Discrete Comput Geom 22:105–117 (1999)

Discrete & Computational
Geometry

Properties of Random Triangulations and Trees*

L. Devroye,[1] P. Flajolet,[2] F. Hurtado,[3] M. Noy,[3] and W. Steiger[4]

[1]School of Computer Science, McGill University,
Montreal, Quebec, Canada H3A 2K6
luc@kriek.cs.mcgill.ca

[2]INRIA,
Rocquencourt, France
Philippe.Flajolet@inria.fr

[3]Universitat Politecnica de Catalunya,
Spain
{hurtado,noy}@ma2.upc.es

[4]Department of Computer Science, Rutgers University,
New Brunswick, NJ 08903, USA
steiger@cs.rutgers.edu

Abstract. Let T_n denote the set of triangulations of a convex polygon K with n sides. We study functions that measure very natural "geometric" features of a triangulation $\tau \in T_n$, for example, $\Delta_n(\tau)$ which counts the maximal number of diagonals in τ incident to a single vertex of K. It is familiar that T_n is bijectively equivalent to B_n, the set of rooted binary trees with $n-2$ internal nodes, and also to P_n, the set of nonnegative lattice paths that start at 0, make $2n-4$ steps X_i of size ± 1, and end at $X_1 + \cdots + X_{2n-4} = 0$. Δ_n and the other functions translate into interesting properties of trees in B_n, and paths in P_n, that seem not to have been studied before. We treat these functions as random variables under the uniform probability on T_n and can describe their behavior quite precisely. A main result is that Δ_n is very close to $\log n$ (all logs are base 2). Finally we describe efficient algorithms to generate triangulations in T_n uniformly, and in certain interesting subsets.

* Part of the work of W. Steiger was supported by a grant from the Generalitat de Catalunya to UPC and part by NSF Grant CCR-9111491. The work of L. Devroye was supported by NSERC Grant A3456. The work of Philippe Flajolet was supported by the L.T.R. Project Alcom-IT (# 20244) of the European Union.

1. Introduction and Summary

Consider a convex polygon K with n sides. We label the vertices $v_i = i, i = 0, \ldots, n-1$, in clockwise order. A triangulation is a set of $n-3$ noncrossing diagonals $v_i v_j$ which partitions K into $n-2$ triangles. You can imagine constructing a triangulation τ recursively: taking the polygon edge $v_0 v_{n-1}$ as base, just choose the apex of the triangle of τ that it belongs to, say v_i, $0 < i < n-1$, and now continue in the same way on the two polygons $v_0, \ldots, v_i$ with $v_0 v_i$ as base and $v_i, \ldots, v_{n-1}$ with $v_i v_{n-1}$ as base (Fig. 1 has $n = 8$ and $v_i = 2$, 4, and 5, respectively). This shows that t_n, the number of such triangulations, satisfies

$$t_n = t_2 t_{n-1} + t_3 t_{n-2} + \cdots + t_{n-1} t_2 \qquad t_2 = t_3 = 1,$$

the recursion of the Catalan numbers. Therefore

$$t_n = C_{n-2} = \frac{1}{n-1}\binom{2n-4}{n-2} \tag{0}$$

is the size of T_n, the set of triangulations of K.

It is natural to consider certain "geometric" features of a triangulation $\tau \in T_n$. Let d_i denote the *degree* of vertex v_i, the number of diagonals of τ incident with v_i. It is easy to see [10] that τ is characterized by this sequence of degrees. In this paper we study

$$\Delta_n(\tau) = \max(d_i,\ i = 0, \ldots, n-1), \tag{1}$$

the **maximal degree** of the vertices. $\Delta_n = 2$ when τ is a *zigzag* and $n-3$ when it is a *fan* ($d_i = n-3$ for some vertex), as in Fig. 1.

Define the length of a diagonal $v_i v_j$ with $i > j$ to be $\|v_i v_j\| = \min(i-j, n-i+j)$, the (fewest) number of successive edges of K between the endpoints. Another geometric feature of τ that we look at is

$$\lambda_n(\tau) = \max(\|v_i v_j\| \ : v_i v_j \in t), \tag{2}$$

the **length of the longest diagonal** in the triangulation. It is clear that $n/3 \le \lambda_n \le n/2$.

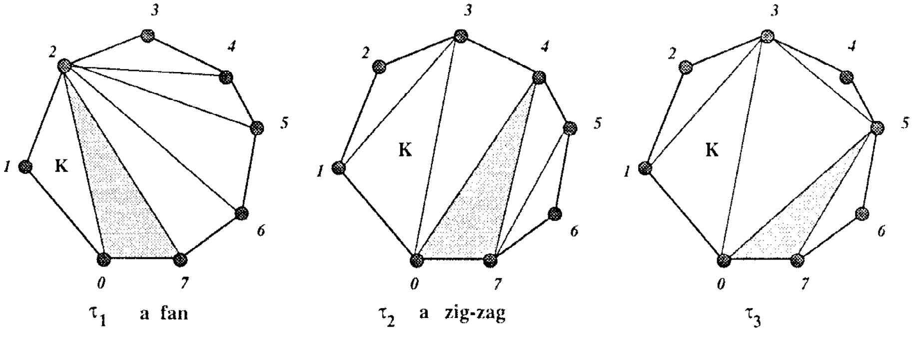

Fig. 1. Three triangulations: $\Delta_8 = 5$, 2, and 3, respectively; $\lambda_8 = 4$, 4, and 3, respectively.

To see how these functions behave across the family of triangulations we treat them as random variables under the uniform probability on T_n. By symmetry, each d_i has the same distribution, but they are not independent because, e.g., $d_0 + \cdots + d_{n-1} = 2(n-3)$. In view of the fact that the expected degree of each v_i is $2(1-3/n)$, it may be somewhat surprising that Δ_n is close to $\log n$ (all logs are base 2). The main result is

Theorem 1. *As $n \to \infty$,*

$$E(\Delta_n)/\log n \to 1.$$

In fact Δ_n is strongly concentrated. For all $c > 0$, as $n \to \infty$,

$$\text{Prob}\{|\Delta_n(\tau) - \log n| \leq (1+c)\log\log n\} \to 1.$$

The upper bound is based on computing the distribution of d_i.

Lemma 1. *For each vertex v_i, the probability that its degree is k is given by*

$$\text{Prob}(d_i = k) = \left(\frac{k+1}{2}\right)\left(\frac{n-1}{2n-5}\right)\prod_{i=1}^{k}\frac{n-2-i}{2n-5-i}\,. \tag{3}$$

Remark 1. Since $\text{Prob}(d_i = k) \leq (k+1)2^{-(k+1)}$ when $n > 3$ (which we assume throughout), this says that d_i has tails that decrease geometrically fast. Theorem 1 indicates that their maximum is logarithmic, like the max of n **independent** geometric random variables (see also Final Remark 1). The proof makes these connections more explicit. It is interesting to wonder about the variance of Δ_n. Simulation indicates that it could be constant.

The key fact about the longest diagonal is

Lemma 2. *The distribution of the length of the longest diagonal is given by*

$$\text{Prob}(\lambda_n = k) = \frac{nC_{k-1}}{C_{n-2}}\sum_{i=n-k}^{2k_{(*)}}{}_{(*)}C_{i-k-1}C_{n-i-1}, \tag{4}$$

where $(*)$ means "multiply the summand by $\frac{1}{2}$ when $i = 2k$ and $i = n-k$, unless $3k = n$, when we multiply by $\frac{1}{3}$."

This enables us to find the limit distribution of λ_n.

Theorem 2. *For each $x \in (\frac{1}{3}, \frac{1}{2})$, as $n \to \infty$,* $\text{Prob}(\lambda_n \leq nx) \to$ *to the distribution with density*

$$w(x) = \frac{1}{\pi}x^{-2}(1-x)^{-2}(3x-1)(1-2x)^{-1/2}.$$

In addition $E(\lambda_n)/n \to \alpha$, *where*

$$\alpha = \frac{\sqrt{3}}{\pi} + \frac{1}{3} - \frac{\log(2+\sqrt{3})}{\pi} = 0.4654615104\ldots.$$

A motivation for the present work—along with deep curiosity about how typical triangulations look—is the inherent interest of binary trees. It is familiar that T_n is bijectively equivalent to B_n, the set of rooted binary trees with $n-2$ internal nodes, each triangulation $\tau \in T_n$ corresponding to a particular tree $b(\tau) \in B_n$. The two features of triangulations that we study translate into interesting and natural properties of the corresponding trees. For example, $\Delta_n(\tau)$ measures a property of $b(\tau)$ that we call the *external-node separation*, $\chi_n(b(\tau))$: this is the maximal distance in the tree between successive external nodes. $\lambda_n(\tau)$ measures a property of $b(\tau)$ that we call the *nearly half* measure, $H_n(b(\tau))$: it is the size of the largest subtree with not more than half the external nodes. Though trees have been studied intensively (e.g., [3], [4], [8], [11], and [12]), we are unaware of any previous work on these two features. Theorems 1 and 2 and Lemmas 1 and 2 thus appear to express interesting, new facts about trees, as well as about triangulations. In Section 2 we translate the functions Δ_n and λ_n into the context of binary trees. We also exploit the correspondence between T_n and nonnegative lattice paths P_n; we interpret our functions in this set as well, to help with the proofs, which appear in Section 3.

In Section 4 we describe some linear-time algorithms to generate elements of T_n randomly. In addition let $T_n(k)$ denote the subset of triangulations in T_n with $\lambda_n(\tau) = k$. We show how to generate quickly triangulations restricted to $T_n(k)$. Finally, if $d_i = 0$, the vertex v_i is called *an ear* of the triangulation. We show how to generate quickly triangulations with a given number of ears. This may be of some interest because ears of τ correspond to leaves of $b(\tau)$.

Remark 2. If we regard the trees in B_n as binary-search trees generated by permutations of $1, \ldots, n-2$, each permutation being equally likely, the bijection gives the (binary-search tree) probability β, on T_n. Trees in B_n are well studied in this model (e.g., [4], [6], [11], and [13]). In contrast to the situation in the uniform distribution, the vertex degrees in this model are not identically distributed. Actually $E_\beta(d_0) = E_\beta(d_{n-1}) = \Theta(\log n)$ as is familiar from [4] and [6]. We can prove that in this distribution $\Delta_n/\log n \to c > 1$ in probability. It seems difficult to analyze λ_n in this model.

2. Preliminaries

We first describe the explicit correspondences between triangulations, trees, and paths that we use. The standard way to associate a tree with a triangulation uses the dual graph of $\tau \in T_n$. This gives a binary tree with $n-2$ internal nodes, one for each triangle of τ; adjacent triangles of τ correspond to nodes joined by an edge of the tree. The triangle with edge $v_0 v_{n-1}$ is associated with the root of the tree. If v_i is the apex of this triangle in τ, label the root with i. The left subtree has $i-1$ internal nodes (the number of vertices

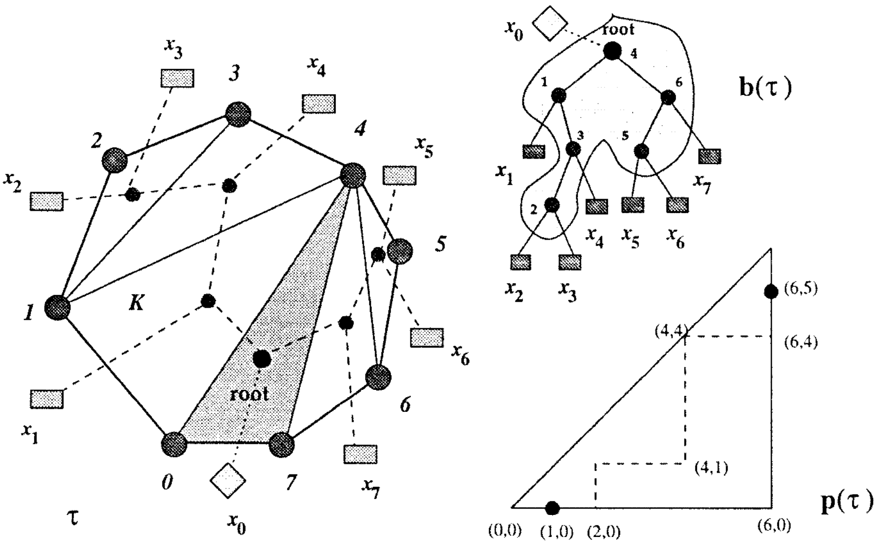

Fig. 2. A triangulation of K and two equivalent representations.

of K between v_0 and v_i) and corresponds to the triangulation of subpolygon $v_0, \ldots, v_i$ in τ; the right subtree has $n - 2 - i$ internal nodes and corresponds to the triangulation of subpolygon $v_i, \ldots, v_{n-1}$ in τ, and now continue recursively in the two subpolygons (subtrees). Once the $n - 2$ internal nodes are placed, external nodes are added so internal nodes have outdegree 2. Call this (binary-search) tree $b(\tau)$. It has $n - 1$ external nodes whose *inorder* traversal corresponds to the edges $v_{i-1}v_i$, $i = 1, \ldots, n-1$. The root misses an external node corresponding to $v_0 v_{n-1}$. We label them x_i, $i = 1, \ldots, n-1$, and the missing external node, x_0 (see Fig. 2). This scheme defines a bijection $T_n \leftrightarrow B_n$.

For $\tau \in T_n$ construct a path $p(\tau) \in P_n$ as follows (we think of elements of P_n as upright rectilinear paths joining points in the integer lattice in R^2 contained in the triangle bounded by the x-axis, $x = n - 2$, and $y = x$). Paths start at $(1, 0)$ and end at $(n - 2, n - 3)$. Suppose τ has j_0 internal diagonals incident to v_0; then the path moves *right* j_0 steps. In general, let j_i denote the number of diagonals from v_i to a higher number vertex. We are currently at vertex v_0. We move clockwise in K to the next vertex v_i with $j_i > 0$. The path moves *up* to the line $y = i$ and then moves *right* for j_i steps. It is easy to see that this procedure gives a path in P_n and that every such path comes from a distinct triangulation. These bijections are frequently exploited when studying the combinatorics in one of these sets (see especially [15]), and also for the task of randomly generating elements from one of the sets (e.g., [2], [5], and [13]).

To understand what Δ_n says about trees, imagine the diagonal $v_i v_j$ in τ as directed from the smaller numbered vertex of K to the larger one. Take $0 < i < n - 1$ and move counterclockwise along the circumference of a sufficiently small circle centered at vertex v_i from edge $v_{i-1}v_i$ to edge $v_i v_{i+1}$. First we meet diagonals (if any) coming from lower vertices *into* v_i and then we meet diagonals (if any) going *out from* v_i to higher vertices. This shows that the degree of v_i in τ is the number of nodes in $b(\tau)$ *between* x_i and α, and the number of nodes between x_{i+1} and α, α being the root of the smallest subtree containing x_i and x_{i+1}; thus it is the path distance in $b(\tau)$ (number of internal nodes) from x_i to x_{i+1}, minus 1. Similarly, and because x_0 is missing from rooted binary trees, d_0 and d_{n-1} count the number of internal nodes between the root and x_1 and the root and x_{n-1}, respectively.

Given a rooted binary tree b with $n-2$ internal nodes and external nodes $x_1, \ldots, x_{n-1}$, tack an external node x_n ($\equiv x_0$) onto the root and define the *external-node separation* by

$$\chi_n(b) = \max(\|x_i x_{i+1}\|,\ i = 0, \ldots, n-1), \tag{5}$$

where $\|x_i x_{i+1}\|$ counts the path distance minus 1 between the external nodes. We just argued that

Lemma 3. *Given* $\tau \in T_n$, $\Delta_n(\tau) = \chi_n(b(\tau))$.

It is more difficult to interpret Δ_n on paths. From the construction of $p(\tau)$ the width j_i of the step along $y = i$ is the outdegree of vertex v_i, $i = 0, \ldots, n-3$ ($j_i = 0$ means the path has no step at level i). The outdegrees of v_{n-2} and v_{n-1} are zero. Similarly, the indegrees of v_0 and v_1 are zero. The other indegrees are more complicated, except for v_{n-1}, where the indegree equals d_{n-1}, and both count the number of times the path meets $y = x$, from $x = 1$ to $x = n-3$. Also both d_{n-2} and the indegree of v_{n-2} can be determined from the intersections of the path with $y = x - 1$. However, in general, d_i seems not to be an easily "seen" feature of the path.

Given a path $p \in P_n$, define its step-width by

$$s_n(p) = \max(j_i, i = 0, \ldots, n-3), \tag{6}$$

where j_i is the width of the step of p at height $y = i$. Since $d_i \geq j_i$,

Lemma 4. *Given a triangulation* $\tau \in T_n$, $\Delta_n(\tau) \geq s_n(p(\tau))$.

Therefore probabilistic lower bounds for step-width imply lower bounds for the maximum degree.

It is straightforward to interpret λ_n. From the construction of $b(\tau)$ from τ, each internal node in the tree other than the root corresponds to the part of τ restricted to some subpolygon $v_i, \ldots, v_j$, $i < j-1$. Therefore $\|v_i v_j\|$ corresponds to the number of external nodes in the subtree rooted at that particular internal node. Given a tree $b \in B_n$, denote its nonroot internal nodes by ν_i and define $\|\nu_i\|$ as the number of external nodes in the subtree rooted at ν_i. The "nearly-half measure" of b is defined by

$$H_n(b) = \max(\min(\|\nu_i\|, n - \|\nu_i\|),\ i = 1, \ldots, n-3). \tag{7}$$

Its the size of the largest subtree with not more than half the external nodes. Because $\lambda_n(\tau) = H_n(b(\tau))$, Lemma 2 gives the distribution of this random variable on trees.

3. Proofs

We sketch the proofs of the results mentioned previously. A main tool is the ballot theorem (see p. 73 of [7]) which says that the number of lattice paths that start at (0, 0), make i unit steps to the right, $j \le i$ unit steps up, and preserve $y \le x$ is

$$N_{ij} = \frac{i+1-j}{i+1+j}\binom{i+1+j}{j}. \tag{8}$$

Proof of Lemma 1. Since the degrees are identically distributed, we only have to consider vertex v_0. If $d_0 = k$ in τ, the corresponding path (a good path) must start at (1, 0), pass through $(k+1, 0)$ and then $(k+1, 1)$, and finally continue to $(n-2, n-2)$. The number of ways a path can continue through $(k+1, 1)$ to $(n-2, n-2)$ is

$$N = \frac{k+1}{2n-5-k}\binom{2n-5-k}{n-2},$$

by the ballot theorem. Since there is only one way a path in P_n can get from (1, 0) to $(k+1, 0)$, N is also the number of good paths. Therefore $\text{Prob}(d_0 = k) = N/C_{n-2}$. Simplification gives (3). □

We prove Theorem 1 in two steps. For the upper bound we want to determine a k so $\text{Prob}(\Delta_n \ge k) \to 0$. Observe that $\text{Prob}(\Delta_n \ge k) = \text{Prob}(\bigcup_{i=0}^{n-1}\{d_i \ge k\})$ which, by Bonferroni's inequality [7, p. 110] is bounded by $n\,\text{Prob}(d_0 \ge k)$. Lemma 1 shows

$$\text{Prob}(\Delta_n \ge k) \le n(k+1)2^{-k}$$

which $\to 0$ for $k \ge \log n + c \log\log n$, $c > 1$.

The lower bound is

Lemma 5. *For any* $c > 0$, $\text{Prob}(\Delta_n \le k) \to 0$ *for* $k \le \log n - (1+c)\log\log n$.

Proof. This is the only tricky part, because the d_i are dependent. From Lemma 4, Δ_n is larger than the size of the largest horizontal step, s_n, in the corresponding path, so we just need to determine k so that $\text{Prob}(\Delta_n(\tau) \le k) \le \text{Prob}(s_n(p(\tau)) \le k) \to 0$.

Let $U_1, \ldots, U_{2n-4}$ be a sequence of i.i.d. uniform random variables on [0, 1]. We describe (X_m, Y_m), the coordinates of a point on a random path in P_n, after $m < 2n-4$ steps of size 1, each up or right, starting from $(1, 0) \equiv (X_0, Y_0)$. Of the C_{n-2} paths in P_n, C_{n-3} pass through (1, 1), the rest through (2, 0). Therefore, letting $I_{[A]}$ denote the indicator of A, if

$$X_1 = I_{[U_1 \le C_{n-3}/C_{n-2}]},$$

$$Y_1 = 2 - X_1,$$

(X_1, Y_1) will be (1, 1) or (2, 0) with the correct probabilities. Next suppose $(X_m, Y_m) = (i, j)$ is a point on p after $m = i + j - 1 < 2n - 5$ steps from (1, 0). By the ballot theorem there are

$$N_{i,j} = \frac{i-j+1}{2n-3-i-j}\binom{2n-3-i-j}{n-1-j}$$

continuations from (i, j) to $(n-2, n-2)$, of which $N_{i+1,j}$ go through $(i+1, j)$. Therefore the probability that the path at $(X_m, Y_m) = (i, j)$ moves right at step m is

$$p_m = \frac{N_{i+1,j}}{N_{i,j}} = \left(\frac{i+2-j}{i+1-j}\right)\left(\frac{n-2-i}{2n-4-i-j}\right), \tag{9}$$

which is 0 when $i = n-2$, and 1 when $i = j$, as required. If we define

$$X_{m+1} = X_m + I_{[U_{m+1} \le p_m]},$$

$$Y_{m+1} = Y_m + (1 - I_{[U_{m+1} \le p_m]})$$

our path will move from (i, j) to $(X_m + 1, Y_m)$ or $(X_m, Y_m + 1)$ with the correct probabilities. We use $m = i + j - 1$ in the equation for p_m and simplify to see $p_m \ge$

$$\frac{1}{2}\left(1 - \frac{1+m-2j}{2n-5-m}\right) \ge \frac{1}{2}\left(1 - \frac{1+m}{2n-5-m}\right) \ge \frac{1}{2}\left(1 - \frac{1+m^*}{2n-5-m^*}\right), \tag{10}$$

where $m^* > m$ is a bound on the number of steps taken.

Disregarding truncations we define $k = \log n - (1+c)\log\log n$, $c > 0$, $m^* = n/(2\log n)$, and

$$p = \frac{1}{2}\left(1 - \frac{1}{3\log n}\right).$$

With this choice of m^* the right-hand side of (10) is at least p, if n is large enough.

Consider the Bernoulli sequence $Z_1, Z_2, \ldots$, where $Z_i = I_{[U_i \le p]}$, $1 \le i \le m^*$, and let $L_1, L_2, \ldots$ be the lengths of its runs of consecutive ones. Each $Z_j = 0$ ends such a run and since $I_{[U_j > p_j]}$ implies $Z_j = 0$, $\Delta_n \ge \max L_i$, $i \le m^*$. Therefore

$$\begin{aligned}\text{Prob}(\Delta_n < k) &\le \text{Prob}\left(\bigcap_{i \le m^*/3} (L_i < k)\right) = [\text{Prob}(L_1 < k)]^{m^*/3} \le \left(1 - p^k\right)^{m^*/3} \\ &\le e^{-p^k m^*/3} = e^{-r(\log n)^c}\end{aligned}$$

for some constant $r > 0$. □

Proof of Lemma 2. First observe that there are $C_{k-1}C_{i-k-1}C_{n-i-1}$ triangulations in T_n containing $\triangle v_0 v_k v_i$, $i > k$. To count the number of triangulations with $\lambda_n = k$, suppose $v_0 v_k$ is a diagonal in τ (its length is k) and that v_i is the apex of its triangle. If $n - k \le i \le 2k$ neither $\|v_0 v_i\|$ nor $\|v_i v_k\|$ exceeds k. So we sum the products $C_{k-1}C_{i-k-1}C_{n-i-1}$ over i, from $n - k$ to $2k$, and multiply this sum by n to reflect the fact that the longest diagonal could as well be $v_1 v_{k+1}, v_2 v_{k+2}, \ldots, v_{n-1} v_{k-1}$. Finally the * in $\Sigma_{(*)}$ in (4) means "multiply the summand by $\frac{1}{2}$ when $i = 2k$ and when $i = n - k$ (these triangles have two edges of length k and would be counted twice), unless $3k = n$, when we multiply by $\frac{1}{3}$." This counts each good triangulation only once. □

Proof of Theorem 2. The first observation is that sum (disregarding the meaning of (*)) in (4) has the closed form

$$\sum_{i=n-k}^{2k} C_{i-k-1}C_{n-i-1} = \frac{(n-2k)(3k+1-n)}{(n-k)(n-k-1)(2n-4k-1)} \binom{2k}{k} \binom{2n-4k}{n-2k},$$

which can be verified easily. Multiply this sum by nC_{k-1}/C_{n-2}, approximate $\binom{2m}{m}$ by $4^m/\sqrt{\pi m}$, and observe that $w(x)$ is the limit as k and $n \to \infty$, with $k/n \to x \in (\frac{1}{3}, \frac{1}{2})$. The use of Σ in place of $\Sigma_{(*)}$ has no effect on this analysis. The constant α arises from direct evaluation of

$$\int_{1/3}^{1/2} x w(x)\, dx.$$

It is also possible to compute higher moments exactly. □

The proofs of the statements in Remark 2 are omitted.

4. Algorithms

There already exist algorithms for the uniform generation of elements of T_n, B_n, and P_n and which have complexity $O(n)$ in the RAM model of computation. In this section we give a new, extremely simple algorithm, based on the proof of Theorem 1, to generate elements of P_n uniformly. From a random path it is then straightforward to obtain the corresponding trangulation in T_n and tree in B_n in $O(n)$ time. Using this as a building block we can uniformly generate triangulations with maximum diagonal of a given length and triangulations with a given number of ears, both in linear time. Throughout we use "uniform" to mean "generate a uniform $[0, 1]$ random number" and "uniform$[i, i + 1, \ldots, j]$" to mean "generate an integer in $[i, j]$, each being equally likely."

4.1. *Generating Paths*

Given n, the following algorithm generates a random path from $(1, 0)$ to $(n-2, n-3)$ which is described by $j_0, \ldots, j_{n-3}$, j_i giving the width of the step made by the path at level $y = i$, and $j_0 + \cdots + j_{n-3} = n - 3$.

Algorithm 1. Rand-Path$(\mathbf{n}; \mathbf{j_0}, \ldots, \mathbf{j_{n-3}})$.

1. **(initialize:)** $\mathbf{m \leftarrow 0}$; $\mathbf{(X_m, Y_m) \leftarrow (1, 0)}$; $\mathbf{i \leftarrow 0}$; $\mathbf{j_i \leftarrow 0}$
2. $\mathbf{p_{m+1} \leftarrow [(X_m + 2 - Y_m)(n - 2 - X_m)]/[(X_m - 1 - Y_m)(2n - 4 - X_m - Y_m)]}$
3. $\mathbf{U_{m+1} \leftarrow}$ **uniform;** $\mathbf{X_{m+1} \leftarrow X_m + I_{[U_{m+1} \le p_{m+1}]}}$
   $\mathbf{Y_{m+1} \leftarrow Y_m + (1 - I_{[U_{m+1} \le p_{m+1}]})}$
4. **IF** $\mathbf{Y_{m+1} > Y_m}$ **THEN** $\mathbf{(i \leftarrow i + 1;\ j_i \leftarrow 0)}$ **ELSE** $\mathbf{j_i \leftarrow j_i + 1}$
5. $\mathbf{m \leftarrow m + 1}$; **IF** $\mathbf{m < 2n - 5}$, $\mathbf{\rightarrow 2}$.

The algorithm is correct. Step 2, given that $(X_m, Y_m) = (i, j)$, computes p_{m+1} as in (9). Therefore (X_{m+1}, Y_{m+1}) is either $(X_m + 1, Y_m)$ or $(X_m, Y_m + 1)$, each with the correct probability. By the remark immediately following (9), this gives a path in P_n. By induction each path has equal probability.

The algorithm is $O(n)$. Since each p_m is computed in constant time, the $2n - 5$ steps take linear time. Since j_i is the outdegree of vertex v_i in the triangulation τ that corresponds to the path just generated, it is straightforward now to obtain τ in linear time, described, for example, by the list of its $n-3$ diagonals. We refer to the process of generating a path and then obtaining its corresponding triangulation by **Rand-Tri(**$v_0, \ldots, v_{n-1}$**)**.

Algorithm 1 is similar to the method of Arnold and Sleep [1] that is mentioned in [5]. A different approach for generating triangulations, paths, or trees appears in [2].

Let $k \ge (n-3)/2$. Suppose we take $m = 0$, $(X_0, Y_0) = (k+1, 1)$, $j_0 \leftarrow k$, $i \leftarrow 1$, and begin the path algorithm with Step 2, terminating when the path reaches $(n - 2, n - 3)$. The generated path corresponds to a random triangulation with $\Delta_n = k$ and whose max-degree vertex is v_0. To randomize the max-degree vertex we choose integer I uniformly in $[0, n - 1]$ and **Rotate(**I**)**, where **Rotate(**j**)** means "change the vertex labeling of K so $v_i \rightarrow v_{(i+j) \bmod n}$. We now have a random triangulation with max-degree k, each one being equally likely. It is not clear how to do this efficiently for smaller k.

4.2. *Triangulations with Fixed* λ

Let $T_n(k) \subset T_n$ denote the triangulations whose longest diagonal has length k, $n/3 \le k \le n/2$. The proofs of Lemma 2 and Theorem 2 suggest an approach for fast uniform generation of triangulations in $T_n(k)$. Suppose that $v_0 v_k$ is the diagonal of max length ($= k$) in the desired triangulation. We generate the apex v_i of its triangle, with $i \in [n-k, 2k]$ chosen according to the correct probability. The counting argument used for Lemma 2 shows that v_i should be chosen with probability $C_{i-k-1} C_{n-i-1} / S_{2k}$, where $S_m = \sum_{j=n-k}^{m} C_{j-k-1} C_{n-j-1}$. Finally, using Rand-Tri we randomly triangulate the polygons defined by $(v_0, \ldots, v_k)$, $(v_k, \ldots, v_i)$, and $(v_i, \ldots, v_0)$, and then rotate K so diagonal $v_j v_{j+k}$ has probability $1/n$ to be the longest, $j = 0, \ldots, n - 1$.

Algorithm 2. Random-max-diag$(\mathbf{k}; \mathbf{v_0}, \ldots, \mathbf{v_{n-1}})$.

1. $\mathbf{U \leftarrow}$ **uniform**
2. $\mathbf{i} \leftarrow \min(\mathbf{j} : \mathbf{S_j} \ge \mathbf{U} * \mathbf{S_{2k}})$
3. **Rand-Tri**$(\mathbf{v_0}, \ldots, \mathbf{v_k})$; **Rand-Tri**$(\mathbf{v_k}, \ldots, \mathbf{v_i})$; **Rand-Tri**$(\mathbf{v_i}, \ldots, \mathbf{v_0})$.
4. $\mathbf{I \leftarrow}$ **uniform**$(\mathbf{0}, \ldots, \mathbf{n - 1})$
5. **Rotate**$(\mathbf{I})$

4.3. *Triangulations with k Ears*

For $k \in [2, \lfloor n/2 \rfloor]$ let $T_n^k \subset T_n$ denote the triangulations of K with exactly k ears. We give an $O(n)$ algorithm to generate these triangulations uniformly. The algorithm is based on a combinatorial proof of the following formula for the number of k-ear triangulations (see [9]):

$$|T_n^k| = \frac{n}{k} 2^{n-2k} \binom{n-4}{n-2k} C_{k-2}.$$

Let τ be a triangulation of K having ears at vertices $v_{j_1}, \ldots, v_{j_k}$, and fix $j_1 = 0$. Obviously $|j_i - j_{i+1}| \geq 2$. τ has $n-3$ diagonals *including* $v_{j_i-1}v_{j_i+1}$, $i = 1 \ldots, k$. We *collapse* τ by removing (in any order) every edge of K that is not incident to an ear of τ, $n-2k$ edges in all. When edge $v_r v_{r+1}$ is removed from, say, $v_r v_{r+1} v_q$, we identify v_{r+1} with v_r and note that the two diagonals $v_q v_r$ and $v_q v_{r+1}$ become one, so $n-2k$ of the diagonals of τ have also been removed, leaving $2k-3$.

Let K' be the resulting collapsed polygon and τ' its triangulation. Since K' is a $2k$-gon and τ' has k ears, there are C_{k-2} different possibilities for τ'; k of its diagonals (one for each ear) form a convex k-gon whose interior has C_{k-2} distinct triangulations. To count the number of triangulations τ that collapse to the same triangulation of K', order the $n-3$ diagonals of τ, for example, so $v_i v_j$ precedes $v_i v_r$ for diagonals where $i < j < r$ and $v_i v_j$ precedes $v_r v_s$ for diagonals where $i < r$ except $d = v_0 v_{n-1}$ is always last. d remains in τ' but $n-2k$ of the other $n-4$ are eliminated when τ collapses to τ' (see Fig. 3). There are $\binom{n-4}{n-2k}$ choices for which diagonals are eliminated, each of which corresponds to a triangulation that collapses to τ'. Finally, suppose the diagonal

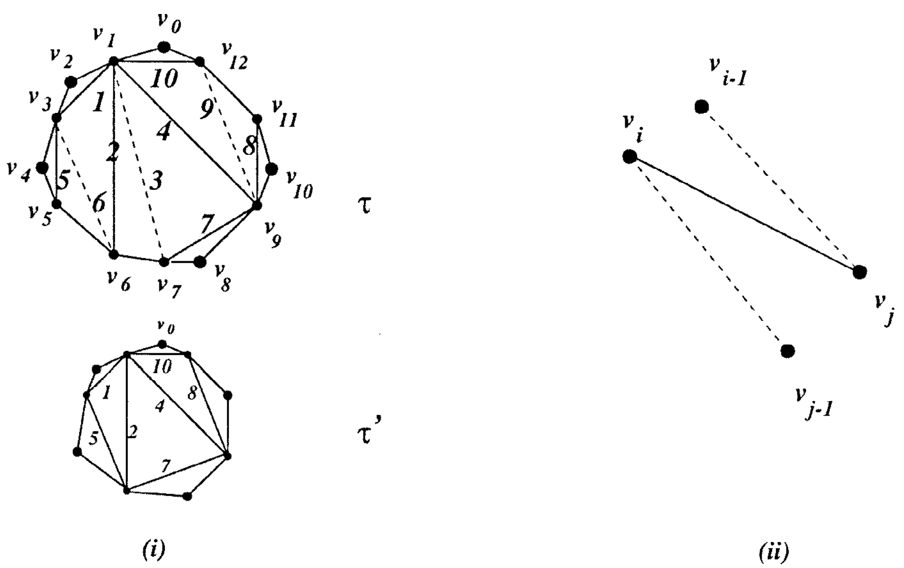

Fig. 3. (i) An example of collapse with n=13 and k=5. The dotted diagonals of τ disappear in the collapse. (ii) Two distinct ways the predecessor of $v_i v_j$ could be removed in collapse, or precede $v_i v_j$ in insertion.

immediately preceding $v_i v_j$ in τ is eliminated in the collapsing. There are two distinct ways this can arise for each eliminated diagonal (see Fig. 3), and thus, $2^{n-2k}\binom{n-4}{n-2k}$ distinct triangulations τ collapse to each triangulation τ' of K'. Finally the term n/k comes from the fact that v_0 was made to be an ear; every one of the n vertices could play this role but then each triangulation would be counted k times, once for each ear.

This argument underlies the following simple linear-time algorithm to generate a k ear triangulation.

Algorithm 3. Rand-ears$(\mathbf{k}; \mathbf{v_0}, \ldots, \mathbf{v_{n-1}})$.

1. **Rand-Tri$(\mathbf{v_0}, \ldots, \mathbf{v_{k-1}})$**
2. $\tau' \leftarrow$ **Add k ears**
3. **S** $\leftarrow$ **uniform**$\binom{(1,\ldots,n-4)}{2k-4}$; **label** τ' **with S**
4. $\tau \leftarrow$ **insert diagonals from** $\mathbf{S^c}$
5. **I** $\leftarrow$ **uniform**$(\mathbf{0}, \ldots, \mathbf{n-1})$
6. **Rotate(I)**

To sketch some details, Step 1 randomly triangulates a k-gon. In Step 2 renumber $v_i \to v_{2i+1}$, $i = 1, \ldots, k$, add vertices v_{2i}, $i = 1, \ldots, k$, and diagonals $v_{2i-1}v_{2i+1}$, $i = 1, \ldots, k$; this puts ears at the k even number vertices of τ', the other diagonals being random. In Step 3 diagonal $v_1 v_{2k-1}$ is labeled $n-3$, the others are labeled by $1 \leq j_1 < \cdots < j_{2k-4} \leq n-4$, the $2k-4$ elements in S, randomly chosen from $1, \ldots, n-4$. Step 4 is done by *merging* S^c into $S \cup (n-3)$. If the current element $r \in S^c$ is larger than the current element $s \in S$, we advance to the next element of S. Otherwise an edge is *inserted* into τ' immediately preceding $s = v_i v_j$, $i < j$. Specifically, if $U \leftarrow$ *uniform* is less than 0.5 we create a new vertex between v_{i-1} and v_i and the corresponding diagonal to v_j; otherwise a new vertex appears between v_{j-1} and v_j and the corresponding diagonal from v_i (see Fig. 3). The details are easily managed in $O(n)$.

5. Final Remarks

This paper studied the behavior of two properties of a random triangulation of a convex n-gon: (1) Δ_n, the maximal degree; (2) λ_n, the length of the longest diagonal. The functions Δ_n and λ_n correspond to interesting features of binary trees and our results on triangulations give new information about random trees. Some other points are:

1. Following the idea in Remark 1, as $n \to \infty$,

$$\text{Prob}(d_i = k) \to (k+1)2^{-(k+2)},$$

   the distribution of G_2, the sum of two independent geometric $(\frac{1}{2})$ random variables. It is interesting to wonder whether Δ_n converges in distribution to the limit distribution of the maximum of n independent copies of G_2.
2. According to Remark 2, $E(d_i) = 2(1 - 3/n)$ for uniform triangulations but $E_\beta(d_0) = E_\beta(d_{n-1}) = \Theta(\log n)$ in search-tree probability, a real difference between the two distributions. At the same time, however, $E(\Delta_n/\log n) \to 1$ in

the uniform case and to $c > 1$ in the other. For the purpose of comparison, some known properties of random binary trees in the two distributions are:

(a) *Height* h_n, the maximal depth of a node. $E(h_n)$ is asymptotic to $2\sqrt{\pi n}$ in the uniform case [8] and concentrated about $4.31107 \ln n$ in the other [4].

(b) *Leaves*. In the limit one expects $n/4$ leaves for uniform trees and $n/3$ for binary search trees (see [14]).

3. We note that it is easy to generate random triangulations in the search-tree probability. When constructing the triangulation as in the opening paragraph, just choose v_i uniformly from $1, \ldots, n-2$, etc. The complexity is $O(n)$.
4. An outstanding question concerns the set $T(K)$ of triangulations of a set K of n points not necessarily in convex position. As opposed to (0), it is not known how to count or even to approximate $|T(K)|$.

Acknowledgments

We thank Imre Bárány, Alistair Sinclair, and Jean-Marc Fédou for valuable conversations.

References

1. D. Arnold and M. Sleep. Uniform Random Number Generation of n Balanced Parenthesis Strings. *ACM Trans. Prog. Languages Systems* **2**, 122–128, 1980.
2. M. Atkinson and J. Sack. Generating Binary Trees at Random. *Inform. Process. Lett.* **41**, 21–23, 1992.
3. N. Dershowitz and S. Zaks. Patterns in Trees. *Discrete Appl. Math.* **25**, 241–255, 1989.
4. L. Devroye. A Note on the Height of Binary Search Trees. *J. Assoc. Comput. Mach.* **33**, 489–498, 1986.
5. L. Devroye. *Non-Uniform Random Variate Generation*, Springer-Verlag, New York, 1986.
6. L. Devroye. Application of the Theory of Records in the Study of Random Trees. *Acta Inform.* **26**, 123–130, 1988.
7. W. Feller. *An Introduction to the Theory of Probability and Its Applications*, Vol. 1, 3rd Edn., Wiley, New York, 1968.
8. P. Flajolet and A. Odlyzko. The Average Height of Binary Trees and Other Simple Trees. *J. Comput. System. Sci.* **25**, 171–213, 1982.
9. F. Hurtado and M. Noy. Ears of triangulations and Catalan Numbers. *Discrete Math.* **149**, 319–324, 1996.
10. F. Hurtado and M. Noy. The Graph of Triangulations of a Convex Polygon. *Proc. Twelfth ACM Symposium on Computational Geometry*, N C7–8, 1996.
11. H. Mahmoud. *Evolution of Random Search Trees*. Wiley, New York, 1992.
12. A. Meir and J. Moon. On the Altitude of Nodes in Random Trees. *Canad. J. Math.* **30**, 997–1015, 1978.
13. B. Pittel. On Growing Random Binary Trees. *J. Math. Anal. Appl.* **103**, 461–480, 1984.
14. R. Sedgewick and P. Flajolet *An Introduction to the Analysis of Algorithms*. Addison-Wesley, Reading, MA, 1996.
15. D. Sleator, R. Tarjan, and W. Thurston. Rotation Distance, Triangulations, and Hyperbolic Geometry. *J. Amer. Math. Soc.* **1**, 647–681, 1988.

Received August 18, 1997, *and in revised form November* 5, 1997.

Note added in proof. Gao and Wormald (preprint) recently proved the conjecture in point 1 of the Final Remarks (Section 5) and sharpened the statements in Theorem 1.