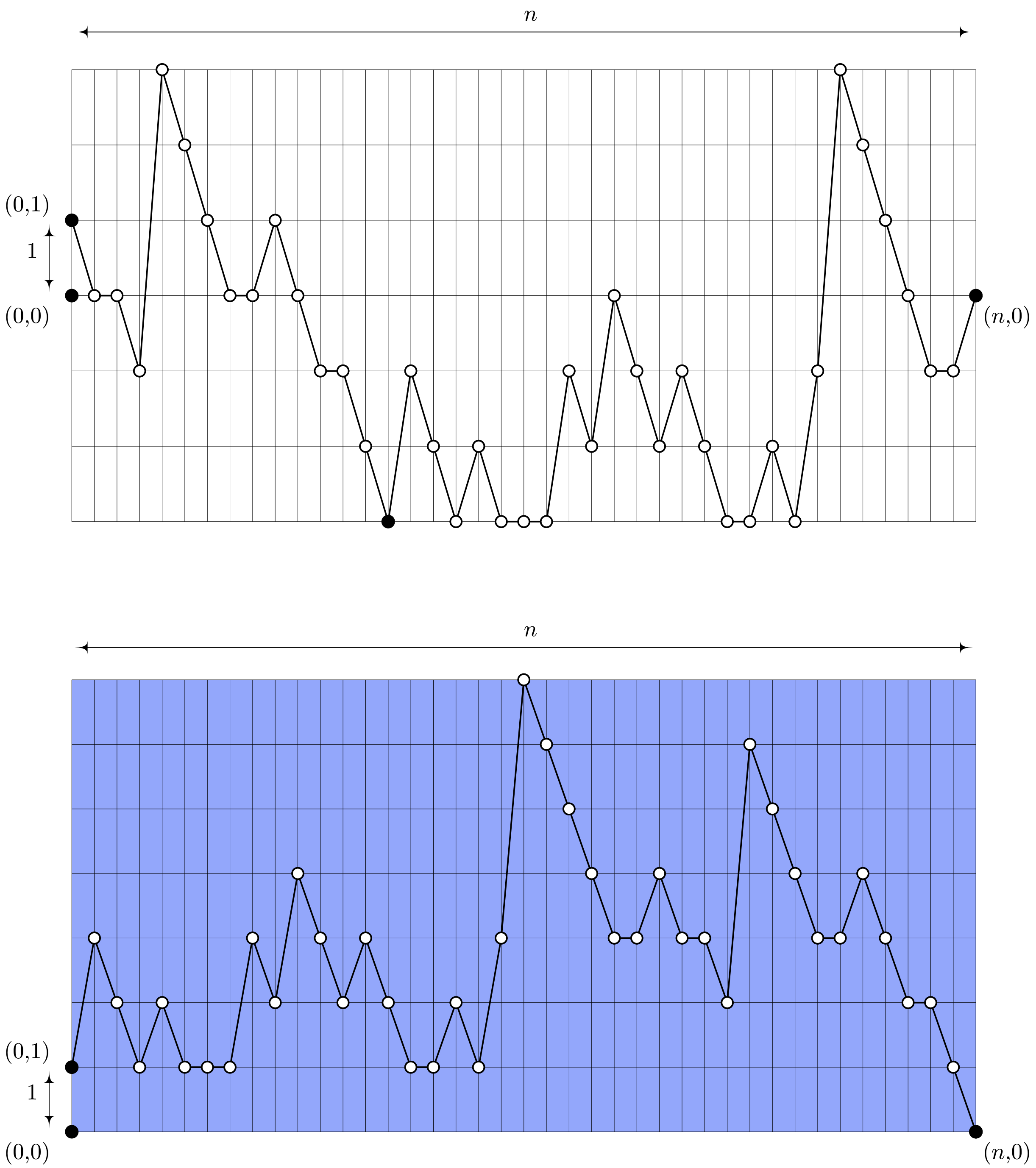

Figure 1. Top figure shows a random walk from $(0,1)$ to $(n,0)$. By starting the walk at the leftmost minimal node (blackened), the walk stays strictly positive until just before the last step, and thus corresponds uniquely to an ordered (Galton-Watson) tree of size n.

This observation implies a routine strategy for simulation: keep generating random sequences Ξ of length n until $\mathcal{S}(\Xi) = n-1$. Then rotate Ξ (in linear time) to get $\Xi(\ell)$ with $|T| = n$. The tree that corresponds to $\Xi(\ell)$ is a conditional Galton-Watson tree of size n.

It is well-known that if ξ is not monoatomic, then

$$\varphi_n \le \sup_x \mathrm{P}\{S_n = x\} \le \frac{c}{\sqrt{n}},$$

where $c > 0$ depends upon the distribution of ξ only. This follows, for example, from general upper bounds for the concentration of mass of sums of independent random variables: see, e.g., Petrov (1975, p. 49). When $\sigma^2 < \infty$, the order of this bound in n is correct, but for $\sigma = \infty$, the upper bound is $o(1/\sqrt{n})$ (Petrov, p. 46).

Thus, for any distribution, the procedure outlined here takes expected time

$$\frac{n}{\mathrm{P}\{\mathcal{S}(\Xi) = n-1\}} = \frac{n}{\varphi_n} = \Omega(n^{3/2}).$$

Furthermore, since $\mathrm{E}\{\xi = 1\}$, we have for increasing n drawn from $\mathcal{N}$, whenever $\sigma^2 < \infty$, $\varphi_n = \Theta(1/\sqrt{n})$ (see, e.g., Kolchin, 1986), and thus the expected time is $\Theta(n^{3/2})$.

Remark: Dwass's formula. This construction makes Dwass's formula (Dwass, 1969) explicit:

$$\mathrm{P}\{S_1 > 0, \ldots, S_{n-1} > 0, S_n = 0\} = \mathrm{P}\{|T| = n\} = \frac{1}{n}\mathrm{P}\{S_n = 0\} = \frac{1}{n}\mathrm{P}\{\xi_1 + \cdots + \xi_n = n-1\} = \frac{\varphi_n}{n}.$$

Generating random samples conditional on the sum

The previous section points up the importance of a fast method for generating a sequence

$$\Xi \stackrel{\text{def}}{=} (\xi_1, \ldots, \xi_n)$$

of i.i.d. integer-valued random variables distributed as $\xi \ge 0$, conditional on

$$\mathcal{S}(\Xi) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \xi_i = n-1.$$

This is a probem of independent interest. In some cases, there are simple explicit solutions. For example, when ξ is Poisson (λ) (for any fixed $\lambda > 0$), then the conditional law of Ξ is multinomial $(n-1, 1/n, \ldots, 1/n)$. This leads to an extremely simple procedure: generate $n-1$ i.i.d. random integers $Z_1, \ldots, Z_{n-1}$ uniformly drawn from $\{1, \ldots, n\}$, Then set

$$\xi_i = \sum_{j=1}^{n-1} 1_{[Z_j = i]}, 1 \le i \le n.$$

This yields the vector Ξ. By the rotation method of the previous section, we thus have a very simple linear time method for generating a random Cayley tree.

Let us concentrate however on generating $\mathcal{S}(\Xi)$. There is a literature on this, which was reviewed and summarized in Devroye (1986, 1988). Under certain conditions, and for certain computational models, one can generate $\mathcal{S}(\Xi)$ in expected time $O(1)$. However, for the purpose of the present paper, and this section, one simple paradigm stands out—the multinomial method. It provides a versatile tool for generating $\mathcal{S}(\Xi)$ in sublinear time. For this method, still using $p_i = \mathbb{P}\{\xi = i\}, i \ge 0$, we first generate the multinomial random vector $(N_0, N_1, N_2, \ldots)$ with parameter $(n, p_0, p_1, p_2, \ldots)$, and the note that

$$\mathcal{S}(\Xi) \stackrel{\mathcal{L}}{=} \sum_{i=0}^{\infty} i N_i.$$

We recall that a binomial (n, p) random variable can be generated in expected time bounded uniformly over n and p by a constant, thanks to algorithms developed in the literature. See, e.g., Ahrens and Dieter (1974, 1980), Devroye (1986, 1987), Hörmann (1993), Hörmann, Leydold and Derflinger (2004). Kachitvichyanukul and Schmeiser (1988, 1989), Schmeiser and Babu (1980), Stadlober (1988, 1989, 1990). Now, N_0 is binomial (n, p_0). Conditional on N_0, N_1 is binomial $(n - N_0, p_1/(1-p_0))$. Conditional on N_0 and N_1, N_2 is binomial $(n - N_0 - N_1, p_2/(1 - p_0 - p_1))$, and so forth. In this manner, we can generate the random multinomial vector $(N_0, \ldots, N_K)$, where K is the last populated (nonzero) component, i.e., $N_j = 0$ for $j > K$. The expected time for generating $(N_0, \ldots, N_K)$ is

$$\mathbb{E}\{1 + K\} = 1 + \mathbb{E}\left\{\max_{1 \le i \le n} \xi_i\right\} = 1 + \tau_n$$

It is a simple exercise to show that when $\mathbb{E}\{\xi\} < \infty$, then this is $o(n)$. However, the situation is typically much better. If ξ has compact support, then the expected time is $O(1)$. A simple bound can be derived in terms of the ρ-th moment, $\rho > 1$:

$$\begin{aligned}
\tau_n &\le \mathbb{E}\left\{\left(\sum_{1 \le i \le n} \xi_i^{\rho}\right)^{1/\rho}\right\} \\
&\le \left(\mathbb{E}\left\{\sum_{1 \le i \le n} \xi_i^{\rho}\right\}\right)^{1/\rho} \\
&= \left(n\mathbb{E}\left\{\xi^{\rho}\right\}\right)^{1/\rho} \\
&= O\left(n^{1/\rho}\right)
\end{aligned}$$

when $\mathbb{E}\{\xi^{\rho}\} < \infty$. We leave it as an exercise to show that $\tau_n = o(\left(n^{1/\rho}\right)$ under the latter condition.

Assume that we repeat the above procedure until for the first time $\mathcal{S}(\Xi) = n - 1$. Then the sum is correct, and we have, as a by-product, a random multinomial vector $(N_0, N_1, \ldots, N_K)$. This vector has the frequencies of occurrences of the ξ_i's, i.e., there are N_0 zeroes, N_1 ones, and so forth. Note that

$$\sum_{i=1}^{n} \xi_i = \sum_{j=0}^{K} j N_j = n - 1,$$

as required. The remainder of the algorithm is trivial: just fill an array of length n with N_j values j, $0 \le j \le K$, and randomly permute it. Random permutations of arrays are easy to implement in situ in linear time—see, e.g., Knuth (1981). The permuted array contains the sought vector Ξ.

By Wald's lemma, the expected time until we have $\mathcal{S}(\Xi) = n-1$ is bounded by

$$\frac{1+\tau_n}{\mathbf{P}\{\mathcal{S}(\Xi) = n-1\}} = \frac{1+\tau_n}{\varphi_n}.$$

For example, if $\mathbf{E}\{\xi^2\} < \infty$, then $\varphi_n = \Theta(1/\sqrt{n})$, and the numerator is $O(\sqrt{n})$, for a total of $O(n)$. However, if $\mathbf{E}\{\xi^\rho\} < \infty$ for fixed $\rho > 2$, then the numerator is $O(n^{1/\rho})$ as pointed out above, and the expected time bound becomes $O(n^{1/2+1/\rho})$.

REMARK: INFINITE VARIANCE. The behavior of φ_n depends upon ξ. If the variance of ξ is infinite, but $\mathbf{E}\xi = 1$, the complexity of the algorithm is still acceptable. To get an idea of this, let ϕ be the characteristic function of $\xi - 1$ (which is of mean zero). If d is the span of ξ, an inversion formula for the characteristic function (see, e.g., Petrov, 1995, p. 15) shows that

$$\varphi_n = \frac{d}{2\pi} \int_{|t|<\pi/d} e^{it} \phi^n(t)\, dt.$$

If ξ is in the domain of attraction of a stable of parameter $\alpha \in (1,2]$, and in particular, if $\phi(t) = \exp(-|t|^\alpha(1+o(1))$, then standard calculations show that

$$\varphi_n = \Theta\left(n^{-1/\alpha}\right). \square$$

REMARK: INFINITE MEAN. If $\mathbf{E}\xi = \infty$, while the algorithm is still valid, the complexity becomes rather unpleasant. To deal with this case, a lot more work is needed to deal with the efficient generation of size-constrained Galton-Watson trees. □

Generating random forests

Random forests can be defined in a number of ways, see, e.g., Pavlov (2000). In the context of the present paper, the most important model is that of a random forest of k Galton-Watson trees of total size n. Each tree has at least a root, and thus, $n \ge k$ is understood. If the span of ξ is d, then $n-k$ is necessarily a multiple of d. The set of all n that are possible, given the distribution of ξ, is $\mathcal{N}$, and $n \in \mathcal{N}$ is understood throughout.

The random forest of interest to us us a collection of non-empty independent Galton-Watson trees $T_1, \ldots, T_k$ conditional on $|T_1| + \cdots + |T_k| = n$. Here too we can make a unique connection with sequences $\xi_1, \ldots, \xi_n$ drawn independently from the distribution of ξ. Each such sequence corresponds to such a random forest provided that

$$\sum_{i=1}^n \xi_i = n-k.$$

The construction generalizes that for the random tree in a natural manner. Random walks are now started with a queue having k elements: $S_0 = k$. Then we proceed as before, and form

$$S_t = k + \sum_{i=1}^t (\xi_i - 1).$$

Note that at $t = n$, we have precisely k trees in the forest if and only if $S_t > 0$ for all $t < n$ and $S_n = 0$. Since S_t decreases by at most one, this means that if this condition is satisfied, S_t passes through each of the values $k-1$, $k-2$, …, 1 and 0 for the first time, say at $t = t_{k-1}$, $t = t_{k-2}$, …, $t = t_0$. Each of these integers wraps up the construction of one tree in the forest, so the tree sizes are t_{k-1}, $t_{k-2} - t_{k-1}$, …, $t_0 - t_1$.

If $S_n = 0$, the condition $S_t > 0$ for all $t < n$ may not be satisfied. When this happens, there is one (and only one) rotation that insures that for the rotated sequence, $S_t > 0$ for all $t < n$. This is illustrated in Figure 2: find the first $t \geq 0$ for which $S_t = \min_{1 \leq i \leq n} S_i$. Then rotate by starting a new random walk at t. This walk, and only this one among all rotated walks, has the desired property. It is then easy to see that we can apply all algorithms of this paper towards random forest generation. For the analysis, note that the crucial parameter φ_n now is

$$\varphi_n = \mathbf{P}\left\{\xi_1 + \cdots + \xi_n = n - k\right\}.$$

It is well-known (see, e.g., Kolchin, 1986, or Petrov, 1975, 1995), that if $0 < \sigma < \infty$,

$$\varphi_n = \frac{(d + o(1)) e^{-k^2/(2\sigma^2 n)} \mathbf{1}_{[n \in \mathcal{N}]}}{\sigma\sqrt{2\pi n}}.$$

This behaves as the old φ_n when $k = o(\sqrt{n})$. Interestingly, the expected complexity for finite σ is $O(n)$ even if k varies with n such that $k = o(\sqrt{n})$.

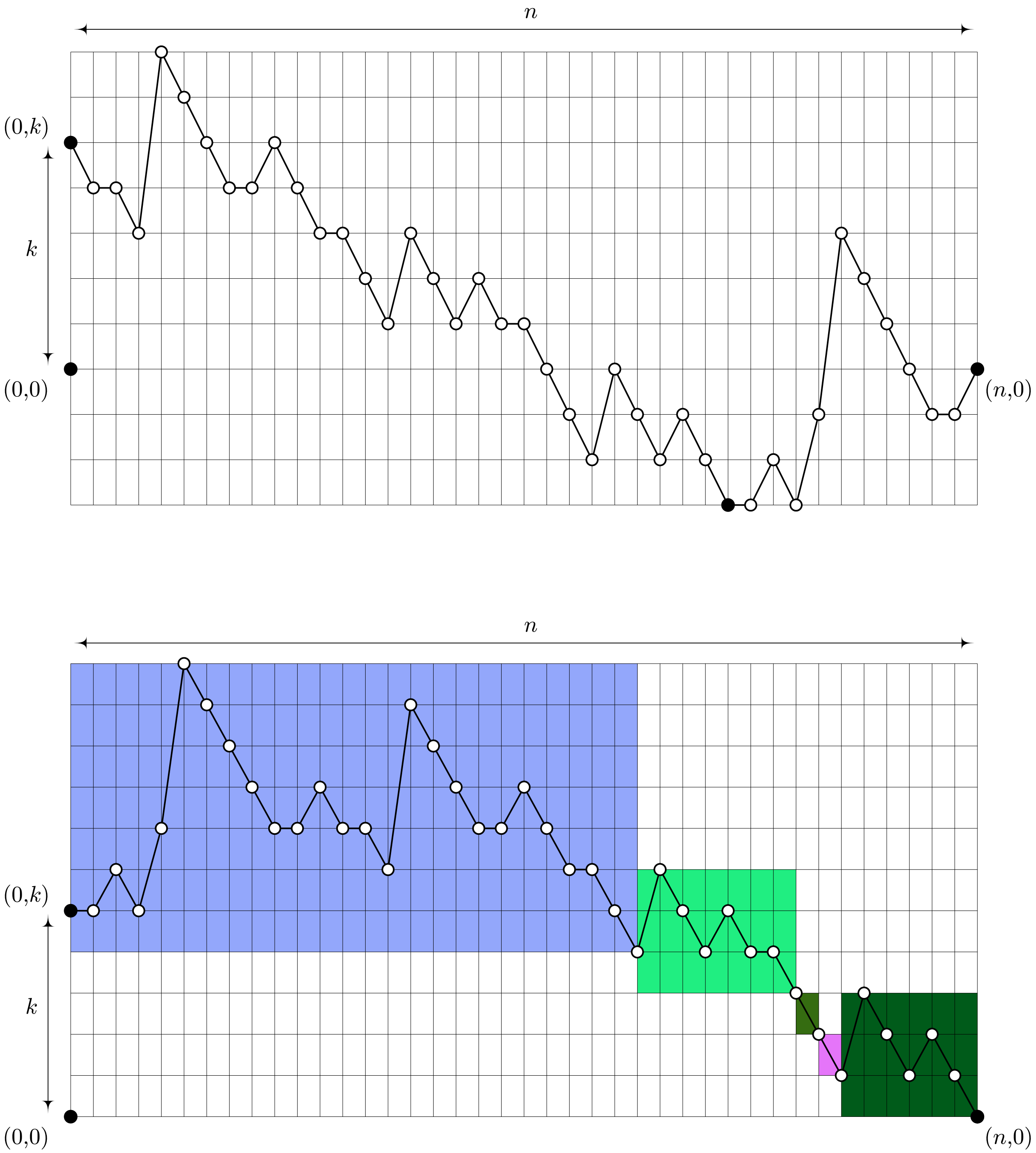

Figure 2. Top figure shows a random walk from $(0, k)$ to $(n, 0)$. By starting the walk at the leftmost minimal node (blackened), the walk results in a clean separation (bottom) into k trees. For each tree, as we know, a walk started at $(0, 1)$ ends when for the first time a node of height 0 is reached.

References

J. H. Ahrens and U. Dieter, “Computer methods for sampling from gamma, beta, Poisson and binomial distributions,” *Computing*, vol. 12, pp. 223–246, 1974.

J. H. Ahrens and U. Dieter, “Sampling from binomial and Poisson distributions: a method with bounded computation times,” *Computing*, vol. 25, pp. 193–208, 1980.

D. Aldous, “The random walk construction of uniform spanning trees and uniform labelled trees,” *SIAM Journal of Discrete Mathematics*, vol. 3, pp. 450–465, 1990.

D. Aldous, “Asymptotic fringe distributions for general families of random trees,” *The Annals of Applied Probability*, vol. 1, pp. 228–266, 1991.

L. Alonso, “Structures arbortescentes, algorithmes de génération, problème d’inclusion, relations maximin,” Thèse de Ph.D., Université Paris-Sud, Orsay, 1992.

L. Alonso and R. Schott, *Random Generation of Trees*, Kluwer, Boston, 1995.

L. Alonso, J. L. Rémy, and R. Schott, “Uniform generation of a Schröder tree,” *Information Processing Letters*, vol. 64, pp. 305–308, 1997.

L. Alonso, J. L. Rémy, and R. Schott, “A linear-time algorithm for the generation of trees,” *Algorithmica*, vol. 17, pp. 162–182, 1997.

D. B. Arnold and M. R. Sleep, “Uniform random number generation of n balanced parenthesis strings,” *ACM Transactions on Programming Languages and Systems*, vol. 2, pp. 122–128, 1980.

K. B. Athreya and P. E. Ney, *Branching Processes*, Springer Verlag, Berlin, 1972.

C. Banderier, M. Bousquet-Mélou, A. Denise, P. Flajolet, D. Gardy, and D. Gouyou-Beauchamps, “Generating functions for generating trees,” *Discrete Mathematics*, vol. 246, pp. 29–55, 2002.

E. Barcucci, R. Pinzani, and R. Sprugnoli, “The random generation of directed animals,” *Theoretical Computer Science*, vol. 127, pp. 333–350, 1992.

E. Barcucci, A. Del Lungo, and E. Pergola, “Random generation of trees and other combinatorial objects,” *Theoretical Computer Science*, vol. 218, pp. 219–232, 1999.

A. Cayley, “A theorem on trees,” *Quarterly Journal of Mathematics*, vol. 23, pp. 376–378, 1889.

L. Chottin and R. Cori, “Une preuve combinatoire de la rationalité d’une série génératrice associée aux arbres,” *RAIRO Informatique Théorique*, vol. 16, pp. 113–128, 1982.

L. Comtet, *Advanced Combinatorics*, Reidel, Dordrecht, 1994.

N. G. de Bruijn, D. E. Knuth, and S. O. Rice, “The average height of planted plane trees,” in: *Graph Theory and Computing*, edited by R. C. Read, pp. 15–22, Academic Press, 1972.

A. Denise, “Méthodes de génération aléatoire d’objets combinatoires de grande taille et problèmes d’énumération,” Thèse, Université Bordeaux I, France, 1994.

N. Dershowitz and S. Zaks, “The cycle lemma and some applications,” *European Journal of Combinatorics*, vol. 11, pp. 35–40, 1990.

L. Devroye, *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.

L. Devroye, “A simple generator for discrete log-concave distributions,” *Computing*, vol. 39, pp. 87–91, 1987.

L. Devroye, “Generating sums in constant average time,” in: *Proceedings of the 1988 Winter Simulation Conference*, edited by M. A. Abrams, P. L. Haigh and J. C. Comfort, pp. 425–431, IEEE, San Diego, CA., 1988.

P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer, “Random sampling from Boltzmann principles,” INRIA Technical Report ALCOMFT-TR-01-189, 2001.

P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer, “Random sampling from Boltzmann principles,” in: *Automata, Languages and Programming (ICALP 2002)*, edited by P. Widmayer, F. Triguero, R. Morales, M. Hennessy, S. Eidenbenz and R. Conejo, vol. 2380, pp. 501–513, Lecture Notes in Computer Science, Springer-Verlag, New York, 2002.

P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer, “Boltzmann samplers for the random generation of combinatorial structures,” *Combinatorics, Probability and Computing*, vol. 13, pp. 577–625, 2004.

M. Dwass, “The total progeny in a branching process,” *Journal of Applied probability*, vol. 6, pp. 682–686, 1969.

P. Flajolet, P. Zimmerman, and B. van Cutsem, “A calculus for the random generation of labelled combinatorial structures,” *Theoretical Computer Science*, vol. 132, pp. 1–35, 1994.

D. Gouyou-Beauchamps, *Quelques exemples d’algorithmes de génération aléatoire*, 1993.

D. Gouyou-Beauchamps, “Combinatorics and random generation,” in: *Algorithms Seminar 2001-2002*, edited by F. Chyzak, pp. 177–182, INRIA, 2003.

C. C. Heyde and E. Seneta, *I.J. Bienaymé: Statistical Theory Anticipated*, Berlin, 1977.

T. Hickey and J. Cohen, “Uniform random generation of strings in a context-free language,” *SIAM Journal on Computing*, vol. 12, pp. 645–655, 1983.

W. Hörmann, “The generation of binomial random variates,” *Journal of Statistical Computation and Simulation*, vol. 46, pp. 101–110, 1993.

W. Hörmann, J. Leydold, and G. Derflinger, *Automatic Nonuniform Random Variate Generation*, Springer-Verlag, Berlin, 2004.

S. Janson, T. Luczak, and A. Rucinski, *Random Graphs*, Wiley-Interscience, New York, 2000.

V. Kachitvichyanukul and B. W. Schmeiser, “Binomial random variate generation,” *Communications of the ACM*, vol. 31, pp. 216–222, 1988.

V. Kachitvichyanukul and B. W. Schmeiser, "Algorithm 678: BTPEC: sampling from the binomial distribution," *ACM Transactions on Mathematical Software*, vol. 15, pp. 394–397, 1989.

D. G. Kendall, "The genealogy of branching processes before (and after) 1873," *Bulletin of the London Mathematical Society*, vol. 7, pp. 225–254, 1975.

D. P. Kennedy, "The Galton-Watson process conditioned on the total progeny," *Journal of Applied Probability*, vol. 12, pp. 800–806, 1975.

D. E. Knuth, *The Art of Computer Programming, Vol. 2*, Addison-Wesley, Reading, Mass., 1981. 2nd Ed..

D. E. Knuth, *The Art of Computer Programming. Vol. 1, Fundamental Algorithms, Third Edition*, Addison-Wesley, Reading, MA, 1997.

V. F. Kolchin, *Random Mappings*, Optimization Software Inc., New York, 1986.

M. Lothaire, "Combinatorics on Words," vol. 17, Encyclopedia of Mathematics and its Applications, Addison-Wesley, 1983.

M. Luczak and P. Winkler, "Building uniformly random subtrees," *Random Structures and Algorithms*, vol. 27, pp. 420–443, 2004.

E. Mäkinen, "Generating random binary trees—a survey," *Information Sciences*, vol. 115, pp. 123–136, 1999.

E. Mäkinen and J. Siltaneva, "A note on Rémy's algorithm for generating random binary trees," Technical Report, Department of Computer and Information Sciences , University of Tampere, Finland, 1999.

U. Manber, *Introduction to Algorithms: A Creative Approach*, Addison-Wesley,, Reading, MA, 1989.

A. Meir and J. W. Moon, "The distance between points in random trees," *Journal of Combinatorial Theory*, vol. 8, pp. 99–103, 1970.

A. Meir and J. W. Moon, "On the altitude of nodes in random trees," *Canadian Journal of Mathematics*, vol. 30, pp. 997–1015, 1978.

J. W. Moon, *Counting Labelled Trees*, Canadian Mathematical Congress, Montreal, 1970.

A. Nijenhuis and H. S. Wilf, *Combinatorial Algorithms, 2nd edition*, Academic Press, New York, 1978.

Yu. L. Pavlov, *Random Forests*, VSP, Utrecht, 2000.

V. V. Petrov, *Sums of Independent Random Variables*, Springer-Verlag, Berlin, 1975.

V. V. Petrov, *Limit Theorems of Probability Theory*, Oxford Science Publications, Clarendon Press, Oxford, 1995.

A. Prüfer, "Neuer Beweis eines Satzes über Permutationen," *Archiv der Mathematik und Physik*, vol. 3, pp. 142–144, 1918.

J. L. Rémy, "Un procédé itératif de dénombrement d'arbres binaires et son application à leur génération aléatoire," *RAIRO Theoretical Informatics and Applications*, vol. 19, pp. 179–195, 1985.

C. Savage, "A survey of combinatorial Gray codes," *Society of Industrial and Applied Mathematics Review*, vol. 39, pp. 605–629, 1997.

B. W. Schmeiser and A. J. G. Babu, "Beta variate generation via exponential majorizing functions," *Operations Research*, vol. 28, pp. 917–926, 1980.

J. Siltaneva, "Random Generation of Binary Trees (In Finnish)," Master's Thesis, Department of Computer and Information Sciences, University of Tampere, Finland, 2000.

E. Stadlober, "Sampling from Poisson, binomial and hypergeometric distributions: ratio of uniforms as a simple fast alternative," Habilitationsschrift, Institute of Statistics, Technical University of Graz, Austria, 1988.

E. Stadlober, "Binomial random variate generation: a method based on ratio of uniforms," *American Journal of Mathematical and Management Sciences*, vol. 9, pp. 1–20, 1989.

E. Stadlober, "The ratio of uniforms approach for generating discrete random variates," *Journal of Computational and Applied Mathematics*, vol. 31, pp. 181–189, 1990.

J. S. Vitter and P. Flajolet, "Analysis of algorithms and data structures," in: *Handbook of Theoretical Computer Science*, edited by J. van Leeuwen, pp. 431–524, North Holland, 1990.

H. S. Wilf, *Combinatorial Algorithms: an Update*, SIAM, Philadelphia, PA, 1989.