Generating Labeled Planar Graphs Uniformly at Random

Manuel Bodirsky[1], Clemens Gröpl[2], and Mihyun Kang[1]

[1] Humboldt-Universität zu Berlin, Germany
{bodirsky,kang}@informatik.hu-berlin.de

[2] Freie Universität Berlin, Germany
groepl@inf.fu-berlin.de

Abstract. We present an expected polynomial time algorithm to generate a labeled planar graph uniformly at random. To generate the planar graphs, we derive recurrence formulas that count all such graphs with n vertices and m edges, based on a decomposition into 1-, 2-, and 3-connected components. For 3-connected graphs we apply a recent random generation algorithm by Schaeffer and a counting formula by Mullin and Schellenberg.

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. [21]. An efficient random generation algorithm was recently obtained by Schaeffer [17]. Much less is known about random planar graphs, although they recently attracted much attention [3, 5, 6, 9, 13, 15]. Even the expected number of edges for random planar graphs is not known (both in the labeled and in the unlabeled case), and the gap between known upper and lower bounds is still large [5, 9, 15]. There are also some results on the asymptotic number of labeled planar graphs [3, 15]. If we had an efficient algorithm to generate a planar graph uniformly at random, we could experimentally verify conjectures about properties of the random planar graph. We could also use it to evaluate the average-case running times of algorithms on planar graphs. Denise, Vasconcellos and Welsh [6] introduced a Markov chain having the uniform distribution on all labeled planar graphs as its stationary distribution. However, the mixing time is unknown and seems hard to analyze, and is perhaps not even polynomial. Moreover, their algorithm only approximates the uniform distribution.

We obtain the first expected polynomial time algorithm to generate a labeled planar graph uniformly at random.

[1] Supported by the Deutsche Forschungsgemeinschaft (DFG) within the European graduate program 'Combinatorics, Geometry, and Computation' (No. GRK 588/2).

[2] 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).

Theorem 1. *A random planar graph with n vertices and m edges can be generated uniformly at random in expected time $O(n^{13/2})$ after a deterministic preprocessing of running time $O(n^7(\log n)^2(\log\log n))$. The memory requirement is $O(n^5 \log n)$ bits.*

We believe that the actual generation is much faster in practice, see Section 6. Our result uses known graph decomposition and counting techniques [22, 25] to reduce the counting and generation of labeled planar graphs to the counting and generation of 3-connected *rooted planar maps*, also called *c-nets*.

Usually a planar graph has many embeddings which are non-isomorphic as maps, but some graphs have a unique embedding. A classical theorem of Whitney (see e.g. [7]) 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, c-nets 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.

The generation 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 having the chosen numbers of vertices and edges. To generate a connected component with given numbers of vertices and edges, we decide for 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 [4], where the goal was to generate random outerplanar graphs. In the planar case we need to go one step further.

Trakhtenbrot [20] 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 template (see Theorem 2 below). Using this composition we can then employ known results about the counting and generation of 3-connected planar maps.

The concept of rooting plays an important role for the enumeration of planar maps. A *face-rooted* map is one with a distinguished edge which lies on the outer face and to which a direction is assigned. The 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 was achieved by Mullin and Schellenberg [14]. We invoke their closed formulas 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 algorithm by Schaeffer [18] running in expected polynomial time.

When we apply the various counting and generation subroutines along the stages of the connectivity decomposition, we must branch with the right probabilities. Instead of explicit (closed-form) counting formulas, which seem difficult to obtain, we derive recurrence formulas that can be evaluated in polynomial time using dynamic programming. These recurrence formulas can be translated immediately into a generation procedure.

The paper is organized as follows: In the next section we give the graph theoretic background for the decomposition of planar graphs along their connectivity structure. This decomposition guides us when we derive the counting formulas for planar graphs in the following three sections. We analyze the running time and memory requirements of the corresponding generation procedure in Section 7. Some results from an implementation of the counting part are shown in Section 8. We conclude with a discussion of variations of the approach.

2 Decomposition by Connectivity

Let us recall and fix some terminology [7, 22–24]. A *graph* will be assumed 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 (considered as vertices) of G, 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 compose 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.) 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 2 or more edges connected in *series* with the poles as its terminal vertices. A *bond* is a network consisting of 2 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 ([20], see [24]) 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: h-networks, p-networks, s-networks. 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 uniqueness is up to orientation of the edges of the core, and also up to their order if the core is a bond.*

A network is *simple* if it 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.

Let us define *compose operations* for the three stages $c = 0, 1, 2$ of the connectivity decomposition formally as follows. Assume that M and X are graphs on the vertex sets $[0 \mathrel{..} k-1]$ and $[0 \mathrel{..} i-1]$ and we want to compose them by identifying the vertices j of X with the vertices v_j of M, for $j = 0, \ldots, c-1$, 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 $[c \mathrel{..} n-1]$ which are designated for the remaining part of X. Let M' be the graph obtained by mapping the vertices of M to the set $[0 \mathrel{..} n-1] \setminus S$, retaining their relative order. Let X' be the graph obtained by mapping the vertices $[c \mathrel{..} i-1]$ of X to the set S, retaining their relative order, and mapping j to the image of v_j in M' for $j = 0, \ldots, c-1$. Then the result of the compose operation for the arguments M, $(v_0, \ldots, v_{c-1})$, X, and S is the graph with vertex set $[0 \mathrel{..} n-1]$ and edge set $E(M') \cup E(X')$.

We use $G^{(c)}(n,m)$ to denote the number of c-connected planar graphs with n vertices and m edges.

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 easy recurrence 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 a 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 recurrence formula

$$F_k(n,m) \;=\; \sum_{i=1}^{n-1} \sum_{j=0}^{m} \binom{n-1}{i-1} G^{(1)}(i,j) F_{k-1}(n-i, m-j)\,.$$

Thus it suffices to count connected graphs. But 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 $k \in [1 \mathrel{..} n]$ with 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

$S' \subseteq [1 \mathinner{..} n-1]$ uniformly at random and 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 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

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

and this immediately translates into a generation procedure.

Next we consider m_1-planars. The *root block* is the block containing the vertex 0. A recurrence formula 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 root block has b vertices and the c least labeled vertices in the root block are no cutvertices. Let us call them *$\mathrm{m}_{b,c}$-planars* and denote the number of $\mathrm{m}_{b,c}$-planars with n vertices and m edges by $M_{b,c}(n,m)$. Then $M_1(n,m) \;=\; \sum_{b=1}^{n} M_{b,1}(n,m)$. The initial cases are graphs without cutvertices. We have

$$M_{b,b}(n,m) \;=\; \begin{cases} G^{(2)}(n,m) & \text{for } \; b = n > 2 \\ 1 & \text{for } \; b = n \in \{1,2\} \; \text{ and } \; m = n-1 \\ 0 & \text{for } \; b \neq n. \end{cases}$$

To count $M_{b,c}$ using $M_{b,c+1}$, we split off the subgraph attached to the c-th least labeled vertex in the root block, if it is a cutvertex. This can be any connected planar graph. The remaining part is an $\mathrm{m}_{b,c+1}$-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 recurrence formula

$$M_{b,c}(n,m) \;=\; \sum_{i=1}^{n-1} \sum_{j=0}^{m-1} \binom{n-1}{i-1} G^{(1)}(i,j) M_{b,c+1}(n-i+1, m-j)\,.$$

Again, the generation procedure is straightforward.

5 2-Connected Planar Graphs

In this section we show how to count and generate 2-connected planar graphs. Note that every labeled 2-connected planar graph with n vertices and m edges is obtained from some simple planar network with n vertices and $m-1$ edges by adding an edge between the poles, then choosing $0 \leq x, y \leq n-1$, $x \neq y$, and exchanging the vertices 0 with x and 1 with y. Thus

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

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. Thus

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

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

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

If a p-network with n vertices is split into a p_1-network with i vertices and a p_{k-1}-network, there are $\binom{n-3}{i-3}$ ways how the vertex labels $[0 \mathrel{..} n-1]$ can be distributed among both sides, as the labels 0, 1, and 2 are fixed. We obtain the recurrence formula

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

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

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

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 how 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 then 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

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

h-Networks. Let us call an h-network whose core is a pseudo-brick on k edges an *h_k-network*, and denote the number of h_k-networks with n vertices and m edges by $H_k(n,m)$. Then $H(n,m) = \sum_{k=5}^{m} H_k(n,m)$, as the smallest pseudo-brick has 5 edges. We can order the edges of the core lexicographically by the vertex numbers. A recurrence formula similar to the p- and s-network case arises from replacing the edges of the core with components one at a time and in lexicographic order. To give names to the intermediate stages, let $H_{k,\ell}(n,m)$ be the number of $\mathrm{h}_{k,\ell}$-networks with n vertices and m edges, where an $\mathrm{h}_{k,\ell}$-network is an h_k-network in which the components corresponding to the first ℓ edges of the core are simple edges. Thus $H_{m,m}(n,m)$ is the number of pseudo-bricks with n vertices and m edges, and $H_{k,k}(n,m)=0$ for $k\neq m$. Applying the recurrence formula derived below for $\ell=k-1$ down to 0, we can calculate $H_k(n,m)=H_{k,0}(n,m)$, and hence, $H(n,m)$. For the initial case, we have

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

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): for we assign 0 to the root vertex, 1 to the other vertex of the root edge and the remaining labels to the remaining vertices, and neglect the orientation. To count $H_{k,\ell}$ using $H_{k,\ell+1}$, we split off the ℓ-th component of an $\mathrm{h}_{k,\ell}$-network, i.e., the component replacing the ℓ-th edge of the core. This can be a network of any of the three kinds. Thus

$$H_1(n,m) \;=\; \begin{cases} N(n,m)+N(n,m-1) & \text{for } n\geq 3,\, m\geq 2 \\ 1 & \text{for } n=2,\, m=1 \\ 0 & \text{otherwise}\,. \end{cases}$$

The remaining part is an $\mathrm{h}_{k,\ell+1}$-network. If the ℓ-th component has i vertices, then there are $\binom{n-2}{i-2}$ ways to choose them, as the vertices 0 and 1 are merged with the endpoints of the ℓ-th edge of the core, respecting their relative order. We obtain the recurrence formula

$$H_{k,\ell}(n,m) \;=\; \sum_{i=2}^{n-2} \sum_{j=1}^{m-k+1} \binom{n-2}{i-2} H_1(i,j) H_{k,\ell+1}(n-i+2, m-j+1)\,.$$

6 c-Nets

In the preceding sections, we have shown how to count and generate random planar graphs assuming that we can do so for *c-nets*, i.e., 3-connected simple rooted maps. For this one can use a formula for their number $Q(n,m)$ derived by Mullin and Schellenberg in [14]. Using Euler's formula, it asserts that

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

and otherwise

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

This concludes the counting task.

A *generation* algorithm for c-nets with given numbers of vertices and edges running in expected polynomial time algorithm is due to Schaeffer et al. [1, 2, 16–18]. Here we only outline the method. The c-net is obtained by extracting the 3-connected core from a 2-connected map. There is a linear time algorithm to generate 2-connected maps [16], and the extraction is linear as well [17]. If the parameters of the 2-connected map are tuned appropriately, chances are good that the resulting c-net will have the desired parameters. Otherwise the sample is rejected and the procedure restarts. A map with n vertices and m edges is said to have an *imbalance* x which is defined by $n+1 = m(\frac{1}{2}+x)$. To obtain a core with m edges and imbalance x, one should select a 2-connected map with imbalance $3x$ and $m/\alpha_0(3x)$ edges, where the *tuning ratio* is $\alpha_0(x) = \frac{(1-2x)(1+2x)}{3(1-2x/3)(1+2x/3)}$ [2, 17]. We have $\alpha_0(x) = \Omega(1/m)$ in the worst case. The expected number of iterations is $O(m^{2/3} + 1/p_\nu)$ for any given number of edges, where the probability p_ν that the core (whose size obeys a bimodal distribution) has around m edges is $p_\nu = \frac{16}{9}\alpha_0(3x)^2 = \Omega(1/m^2)$, and the $O(m^{2/3})$ term accounts for prescribing the exact number of edges. Prescribing also the number of vertices exactly (and not just up to a constant factor as in [17]) increases the running time by another factor $O(n^{1/2})$ (see [16, p. 140] and [18]). Thus a random c-net with m edges and imbalance x can be generated in expected time $O(m^{1+2+1/2+2}) = O(n^{11/2})$.

We conjecture that in fact a much faster generation should be possible based on two grounds: Most c-nets have an imbalance with $|x| \leq 1/2 - \varepsilon$, where $\varepsilon > 0$ is any constant. In this case the tuning ratio α_0 and hitting probability p_ν are bounded by constants and the expected running time reduces to $O(m^{1+2/3+1/2}) = O(n^{13/6})$.

Moreover, if we are about to generate many planar graphs, we might store the rejected samples for future use, possibly resulting in a near-linear amortized running time at the expense of a larger (but still polynomial) memory requirement.

7 Running Time and Memory Requirements

In this section we establish a polynomial upper bound on the expected running time and the memory requirement of our algorithm.

A number of dynamic programming arrays has to be pre-calculated before the actual random generation starts. As an example, consider the recurrence formula for $H_{k,\ell}(n,m)$. The number of entries is $O(n^4)$ for all tables. All entries are bounded by the number of all planar graphs. Therefore the encoding length of each entry is $O(\log(n!\,38^n)) = O(n\log n)$ [6, 15] and the total space requirement is $O(n^5 \log n)$ bit. The calculation of each entry involves a summation over $O(n^2)$ terms. Using a fast multiplication algorithm, the precomputation time is $O(n^7(\log n)^2(\log\log n))$.

We assume that we can obtain random bits at unit cost. In order to prepare for branching with the right probabilities, we can easily calculate the necessary partial sums in a second pass over the dynamic programming arrays. We can then perform random decisions with the right probabilities in time linear in the encoding length, i.e., in $O(n \log n)$.

The total expected time spent in all calls to Schaeffer's c-net generation algorithm is bounded by $O(n^{13/2})$ (but we believe it is much faster in practice, see Section 6). Similarly, the random decisions for the connectivity decomposition require $O(n^2 \log n)$ time in total. An h-element subset of a k-element ground set can be chosen in $O(h \log k)$ time, hence the total time spent for random decisions for the label assignments during the composition is $O(n^2 \log n)$ as well. The compose operation itself is linear and requires at most $O(n^2)$ total time.

We see that the running time is dominated by $O(n^7(\log n)^2(\log\log n))$ for the preprocessing and $O(n^{13/2})$ (in expectation) for the random generation of c-nets. The space requirement is $O(n^5 \log n)$ bits due to the dynamic programming arrays.

8 Experimental Results

In this section we report on computational results from an implementation of the counting formulas. The program was written in C++ using the GMP library for exact arithmetic [10]. A run for 30 vertices completed within one hour on a 1.3 GHz PC using 605 MB RAM. We also checked the recurrences and initial cases in Section 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 [11]. From these the labeled planar graphs were enumerated by 'brute force'. The unlabeled numbers, in turn, were confirmed by entries in Sloane's encyclopedia of integer sequences [19] and by [14].

A basic open question is the expected edge density of a random labeled planar graph. The limit for general (no connectivity requirement) labeled planar graphs is known to be $\geq 13/6 \doteq 1.86$ [9] and ≤ 2.54 [5] (even ≤ 2.52 [12]), and Markov chain experiments [12] indicate a value around 2.21. The precise values for up to 30 vertices are shown in Figure 1 (a). Note the high influence of connectivity.

McDiarmid, Steger, and Welsh proved that the quantity $(G(n)/n!)^{1/n}$ converges to a limit γ_ℓ, the *labeled planar graph growth constant* [13], as $n \to \infty$. (Here and in the following, we let $G(n) := \sum_m G(n,m)$, etc.) As an indicator for the speed of convergence, we plot the value of $G(n)/G(n-1)/n$ for several ranges of the connecitvity c in Figure 1 (b). For $c = 2$ the limit is $\doteq 26.2$ (namely, $1/x_0$ in [3, Thm. 1 (b)]).

The asympototic fraction of disconnected labeled planar graphs is $\leq 1 - 1/e$ and > 0 [13]. Figure 1 (c) shows the value of $G^{(c)}(n)/G(n)$ for various ranges of the connectivity c, and Figure 1 (d) shows the distribution of the three types of a network, i. e., P, S, or H. As some of these quantities apparently converge fast, here we provide numerical values:

n	$\frac{G^{(0)}(n)}{G(n)}$	$\frac{G^{\{3,4,5\}}(n)}{G(n)}$	$\frac{P(n)}{N(n)}$	$\frac{S(n)}{N(n)}$	$\frac{H(n)}{N(n)}$
29	.0420555	.0003418	.0511210	.1740587	.7748204
30	.0418449	.0002650	.0508880	.1734036	.7757084

Figures 2 (e), (f), (g) illustrate the influence of the edge density on the connectivity. It would be very interesting to determine the asymptotic shape of Figure 2 (g). Perhaps surprisingly, the edge density is also highly related to the expected type of a network, see Figure 2 (h). – All this calls for further investigation.

9 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. Therefore 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:

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

It seems difficult to simplify our counting recurrences to closed formulas. In this way one could eliminate the need for a preprocessing stage. Using generating functions Bender, Gao and Wormald obtained an *asymptotic* formula for the number of labeled 2-connected graphs [3].

To increase the efficiency of the algorithm one might want to apply a technique where the generated combinatorial objects only have 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 [2, 8] 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.

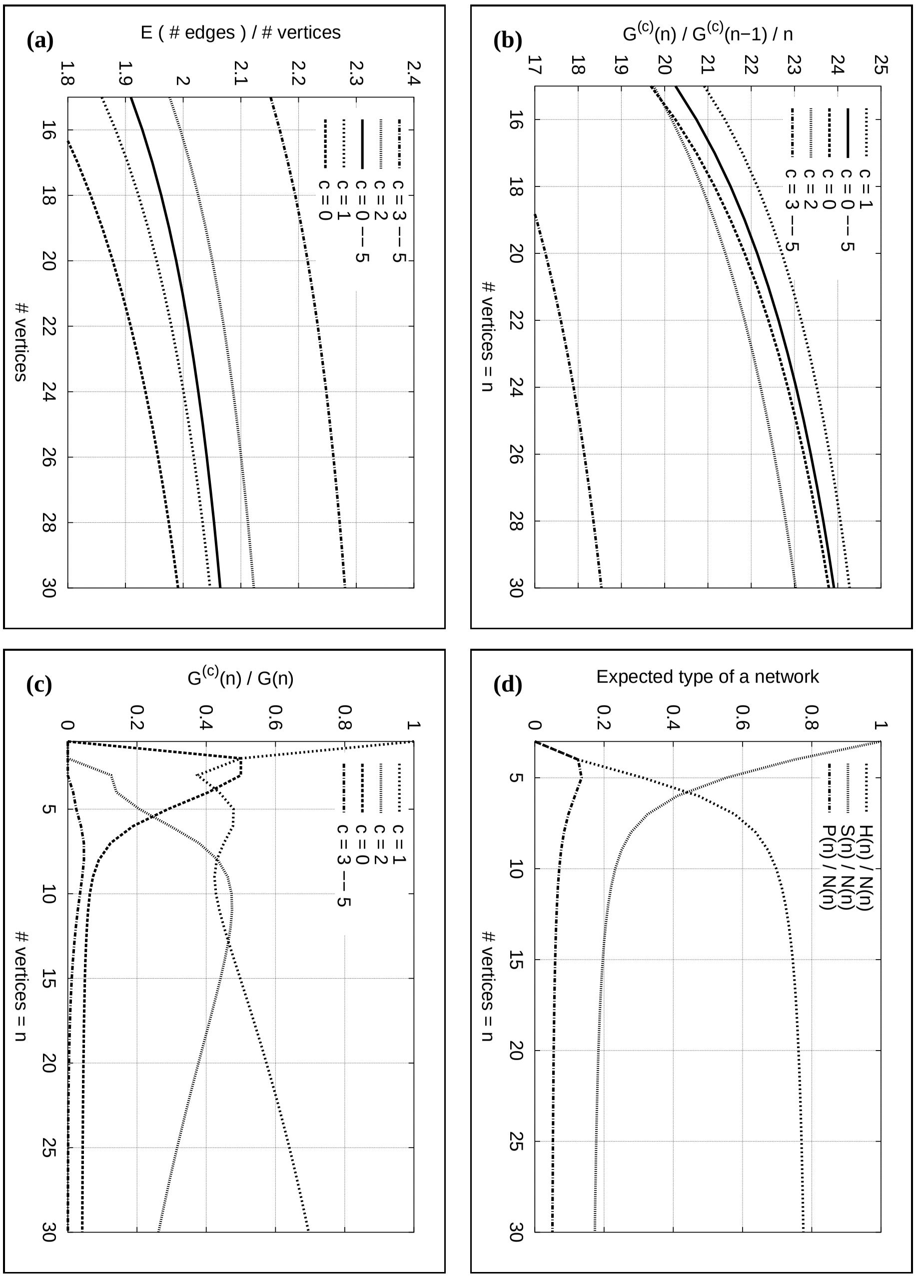

Fig. 1. Some counting results showing the dependency on the number of vertices n. (a) Edge density of a random labeled planar graph. (b) Growth rate of the number of labeled planar graphs. (c) Expected connectivity. (d) Expected type of a network.

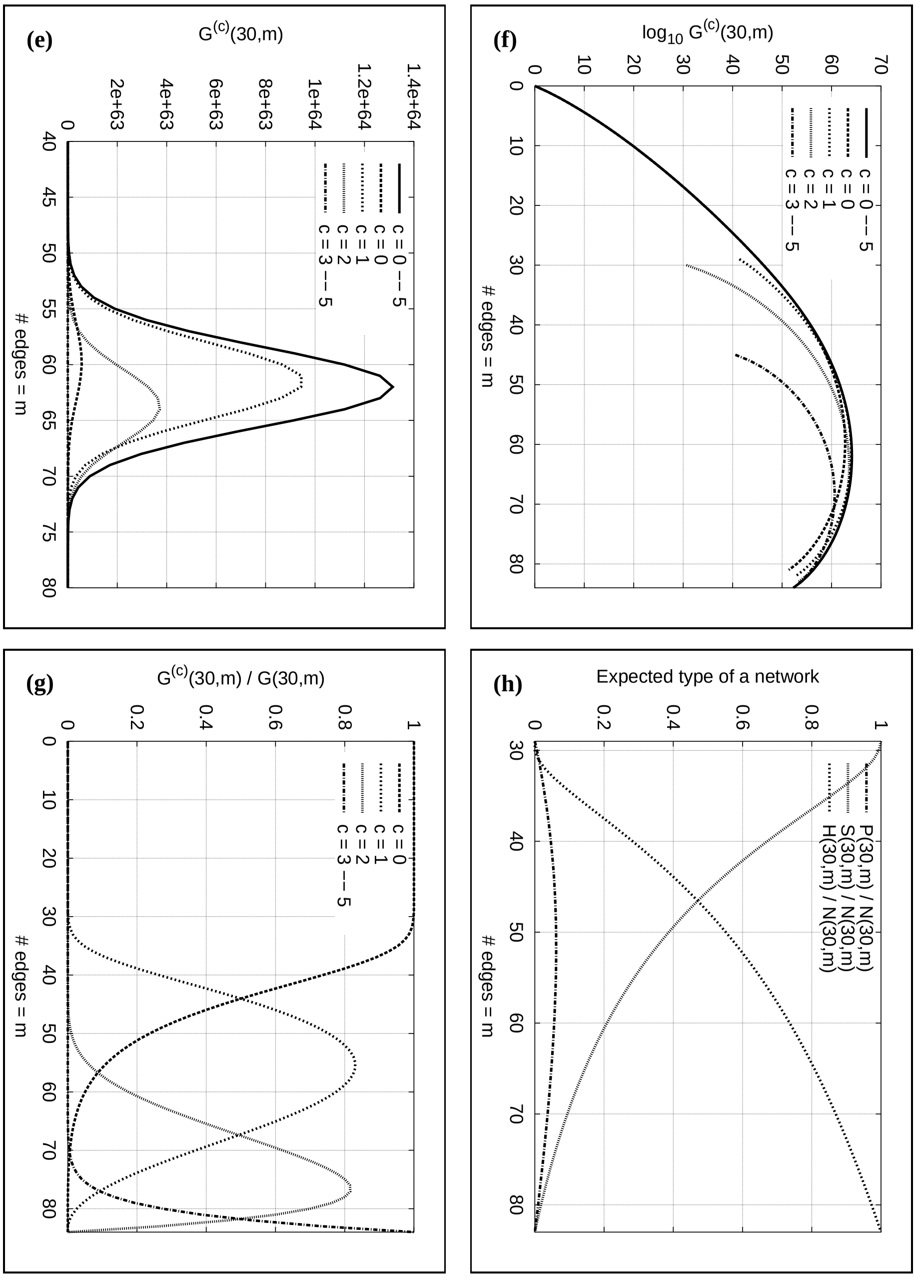

Fig. 2. Some counting results for labeled planar graphs on 30 vertices. The figures show the dependency on the number of edges m and the connectivity c. (e) Number of c-connected labeled planar graphs. (f) Similar in logarithmic scale. (g) Expected connectivity. (h) Expected type of a network.

References

1. C. Banderier, P. Flajolet, G. Schaeffer, and M. Soria. Planar maps and Airy phenomena. In *ICALP'00*, number 1853 in LNCS, pages 388–402, 2000.
2. C. Banderier, P. Flajolet, G. Schaeffer, and M. Soria. Random maps, coalescing saddles, singularity analysis, and Airy phenomena. *Random Structures and Algorithms*, 19:194–246, 2001.
3. A. Bender, Z. Gao, and N. Wormald. The number of labeled 2-connected planar graphs. Preprint, 2000.
4. M. Bodirsky and M. Kang. Generating random outerplanar graphs. Submitted. Presented at the 1st workshop on Algorithms for Listing, Counting, and Enumeration ALICE 03, 2003. Available at `http://www.informatik.hu-berlin.de/~bodirsky/publications`.
5. N. Bonichon, C. Gavoille, and N. Hanusse. An information-theoretic upper bound of planar graphs using triangulation. In *In 20th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, 2003.
6. A. Denise, M. Vasconcellos, and D. Welsh. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.
7. R. Diestel. *Graph Theory*. Springer–Verlag, New York, 1997.
8. P. Duchon, P. Flajolet, G. Louchard, and G. Schaeffer. Random sampling from Boltzmann principles. In *ICALP 2002*, LNCS, pages 501–513.
9. S. Gerke and C. McDiarmid. On the number of edges in random planar graphs. Submitted.
10. The GNU multiple precision arithmetic library, version 4.1.2. `http://swox.com/gmp/`.
11. I. Köthnig. Personal communication. Humboldt-Universität zu Berlin, 2002.
12. M. Löffler. Zufällige planare Graphen. Studienarbeit, Humboldt-Universität zu Berlin, 2003.
13. C. McDiarmid, A. Steger, and D. J. Welsh. Random planar graphs. Submitted, 2003.
14. R. Mullin and P. Schellenberg. The enumeration of c-nets via quadrangulations. *Journal of Combinatorial Theory*, 4:259–276, 1968.
15. D. Osthus, H. J. Prömel, and A. Taraz. On random planar graphs, the number of planar graphs and their triangulations. *Jombinatorial Theory, Series B*, to appear.
16. G. Schaeffer. *Conjugaison d'arbres et cartes combinatoires aléatoires*. PhD thesis, Université Bordeaux I, 1998.
17. 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)*, pages 760–769, Atlanta, Georgia, May 1999. ACM press.
18. G. Schaeffer. Personal communication, 2002.
19. N. J. A. Sloane. The on-line encyclopedia of integer sequences. `http://www.research.att.com/~njas/sequences/index.html`, 2002.
20. B. A. Trakhtenbrot. Towards a theory of non-repeating contact schemes. *Trudi Mat. Inst. Akad. Nauk SSSR*, 51:226–269, 1958. [In Russian].
21. W. Tutte. A census of planar maps. *Canad. J. Math.*, 15:249–271, 1963.
22. T. Walsh. Counting labelled three-connected and homeomorphically irreducible two-connected graphs. *J. Combin. Theory*, 32:1–11, 1982.
23. T. Walsh. Counting nonisomorphic three-connected planar maps. *J. Combin. Theory*, 32:33–44, 1982.
24. T. Walsh. Counting unlabelled three-connected and homeomorphically irreducible two-connected graphs. *J. Combin. Theory*, 32:12–32, 1982.
25. T. Walsh and V. A. Liskovets. Ten steps to counting planar graphs. In *Eighteenth Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Congr. Numer.*, volume 60, pages 269–277, 1987.