Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 379 (2007) 377–386
Theoretical Computer Science
www.elsevier.com/locate/tcs

# Generating labeled planar graphs uniformly at random

Manuel Bodirsky[a], Clemens Gröpl[b,*], Mihyun Kang[a]

[a] Humboldt-Universität zu Berlin, Institut für Informatik, Abteilung Algorithmen und Komplexität I, Unter den Linden 6, 10099 Berlin, Germany
[b] Freie Universität Berlin, Institut für Informatik, Algorithmische Bioinformatik, Takustraße 9, 14195 Berlin, Germany

**Abstract**

We present a deterministic polynomial time algorithm to sample a labeled planar graph uniformly at random. Our approach uses recursive formulae for the exact number of labeled planar graphs with $n$ vertices and $m$ edges, based on a decomposition into 1-, 2-, and 3-connected components. We can then use known sampling algorithms and counting formulae for 3-connected planar graphs.


*Keywords:* Labeled planar graph; Enumeration; Decomposition; Sampling algorithm; Dynamic programming; Graph theory

## 1. Introduction

A *planar graph* is a graph which can be embedded in the plane, as opposed to a *map*, which is an embedded graph. There is a rich literature on the enumerative combinatorics of maps, starting with Tutte's census papers, e.g. [27]. An efficient random sampling algorithm was developed by Schaeffer [24]. Little is known about random planar graphs, although they recently attracted much attention [2,4,6,8,14,19,22]. If we had an efficient algorithm to sample a planar graph uniformly at random, we could experimentally verify conjectures about properties of random planar graphs. We could also use it to evaluate the average-case running times of algorithms on planar graphs. Denise, Vasconcellos, and Welsh [8] introduced a Markov chain whose stationary distribution is the uniform distribution on all labeled planar graphs. However, its mixing time is unknown and seems hard to analyze, and is perhaps not polynomial. Moreover, the corresponding sampling algorithm only approximates the uniform distribution.

We obtain the first deterministic polynomial time algorithm for generating a labeled planar graph uniformly at random.

* Corresponding address: Institut fur Informatik, Freie Universität Berlin, Takustrasse 9, Room number: 011, 14195 Berlin, Germany. Tel.: +49 30 838 75 229; fax: +49 30 838 75 218.

*E-mail addresses:* bodirsky@informatik.hu-berlin.de (M. Bodirsky), groepl@inf.fu-berlin.de (C. Gröpl), kang@informatik.hu-berlin.de (M. Kang).

*URLs:* http://www2.informatik.hu-berlin.de/~bodirsky/ (M. Bodirsky), http://www.inf.fu-berlin.de/~groepl/ (C. Gröpl), http://www2.informatik.hu-berlin.de/~kang/ (M. Kang).

0304-3975/$ - see front matter 
doi:10.1016/j.tcs.2007.02.045

**Theorem 1.** *Labeled planar graphs on n vertices and m edges can be sampled uniformly at random in deterministic time* $\tilde{O}(n^7)$ *and space* $O(n^4 \log n)$*. If we apply a preprocessing step, this can also be done in deterministic time* $\tilde{O}(n^3)$ *and space* $O(n^7)$.

Our result uses known graph decomposition and counting techniques [28,31] to reduce the counting and random sampling of labeled planar graphs to the counting and random sampling of 3-connected *rooted planar maps*. Usually, a planar graph has many embeddings that are non-isomorphic like maps, but some graphs have a unique embedding. A classical theorem of Whitney (see e.g. [10]) asserts that 3-connected planar graphs are *rigid* in the sense that all embeddings in the sphere are combinatorially equivalent. As *rooting* destroys any further symmetries; they are closely related to 3-connected *labeled* planar graphs. Moreover, the 'degrees of freedom' of the embedding of a planar graph are governed by its connectivity structure. We exploit this fact by composing a planar graph out of 1-, 2-, and 3-connected components.

Our sampling procedure first determines the number of components, and how many vertices and edges they shall contain. Each connected component is generated independently from the others, but has the chosen numbers of vertices and edges. To generate a connected component with given numbers of vertices and edges, we decide on a decomposition into 2-connected subgraphs, and how the vertices and edges shall be distributed among its parts. So far, this approach is similar to the one used in [5], where the goal was to sample random outerplanar graphs. In the planar case, we need to go one step further.

Trakhtenbrot [26] showed that every 2-connected graph is uniquely composed of special graphs (called *networks*) of three kinds. Such networks can be combined in series, in parallel, or using a 3-connected graph as a core (see Theorem 2 below). Using this composition, we can then employ known results about counting and the random sampling of 3-connected planar maps.

The concept of rooting plays an important role in the enumeration of planar maps. A *face-rooted* map is one with a distinguished edge which lies on the outer face, to which a direction is assigned. Rooting forces isomorphisms to: map the outer face to the outer face, keep the root edge incident to the outer face, and preserve its direction. The enumeration of 3-connected face-rooted unlabeled maps with given numbers of vertices and faces – also called *c-nets* – was achieved by Mullin and Schellenberg [20]. We invoke their closed formulae in order to count 3-connected labeled planar graphs with given numbers of vertices and edges. For the generation of 3-connected labeled planar graphs with given numbers of vertices and edges, we employ a recent deterministic polynomial time algorithm [3]. Alternatively, we can use a sampling procedure that runs in linear time that was recently presented in [13]; in this case we obtain an *expected* polynomial time sample for labeled planar graphs, which runs in time $O(n^3)$ and space $\tilde{O}(n^6)$ after a preprocessing step in time $\tilde{O}(n^6)$, or in time $\tilde{O}(n^6)$ and space $O(n^4 \log n)$ without preprocessing.

When we apply the various counting sampling subroutines along the stages of the connectivity decomposition, we must branch with the correct probabilities. To compute those probabilities,we use recurrence formulae that can be evaluated in polynomial time using dynamic programming. Then the decomposition can be translated immediately into a sampling procedure.

The paper is organized as follows. In the next section we give the graph theoretic background for the decomposition of planar graphs, which guides us when we derive counting formulae for planar graphs in the following three sections. In Section 7 we analyze the running time and memory requirements of the corresponding sampling procedures, and discuss results from an implementation of the counting part. We conclude with a discussion of variations of the approach.

## 2. Decomposition by connectivity

Let us recall and fix some terminology [10,28–30]. A *graph* will be assumed to be unoriented and *simple*, i.e., having no loops or multiple (also called *parallel*) edges; if multiple edges are allowed, the term *multigraph* will be used. We consider labeled graphs whose vertex sets are initial segments of $\mathbb{N}_0$.

Every connected graph can be decomposed into *blocks* by being split at cutvertices. Here a block is a maximal subgraph that is either 2-connected, or a pair of adjacent vertices, or an isolated vertex. The *block structure* of a graph $G$ is a tree whose vertices are the cutvertices of $G$ and the blocks of $G$ (considered as vertices), where adjacency is defined by containment. Conversely, we will *compose* connected graphs by identifying the vertex 0 of one part with an arbitrary vertex of the other. A formal definition of composition operations is given at the end of this section.

A *network* $N$ is a multigraph with two distinguished vertices 0 and 1, called its *poles*, such that the multigraph $N^*$ obtained from $N$ by adding an edge between its poles is 2-connected. The new edge is not considered a part of the network $N$. We can replace an edge $uv$ of a network $M$ with another network $X_{uv}$ by identifying $u$ and $v$ with the poles 0 and 1 of $X_{uv}$, and iterate the process for all edges of $M$. Then the resulting graph $G$ is said to have a *decomposition* with *core* $M$ and *components* $X_e$, $e \in E(M)$.

Every network can be decomposed into (or composed out of) networks of three special types. A *chain* is a network consisting of two or more edges connected in *series* with the poles as its terminal vertices. A *bond* is a network consisting of two or more edges connected in *parallel*. A *pseudo-brick* is a network $N$ with no edge between its poles such that $N^*$ is 3-connected. (3-connected subgraphs are sometimes called bricks.) A network $N$ is called an *h-network* (respectively, a *p-network*, or an *s-network*) if it has a decomposition whose core is a pseudo-brick (respectively, a bond, or a chain). Trakhtenbrot [26] (here cited from [30]) has formulated a canonical decomposition theorem for networks.

**Theorem 2** (*Trakhtenbrot*). *Any network with at least* 2 *edges belongs to exactly one of the* 3 *classes: it is either an h-network, p-network, or s-network. An h-network has a unique decomposition and a p-network (respectively, an s-network) can be uniquely decomposed into components which are not themselves p-networks (s-networks), where the uniqueness is up to the orientation of the edges of the core, and also up to their order if the core is a bond.*

A network $N$ is *simple* if $N^*$ is a simple graph. Let $N(n, m)$ be the number of simple planar networks on $n$ vertices and $m$ edges. In view of Theorem 2, we introduce the functions $H(n, m)$, $P(n, m)$, and $S(n, m)$ that count the number of simple planar h-, p-, and s-networks on $n$ vertices and $m$ edges. Note that the components of simple networks are simple networks (or just edges). For example, $K_3$ (the complete graph on three vertices) is a (non-simple) p-network composed of an edge and a path of length two, which in turn is a simple s-network composed of two edges. The graph $K_4 - \{0, 1\}$ is a simple h-network, and all its components are simple edges.

Let $G^{(c)}(n, m)$ denote the number of $c$-connected planar graphs with $n$ vertices and $m$ edges. For $c = 0, 1, 2$ let us define *compose operations* for the three stages of the connectivity decomposition. Informally, for $c = 0$ the composition equals the disjoint union. For $c = 1$, we join the parts at a single vertex. For $c = 2$ we replace one edge of the first part by the second part. A formal definition is as follows: Assume that $M$ and $X$ are graphs on the vertex sets $[0 .. k-1]$ and $[0 .. i-1]$, and we want to compose them by identifying the vertices $j$ of $X$ with the vertices $v_j$ of $M$, for $j = 0, \dots, c-1$, such that the resulting graph will have $n := k+i-c$ vertices. (No vertices are identified for $c = 0$.) Moreover, let $S$ be a set of $i - c$ vertices from $[c \dots n-1]$ which are designated for the remaining part of $X$. Let $M'$ be the graph obtained by mapping the vertices of $M$ to the set $[0 \dots n-1] \setminus S$, retaining their relative order. Let $X'$ be the graph obtained by mapping the vertices $[c \dots i-1]$ of $X$ to the set $S$, retaining their relative order, and mapping $j$ to the image of $v_j$ in $M'$ for $j = 0, \dots, c-1$. Then the result of the compose operation for the arguments $M$, $(v_0, \dots, v_{c-1})$, $X$, and $S$ is the graph with the vertex set $[0 \dots n-1]$ and edge set $E(M') \cup E(X')$. If $c = 2$ and $M$ contains an edge $\{v_0, v_1\}$, it is deleted.

## 3. Planar graphs

We show how to count and generate labeled planar graphs with a given number of vertices and edges in three steps. A first simple recursive formula reduces the problem to the case of connected graphs. In the next section, we will use the block structure to reduce the problem to the 2-connected case. This may serve as an introduction to the method before we go into the more involved arguments of Section 5.

Let $F_k(n, m)$ denote the number of planar graphs with $n$ vertices and $m$ edges having $k$ connected components. Clearly, $F_1(n, m) = G^{(1)}(n, m)$ and $G^{(0)}(n, m) = \sum_{k=1}^{n} F_k(n, m)$. Moreover,

$$F_k(n, m) = 0 \quad \text{for } m + k < n .$$

We count $F_k(n, m)$ by induction on $k$. Every graph with $k \geq 2$ connected components can be decomposed into the connected component containing the vertex 0 and the remaining part, using the inverse of the compose operation for $c = 0$ as defined in Section 2. If the split-off part has $i$ vertices, then there are $\binom{n-1}{i-1}$ ways to choose its vertex set, as the vertex 0 is always contained in it. The remaining part has $k - 1$ connected components. We obtain the recursive formula

$$F_k(n, m) = \sum_{i=1}^{n-1} \sum_{j=0}^{m} \binom{n-1}{i-1} G^{(1)}(i, j) F_{k-1}(n-i, m-j) \quad \text{for } k \geq 2 .$$

Thus it suffices to count connected graphs. The counting recurrence also has an analogue for generation: assume that we want to generate a planar graph $G$ with $n$ vertices and $m$ edges uniformly at random. First, we choose $k \in [1 .. n]$ with a probability proportional to $F_k(n, m)$. Then we choose the number of vertices $i$ of the component containing the vertex 0, and its number of edges $j$, with a joint probability proportional to $\binom{n-1}{i-1} G^{(1)}(i, j) F_{k-1}(n - i, m - j)$. We also pick an $(i - 1)$-element subset $S' \subseteq [1 .. n - 1]$ uniformly at random and the set $S := S' \cup \{0\}$. Then we compose $G$ (as explained in Section 2) out of a random connected planar graph with parameters $i$ and $j$, which is being mapped to the vertex set $S$, and a random planar graph with parameters $n - i$ and $m - j$ having $k - 1$ connected components, which is generated in the same manner.

## 4. Connected planar graphs

In this section, we reduce the counting and generation of connected labeled planar graphs to the 2-connected case. Let $M_d(n, m)$ denote the number of connected labeled planar graphs in which the vertex 0 is contained in $d$ blocks. Here we will call them *$m_d$-planars*. An $m_1$-planar is a connected planar graph in which 0 is not a cutvertex. Clearly, $G^{(1)}(n, m) = \sum_{d=1}^{n-1} M_d(n, m)$ and

$$M_d(n, m) = 0 \quad \text{for } n < d \text{ or } m < d.$$

In order to count $m_d$-planars by induction on $d$ (for $d \geq 2$), we split off the largest connected subgraph containing the vertex 1 in which 0 is not a cutvertex. This is done by performing the inverse of the compose operation for $c = 1$ as defined in Section 2. If the split off $m_1$-planar has $i$ vertices, then there are $\binom{n-2}{i-2}$ possible choices for its vertex set, as the vertices 0 and 1 are always contained in it. The remaining part is an $m_{d-1}$-planar. Thus

$$M_d(n, m) = \sum_{i=2}^{n-d+1} \sum_{j=1}^{m-1} \binom{n-2}{i-2} M_1(i, j) M_{d-1}(n - i + 1, m - j) \quad \text{for } d \geq 2,$$

and this immediately translates into a generation procedure.

Next we consider $m_1$-planars. The *root block* is the unique block containing the vertex 0. A recurrence for $m_1$-planars arises from splitting off the subgraphs attached to the root block at its cutvertices one at a time. Thus we consider $m_1$-planars such that the $b$ least labeled vertices in the root block are not cutvertices. Let us call them *$l_b$-planars*, and denote the number of $l_b$-planars with $n$ vertices and $m$ edges by $L_b(n, m)$. The initial cases ($b = n$) of the recurrence are connected graphs without cutvertices. We have

$$L_n(n, m) = \begin{cases} G^{(2)}(n, m) & \text{for } n \geq 3 \\ 1 & \text{for } n \in \{1, 2\} \text{ and } m = n - 1 . \end{cases}$$

We calculate $L_b(n, m)$ for $b = n - 1, \ldots, 1$, and eventually $M_1(n, m) = L_1(n, m)$ recursively, as follows. To count $L_b$ using $L_{b+1}$, we split off the subgraph attached to the $b$-th-least labeled vertex in the root block, if it is a cutvertex. This can be any connected planar graph. The remaining part is $l_{b+1}$-planar. If the split off subgraph has $i$ vertices, then there are $\binom{n-1}{i-1}$ ways to choose them, as the vertex 0 of the subgraph will be replaced with the cutvertex. We obtain the recursive formula

$$L_b(n, m) = \sum_{i=1}^{n-1} \sum_{j=0}^{m-1} \binom{n-1}{i-1} G^{(1)}(i, j) L_{b+1}(n - i + 1, m - j) \quad \text{for } m \geq b \geq 1 .$$

The values $G^{(1)}(i, j)$ are known since $i < n$, $j < m$. Again, the generation procedure is straightforward.

## 5. Two-connected planar graphs

In this section, we show how to count and generate 2-connected planar graphs. If we take an arbitrary simple planar network with $n$ vertices and $m - 1$ edges, add an edge between the poles, then choose a pair $0 \leq x < y \leq n - 1$, and exchange the vertex labels (0, 1) with $(x, y)$, then we obtain every 2-connected labeled planar graph with $n$ vertices and $m$ edges in $m$ ways. Thus

$$G^{(2)}(n,m) \;=\; \begin{cases} \dfrac{\binom{n}{2}}{m} N(n, m-1) & \text{for } n \geq 3\,, m \geq 3 \\ 0 & \text{otherwise.} \end{cases}$$

Now we derive recurrence formulas for the number $N$ of simple planar networks. Trakhtenbrot's decomposition theorem implies

$$N(n,m) \;=\; \begin{cases} P(n,m) + S(n,m) + H(n,m) & \text{for } n \geq 3\,, m \geq 2 \\ 0 & \text{otherwise}\,. \end{cases}$$

**p-Networks.** Let us call a p-network with a core consisting of $k$ parallel edges a *$p_k$-network*, and let $P_k(n, m)$ be the number of $p_k$-networks having $n$ vertices and $m$ edges. Clearly, $P(n, m) = \sum_{k=2}^{m} P_k(n, m)$. In order to count $p_k$-networks by induction on $k$, we split off the component containing the vertex labeled 2 by performing the inverse of the compose operation for $c = 2$ as defined in Section 2. Technically, it is convenient to consider the split off component as a *$p_1$-network*. But note that according to the canonical decomposition, a $p_1$-network is either an h- or an s-network. Assume that it has $i$ vertices and $j$ edges. Then

$$P_1(i,j) \;=\; \begin{cases} H(i,j) + S(i,j) & \text{for } i \geq 3\,, j \geq 2 \\ 0 & \text{otherwise}\,. \end{cases}$$

The remaining part is a $p_{k-1}$-network (even if $k = 2$). For $k \geq 2$ we have

$$P_k(n,m) \;=\; 0 \quad \text{if } n \leq 2 \text{ or } m < k.$$

There are $\binom{n-3}{i-3}$ ways in which the vertex labels $[0 \ldots n-1]$ can be distributed among both sides, as the labels 0, 1, and 2 are fixed. We obtain the recurrence formula

$$P_k(n,m) \;=\; \sum_{i=3}^{n-1}\sum_{j=2}^{m-1} \binom{n-3}{i-3} P_1(i,j) P_{k-1}(n-i+2, m-j) \quad \text{for } k \geq 2\,.$$

**s-Networks.** Let us call an s-network whose core is a path of $k$ edges an *$s_k$-network*, and denote the number of $s_k$-networks which have $n$ vertices and $m$ edges by $S_k(n, m)$. Then $S(n, m) = \sum_{k=2}^{m} S_k(n, m)$. We use induction on $k$ again, but for $s_k$-networks, we split-off the component containing the vertex labeled 0. Again, it can be considered as an $s_1$-network, and it is either an h- or a p-network, according to the canonical decomposition. Thus

$$S_1(i,j) \;=\; \begin{cases} H(i,j) + P(i,j) & \text{for } i \geq 3,\, j \geq 2 \\ 1 & \text{for } i = 2,\, j = 1 \\ 0 & \text{otherwise}\,. \end{cases}$$

The remaining part is an $s_{k-1}$-network (even if $k = 2$). For $k \geq 2$, we have

$$S_k(n,m) \;=\; 0 \quad \text{if } n < k+1 \text{ or } m < k.$$

Concerning the number of ways in which the labels can be distributed among both parts, note that the labels 0 and 1 are fixed; hence the new 0-root for the remaining part can be one out of $n - 2$ vertices, and so the number of choices for the internal vertices of the split off $s_1$-network is $\binom{n-3}{i-2}$. We obtain the recurrence formula

$$S_k(n,m) \;=\; (n-2)\sum_{i=2}^{n-1}\sum_{j=1}^{m-1} \binom{n-3}{i-2} S_1(i,j) S_{k-1}(n-i+1, m-j) \quad \text{for } k \geq 2\,.$$

**h-Networks.** The core of an h-network is a pseudo-brick. We can order the edges of the core lexicographically using the vertex numbers. A recurrence formula similar to the p- and s-network cases arises from replacing the edges of the core with components one at a time and in lexicographic order. In order to give names to the intermediate stages, let us call an h-network where the components corresponding to the first $k$ edges of the core are simple edges an *$h_k$-network*, and denote the number of $h_k$-networks with $n$ vertices and $m$ edges by $H_k(n, m)$. For $k \geq m$, all components must be simple edges. $H_m(n, m)$ is the number of pseudo-bricks with $n$ vertices and $m$ edges, the initial case of our recursion. We have

$$H_m(n, m) = \frac{(n-2)!}{2} Q(n, m+1),$$

where $Q(n, m)$ denotes the number of c-nets, i.e., rooted 3-connected simple maps, with $n$ vertices and $m$ edges (see the next section). If we take an arbitrary c-net, assign the labels 0 and 1 to the root vertex and the other vertex of the root edge, delete the root edge, and then number the remaining vertices arbitrarily, we obtain each pseudo-brick in two ways (namely, one for each face routing).

Next we derive a recurrence formula to calculate $H_k(n, m)$ for $k = m-1, \ldots, 0$, and eventually $H(n, m) = H_0(n, m)$. To count $H_k$ using $H_{k+1}$, we split off the $k$-th component of an $h_k$-network, i.e., the component replacing the $k$-th edge of the core. This can be a simple network of any of the three kinds, or such a simple network together with an edge between its poles. Assume that it has $i$ vertices and $j$ edges. Then the number of choices for the component network is

$$H'(i, j) = \begin{cases} N(i, j) + N(i, j-1) & \text{for } i \geq 3, j \geq 2 \\ 1 & \text{for } i = 2, j = 1 \\ 0 & \text{otherwise}. \end{cases}$$

The remaining part is an $h_{k+1}$-network. There are $\binom{n-2}{i-2}$ ways to choose the vertices of the component, as the vertices 0 and 1 are merged with the endpoints of the $k$-th edge of the core, respecting their relative order. We obtain the recurrence formula

$$H_k(n, m) = \sum_{i=2}^{n-2} \sum_{j=1}^{m-k+1} \binom{n-2}{i-2} H'(i, j) H_{k+1}(n-i+2, m-j+1) \quad \text{for } m > k \geq 0.$$

## 6. c-Nets

In the preceding sections, we have shown how to count and sample random planar graphs assuming that we can do so for *c-nets*, i.e., 3-connected simple rooted planar maps. For this we use a formula for their number $Q(n, m)$ derived by Mullin and Schellenberg in [20]. Using Euler's formula, it asserts that

$$Q(n, m) = 0 \quad \text{for } n < 4 \text{ or } m < n+2$$

and otherwise

$$Q(n, m) = -\sum_{i=2}^{n} \sum_{j=n}^{m} (-1)^{i+j-n} \binom{i+j-n}{i} \binom{i}{2} \times \left[ \binom{2m-2n+2}{n-i} \binom{2n-2}{m-j} - 4 \binom{2m-2n+1}{n-i-1} \binom{2n-3}{m-j-1} \right].$$

This concludes the counting task.

The first sampling algorithm for c-nets with given numbers of vertices running in *expected* polynomial time is due to Schaeffer et al. [1,23,24]. For our sampling algorithm, we also need to control the number of edges. A sampling procedure with this additional requirement has been described in [13]. It runs in expected time $O(n^2)$ for a fixed edge density ratio $\alpha \in ]\frac{3}{2}, 3[$ , where $\frac{m}{n} \to \alpha$, and in expected time $O(n^3)$ for triangulations (where $\frac{m}{n} \to 3$), which is also the worst case [13].

For a *deterministic* polynomial running time, we use an extended version of the algorithm presented in [3] with an additional parameter for the number of edges, as explained in the conclusion of [3]. The resulting algorithm runs in deterministic $\tilde{O}(n^7)$ time and $O(n^4)$ space, or, if a pre-computation is allowed, $\tilde{O}(n^3)$ time and $O(n^7)$ space.

| $G^{(0)}$ | $m$=0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | $m$=11 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $n$=2 | 1 | 1 | | | | | | | | | | |
| 3 | 1 | 3 | 3 | 1 | | | | | | | | |
| 4 | 1 | 6 | 15 | 20 | 15 | 6 | 1 | | | | | |
| 5 | 1 | 10 | 45 | 120 | 210 | 252 | 210 | 120 | 45 | 10 | | |
| 6 | 1 | 15 | 105 | 455 | 1365 | 3003 | 5005 | 6435 | 6435 | 4995 | 2937 | 1125 |
| 7 | 1 | 21 | 210 | 1330 | 5985 | 20349 | 54264 | 116280 | 203490 | 293860 | 351225 | 342405 |
| 8 | 1 | 28 | 378 | 3276 | 20475 | 98280 | 376740 | 1184040 | 3108105 | 6906620 | 13112694 | 21322812 |
| 9 | 1 | 36 | 630 | 7140 | 58905 | 376992 | 1947792 | 8347680 | 30260340 | 94142440 | 254141370 | 599753700 |
| 10 | 1 | 45 | 990 | 14190 | 148995 | 1221759 | 8145060 | 45379620 | 215553195 | 886161035 | 3190035834 | 10145698290 |
| 11 | 1 | 55 | 1485 | 26235 | 341055 | 3478761 | 28989675 | 202927725 | 1217566350 | 6358397430 | 29248228548 | 119635845840 |
| $n$=12 | 1 | 66 | 2145 | 45760 | 720720 | 8936928 | 90858768 | 778789440 | 5743572120 | 37014122200 | 210979522776 | 1074029030256 |

| $G^{(0)}$ | $m$=12 | 13 | 14 | 15 | 16 | 17 | $m$= 18 |
|---|---|---|---|---|---|---|---|
| $n$=6 | 195 | | | | | | |
| 7 | 255640 | 131985 | 40950 | 5712 | | | |
| 8 | 29332947 | 32823084 | 28286520 | 17712016 | 7513632 | 1922760 | 223440 |
| 9 | 1238425650 | 2211404580 | 3316798800 | 4027258116 | 3822261219 | 2741630976 | 1427396544 |
| 10 | 28668181605 | 71916779655 | 158332102290 | 298777183440 | 469939341285 | 600955009695 | 611760126880 |
| 11 | 438280046820 | 1443628154430 | 4265969426730 | 11181453865032 | 25476016657410 | 49330500136830 | 79624401580350 |
| $n$=12 | 4921137880120 | 20410354904940 | 76804615396080 | 261330033475764 | 795039490678455 | 2124604757997810 | 4893346186174215 |

| $G^{(0)}$ | $m$=19 | 20 | 21 | 22 | 23 | $m$=24 |
|---|---|---|---|---|---|---|
| $n$=9 | 507370500 | 109907280 | 10929600 | | | |
| 10 | 485531549370 | 292849358445 | 129356267805 | 39394738800 | 7383474000 | 641277000 |
| 11 | 105518952278190 | 113319722405439 | 97279122118035 | 65610814845015 | 33933103318125 | 12970861393050 |
| $n$=12 | 9560350362065580 | 15657657703665516 | 21299396020002540 | 23862919970813940 | 21811038563094660 | 16066761920044110 |

| $G^{(0)}$ | $m$=25 | 26 | 27 | 28 | 29 | $m$=30 |
|---|---|---|---|---|---|---|
| $n$=11 | 3448843203960 | 569098807200 | 43859692800 | | | |
| $n$=12 | 9378949466187576 | 4233883781116440 | 1424007585518760 | 335673749980800 | 49451047430400 | 3424685806080 |

Fig. 1. Values of $G^{(0)}(n, m)$ for up to 12 vertices.

## 7. Running time and memory requirements

In this section, we establish a polynomial upper bound on the running time and the memory requirement of our sampling algorithm. We also report on computational results from an implementation of the counting formulae.

Since our algorithm for sampling random labeled planar graphs is an application of the well-known 'recursive method' for sampling [9,12,21], we outline the essentials only. The algorithm pre-calculates a number of dynamic programming arrays containing the values of $F$, $M$, $L$, $N$, $P$, $S$, $H$, $Q$, and $G$, before the actual random generation starts. Altogether these tables have $O(n^3)$ entries, and all entries are bounded by the number of planar graphs. Therefore the encoding length is $O(\log(n!\,38^n)) = O(n \log n)$ [8,22], and the total space requirement is in $O(n^4 \log n)$ bits. The computation of each entry involves a summation over $O(n^2)$ terms. Using a fast multiplication algorithm (see e.g. [7]), the tables can be filled in $\tilde{O}(n^6)$ time.

The values in the dynamic programming tables are used during the probabilistic decisions in a recursive construction of the labeled planar graph; it is essentially the inversion of the presented decomposition. For each entry, we scan over all the entries from which it was computed (there are at most $nm$ of them) and store the partial sums in a balanced binary tree, where each internal node contains the maximum of its left-hand siblings. The total size of the resulting data structure is $\tilde{O}(n^6)$, and it can be initialized in $\tilde{O}(n^6)$ time.

We assume that we can obtain random bits at unit cost. When given a random number between 1 and the sum over all leaves, we can find the corresponding table entry in one pass through the tree of partial sums, while reading each bit of the random number only a constant number of times, and hence in $O(n \log n)$ time. Then the procedure calls itself recursively for both factors of the product. Note that the sum of the bit lengths of both factors is linear in the bit length of the entry. It follows that the total running time for traversing the decomposition tree and creating the output is $\tilde{O}(n^2)$, and hence dominated by the generation of c-nets.

It is not necessary to create the binary trees for each entry of the tables. Instead, one can simply recompute some of the values from the preprocessing step and stop if the partial sum exceeds the random number. In this way, the recursive decomposition uses $\tilde{O}(n^6)$ time and $O(n^4 \log n)$ space. Now Theorem 1 follows by combining the results of this and the preceding section.

The counting part of our recurrences has been implemented in C++ using the GMP library for exact arithmetic [16]. A run for 50 vertices is completed within one hour on a 1.3 GHz PC using ca. 100 MB RAM. We also checked the recurrences and initial cases in Sections 3–6 using an independent counting method. A list of all unlabeled planar graphs with up to 12 vertices was generated by a program of Köthnig [17]. From these, the labeled planar graphs were

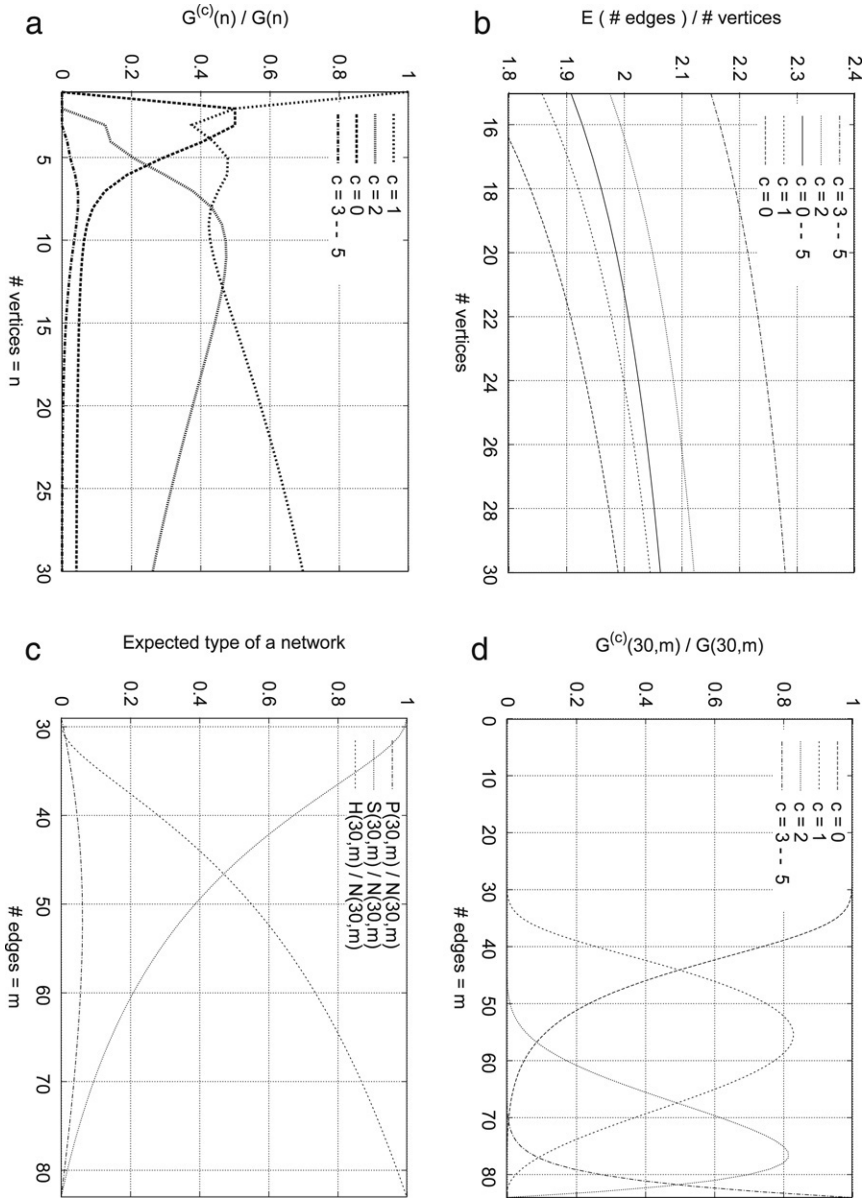


Fig. 2. Some counting results. (a) Expected connectivity, $n$ running (b) Edge density, $n$ running (c) Expected type of a network, $m$ running (d) Expected connectivity, $m$ running.

enumerated by 'brute force'. The unlabeled numbers, in turn, were confirmed by entries in Sloane's encyclopedia of integer sequences [25] and by [20]. The source code is available online, as well as the computed numbers for $n \leq 50$. The numbers for planar graphs upto 12 vertices are given in Fig. 1.

Using the computed numbers, we can study several basic questions about random labeled planar graphs. Here, and in the following, we let $G^{(c)}(n) := \sum_m G^{(c)}(n, m)$, etc. McDiarmid, Steger, and Welsh have shown that the quantity $(G^{(0)}(n)/n!)^{1/n}$ converges to a limit $\gamma_\ell$, the *labeled planar graph growth constant* [19], as $n \to \infty$. As an indicator for the speed of convergence, we computed the value of $G^{(c)}(n)/G^{(c)}(n-1)/n$ for various connecitvities $c$, e.g., $G^{(0)}(50)/G^{(0)}(49)/50 \doteq 25.2737$. The asymptotic fraction $c_\ell$ of connected labeled planar graphs is between $1/e$ and 1 [19]. Fig. 2(a) shows the value of $G^{(c)}(n)/G^{(0)}(n)$ for several ranges of the connectivity $c$. In particular, we have $G^{(1)}(50)/G^{(0)}(50) \doteq 0.960409$. The limit $e_\ell$ of the expected edge density of general (no connectivity requirement) labeled planar graphs is known to be $\geq 13/6 \doteq 1.86$ [14] and smaller than 2.54 [6] (even smaller than 2.52 [18]). The precise values for up to 50 vertices are shown in Fig. 2(b). In particular, this value for $n = 50$ is 2.12435. Fig. 2(c) and Fig. 2(d) show the distribution of the three types of a network and the connectivity of labeled planar graphs on 50 vertices with varying number of edges $m$, respectively.

Very recently, Giménez and Noy [15] determined the labeled planar graph growth constant $\gamma_\ell \doteq 27.2$, the asymptotic fraction of connected graphs $c_\ell \doteq 0.963$, and the limit of expected edge density $e_\ell \doteq 2.21$.

## 8. Conclusion

We have seen how to count and generate random planar graphs on a given number of vertices and edges using a recursive decomposition along the connectivity structure. A by-product of our result is that we can also generate *connected* and 2-*connected* labeled planar graphs uniformly at random. Moreover, it is easy to see that we can count and generate random planar *multigraphs* by only changing the initial values for planar networks as follows:

$$N(n,m) = P(n,m) \qquad \text{for } n = 2\,, m \geq 2$$
$$P_k(n,m) = 1 \qquad \text{for } n = 2\,, m = k\,, k \geq 1\,.$$

To increase the efficiency of the algorithm, one might want to apply a technique where the generated combinatorial objects have only approximately the correct size; this can then be turned into an exact generation procedure by rejection sampling. A general framework to tune and analyze such procedures has been developed in [1,11] and applied to structures derived by e.g. disjoint unions, products, sequences, and sets. To deal with planar graphs, it needs to be extended to the compose operation used in this paper.

## Acknowledgements

The authors would like to thank the referees for valuable comments. The first and the third authors were supported by the Deutsche Forschungsgemeinschaft (DFG) within the European graduate program 'Combinatorics, Geometry, and Computation' (No. GRK 588/2) and DFG grant Pr 413/1-2. The second author was supported by DFG grant Pr 296/3 and the German Federal Ministry of Education and Research within the Berlin Center for Genome Based Bioinformatics (BMBF grant no. 031 2705A).

## References

[1] C. Banderier, P. Flajolet, G. Schaeffer, M. Soria, Random maps, coalescing saddles, singularity analysis, and Airy phenomena, Random Structures and Algorithms 19 (2001) 194–246.
[2] E.A. Bender, Z. Gao, N.C. Wormald, The number of 2-connected labelled planar graph, Electronic Journal of Combinatorics 9 (2002).
[3] M. Bodirsky, C. Gröpl, D. Johannsen, M. Kang, A direct decomposition of 3-connected planar graphs, in: Proceedings of the 17th Annual International Conference on Formal Power Series and Algebraic Combinatorics, FPSAC05, Taormina, 2005.
[4] M. Bodirsky, C. Gröpl, M. Kang, Sampling unlabeled biconnected planar graphs, in: Proceedings of the 16th Annual International Symposium on Algorithms and Computation, 2005.
[5] M. Bodirsky, M. Kang, Generating outerplanar graphs uniformly at random, Combinatorics, Probability and Computing 15 (2006) 333–343.
[6] N. Bonichon, C. Gavoille, N. Hanusse, An information-theoretic upper bound of planar graphs using triangulation, in: Proceedings of the 20th Annual Symposium on Theoretical Aspects of Computer Science, STACS, 2003, pp. 499–510.
[7] P. Bürgisser, M. Clausen, M. Shokrollahi, Algebraic Complexity Theory, in: Grundlehren der Mathematischen Wissenschaften, vol. 315, Springer Verlag, 1997.
[8] A. Denise, M. Vasconcellos, D. Welsh, The random planar graph, Congressus Numerantium 113 (1996) 61–79.
[9] A. Denise, P. Zimmermann, Uniform random generation of decomposable structures using floating-point arithmetic, Theoretical Computer Science 218 (1999) 233–248.
[10] R. Diestel, Graph Theory, Springer-Verlag, New York, 1997.
[11] P. Duchon, P. Flajolet, G. Louchard, G. Schaeffer, Boltzmann samplers for the random generation of combinatorial structures, Combinatorics, Probability and Computing 13 (4–5) (2004) 577–625.
[12] P. Flajolet, P. Zimmerman, B. Van Cutsem, A calculus for the random generation of labelled combinatorial structures, Theoretical Computer Science 132 (1–2) (1994) 1–35.
[13] E. Fusy, D. Poulalhon, G. Schaeffer, Dissections and trees: Applications to optimal mesh encoding and random sampling, in: Proceedings of the Symposium on Discrete Algorithms, SODA'05, 2005.
[14] S. Gerke, C. McDiarmid, On the number of edges in random planar graphs, Combinatorics, Probability and Computing 13 (2004) 165–183.
[15] O. Giménez, M. Noy, Asymptotic enumeration and limit laws of planar graphs, Preprint.
[16] The GNU multiple precision arithmetic library, version 4.1.2. http://swox.com/gmp/.
[17] I. Köthnig, Humboldt-Universität zu Berlin, 2002. Personal communication.
[18] M. Löffler, Zufällige planare Graphen. Studienarbeit, Humboldt-Universität zu Berlin, 2003.

[19] C. McDiarmid, A. Steger, D.J. Welsh, Random planar graphs, Journal of Combinatorial Theory, Series B 93 (2005) 187–205.
[20] R. Mullin, P. Schellenberg, The enumeration of c-nets via quadrangulations, Journal of Combinatorial Theory 4 (1968) 259–276.
[21] A. Nijenhuis, H. Wilf, Combinatorial Algorithms, Academic Press Inc., 1979.
[22] D. Osthus, H.J. Prömel, A. Taraz, On random planar graphs, the number of planar graphs and their triangulations, Journal of Combinatorial Theory, Series B 88 (2003) 119–134.
[23] G. Schaeffer, Conjugaison d'arbres et cartes combinatoires aléatoires, Ph.D. Thesis, Université Bordeaux I, 1998.
[24] G. Schaeffer, Random sampling of large planar maps and convex polyhedra, in: Proc. of the Thirty-first Annual ACM Symposium on Theory of Computing, STOC'99, ACM press, Atlanta, Georgia, 1999, pp. 760–769.
[25] N.J.A. Sloane, The on-line encyclopedia of integer sequences, http://www.research.att.com/~njas/sequences/index.html, 2002.
[26] B.A. Trakhtenbrot, Towards a theory of non-repeating contact schemes, Trudy Mathematichiskogo Instituta imeni V.A. Steklova 51 (1958) 226–269 (in Russian).
[27] W. Tutte, A census of planar maps, Canadian Journal of Mathematics 15 (1963) 249–271.
[28] T. Walsh, Counting labelled three-connected and homeomorphically irreducible two-connected graphs, Journal of Combinatorial Theory 32 (1982) 1–11.
[29] T. Walsh, Counting nonisomorphic three-connected planar maps, Journal of Combinatorial Theory 32 (1982) 33–44.
[30] T. Walsh, Counting unlabelled three-connected and homeomorphically irreducible two-connected graphs, Journal of Combinatorial Theory 32 (1982) 12–32.
[31] T. Walsh, V.A. Liskovets, Ten steps to counting planar graphs, Congressus Numerantium 60 (1987) 269–277.