Sampling Unlabeled Biconnected Planar Graphs

Manuel Bodirsky[1], Clemens Gröpl[2,4], and Mihyun Kang [1,3]

[1] Humboldt-Universität zu Berlin, Institut für Informatik
Unter den Linden 6, D-10099 Berlin, Germany
{bodirsky,kang}@informatik.hu-berlin.de

[2] Freie Universität Berlin, Institut für Informatik
Takustraße 9, D-14195 Berlin, Germany
groepl@inf.fu-berlin.de

Abstract. We present an expected polynomial time algorithm to generate a 2-connected *unlabeled* planar graph uniformly at random. To do this we first derive recurrence formulas to count the exact number of *rooted* 2-connected planar graphs, based on a decomposition along the connectivity structure. For 3-connected planar graphs we use the fact that they have a unique embedding on the sphere. Special care has to be taken for rooted graphs that have a *sense-reversing* or a *pole-exchanging* automorphism. We prove a bijection between such symmetric objects and certain *colored networks*. These colored networks can again be decomposed along their connectivity structure. All the numbers can be evaluated in polynomial time by dynamic programming. To generate 2-connected unlabeled planar graphs *without a root* uniformly at random we apply *rejection sampling* and obtain an expected polynomial time algorithm.

1 Introduction

While there is an exact recursive counting formula and an efficient sampling algorithm for *rooted maps* [22], for *labeled and unlabeled outerplanar* graphs [8], and for *labeled planar graphs* [7], there is no such counting formula and sampling algorithm known for *unlabeled* planar graphs, i.e., graphs that can be embedded in the plane, considered up to isomorphism. Such a counting formula and sampling algorithm would be useful to verify statements about the *random planar graph*, which recently attracted attention [2,9,10,15,19,21], mainly in the labeled setting due to the lack of techniques for unlabeled planar structures. It is well known that almost all graphs have a small automorphism group. However, this is not the case for planar graphs, even if they are 2-connected: Bender, Gao, and Wormald [2] showed that almost all 2-connected planar graphs have a large automorphism group. Thus the difference between the labeled and the unlabeled setting is essential.

All approaches to count planar graphs are based on decompositions along the k-connected components of a graph [2,4,7,16,27]. A graph is decomposed into components, a component into blocks, and a block into bricks, which are essentially the 3-connected parts of the graph. Three-connected graphs have a unique embedding on the sphere, and thus can be further decomposed by geometric arguments (they stand in a one-to-one correspondence to the number of isomorphism types of the edge graphs of convex polyhedra [23]).

[3] Supported by the Deutsche Forschungsgemeinschaft (DFG Pr 296/7-3).
[4] Supported by the German Federal Ministry of Education and Research within the Berlin Center for Genome Based Bioinformatics (BMBF grant no. 031 2705A).

The asymptotic number [3] and sampling procedures [5, 13] for 3-connected planar graphs are known.

In this paper we present an algorithm that generates *two-connected* graphs on m edges uniformly at random in expected polynomial time in m. Such graphs have in general many automorphisms and also might have many embeddings on the sphere. It is easy to modify the algorithm e.g. to sample graphs with a given number of edges *and* vertices.

Strategy. To count and sample unlabeled 2-connected planar graphs, we first have to *root* them. Here, the *root* of a graph is a distinguished directed edge, and rooted planar graphs are counted up to isomorphisms that map the root of one graph to the root of the other graph. We also call such a rooted 2-connected planar graph a *(planar) network*. Clearly, generating a random rooted planar graph and then simply ignoring the root edge does not yield the uniform distribution, since unlabeled graphs might correspond to different numbers of rooted graphs. But this imbalance can be compensated by *rejection sampling*, i.e., the sampling procedure is restarted with a probability that is inverse proportional to the size of the orbit of the root. In this way we can sample unlabeled 2-connected planar graphs in expected polynomial time.

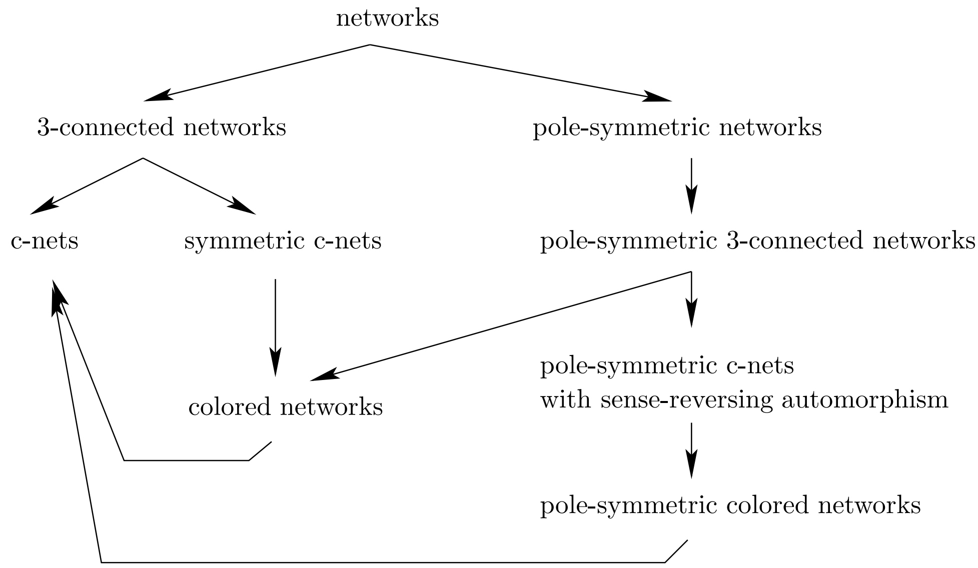

Fig. 1. Dependencies of the sections and concepts in this paper.

We *decompose* networks along their connectivity structure. Our approach here is related to the one described in [7] for labeled planar graphs, but for unlabeled structures several new techniques are necessary. A classical theorem of Whitney (see e. g. [12]) says that rooted 3-connected planar graphs, i.e., 3-connected networks, can have either one or two embeddings in the plane where the root edge is embedded on the outer face. Such embedded three-connected networks are called *c-nets*. In the case that both embeddings of the 3-

connected graph are isomorphic, we say that it has a *sense-reversing automorphism* or it is *symmetric*. To count symmetric c-nets we present a new bijective correspondence to *colored networks* (defined below), and a decomposition of these objects. We also have to consider rooted graphs with an automorphism that reverts the direction of the root; such a graph is called *pole-symmetric*. We present a decomposition of pole-symmetric networks, and finally also that of pole-symmetric c-nets with a sense-reversing automorphism. We exploit the fact that the dual of a pole-symmetric c-net is a c-net with a sense-reversing automorphism.

As a final step we use a deterministic polynomial time sampling algorithm for c-nets described in [5]. A faster algorithm to sample c-nets uniformly at random in expected polynomial running time can be found in [1]. However, we need the new algorithm, since it can be adapted to generate c-nets with a certain specified number of edges on the outer face, which we need in the generation algorithm for unlabeled 2-connected planar graphs.

2 Graph Decompositions

In this section we introduce concepts, which we need for a decomposition of graphs. A *(planar) network* N is a simple connected graph with distinguished vertices $s \neq t$ and a directed edge $e = st$ such that if we insert the edge e into N, then the resulting multi-graph N^* is 2-connected and planar. The edge $e = st$ is called the *root* of the network, and the vertices s and t are called the *poles* of the network; if s and t were already connected by an edge in N, we introduce a multi-edge in N^*. We say that a network N is k-connected iff N^* is k-connected.

A k-point set K of a graph G such that $G-K$ becomes disconnected is called a *k-cut* of G. A single point that forms a 1-cut is called a *cut-vertex*. Every 2-cut $\{k_1, k_2\}$ of a network N induces a partition of the edge set, and each of the partition classes is again a network with the poles k_1 and k_2 . These networks are called *subnetworks* of N, and they are called *proper* if they contain at least two edges. We now show that every network is either an *s-, p-, or h-network*; these appeared in various slightly different forms in the literature [24–26].

Theorem 1. *A network N with more than two edges is of precisely one of the following types:*

s: There is a cut-vertex in N that separates s and t.
p: The vertices s and t are adjacent, or $\{s, t\}$ is a 2-cut of N.
h: The vertices s and t are not adjacent, and N is built from a uniquely determined 3-connected network H by replacing edges of H with other networks.

Proof. Clearly, the three types of networks are disjoint. If N contains a 2-cut, one of the corresponding subnetworks does not contain the root. We replace this subnetwork by an edge (if the 2-cut was not already joined by an edge). If we iterate this process, eventually the graph will be 3-connected, or an s- or p-graph. Since the replacement operations are confluent and terminating, the resulting 3-connected graph is unique. □

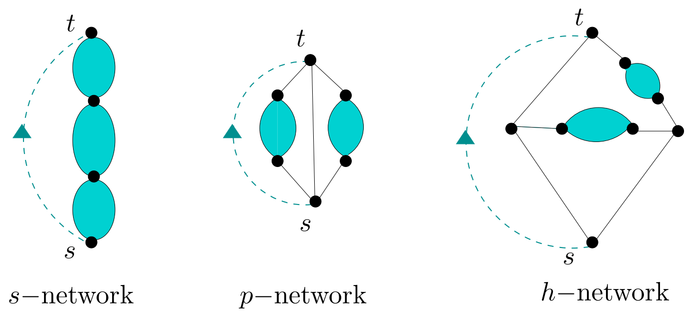

Fig. 2. Examples of the three types of networks

For examples of the three types of networks see Figure 2. As already mentioned in the introduction, a 3-connected network might have one or two embeddings where the root lies on the outer face. A 3-connected network embedded in one of these ways is called a *c-net*.

3 Counting Networks

In this section we present a decomposition of networks and derive recurrence formulas to count them. Let $n(m)$ be the number of networks with m edges. According to Theorem 1 we have $n(m) = s(m) + p(m) + h(m)$, where $s(m)$, $p(m)$, and $h(m)$ counts the number of s-, p-, and h-networks with m edges, respectively. The generation of these objects can be considered as a reversed decomposition, where the decisions are made with the right probabilities that can be computed using the counting formulas.

s-networks. Note that each s-network has a unique cut vertex v that is closest to the pole s (in this paper, *closest* is meant with respect to the length of the shortest connecting path). Then s and v determine a subnetwork, which is a p- or an h-network, and the remaining network with the poles v and t. Thus we have:

$$s(m) = \sum_{j=1} (p(j) + h(j))n(m-j) \, .$$

p-networks. Let $p_l(m)$ denote the number of p-networks where the number of edges of the largest network that replaces an edge of the core is bounded by l. The index k in the formula below denotes the number of networks of order l that replace an edge in the core.

$$\begin{aligned} p(m) &= p_m(m) \\ p_l(m) &= \sum_{k=0}^{\lfloor m/l \rfloor} \binom{s(l) + h(l) + k - 1}{k} p_{l-1}(m - kl) \, . \end{aligned}$$

h-networks. Let N be an h-network. Theorem 1 asserts that there is a unique rooted 3-connected network H, such that we can derive N from H by replacing edges of H with subnetworks. We call H the *core* of N and denote $H = \text{core}(N)$. We call N *symmetric* if it has a *sense-reversing automorphism* φ, i. e., $\varphi \neq \mathrm{id}$, but $\varphi(s) = s$ and $\varphi(t) = t$, and *asymmetric* otherwise. If N is asymmetric, one can uniquely order the edges of core(N): The idea is to label the vertices of the core according to their occurrence in a depth first search traversal of the core, beginning with the root edge and visiting the neighbors of a vertex in clockwise order with respect to one of the (at most two) possible embeddings of the core. The edges are then labeled by the vertex labels obtained from the depth first search traversal. Then we lexicographically compare the sequence of these edge labels in the order they were visited by the depth first search. If the core is asymmetric, one of the sequences is smaller than the other; thus we can distinguish between the two embeddings. If the network has a symmetric core, both edge sequences are the same unless we have inserted two different subnetworks into a pair of core edges, in which case we can again distinguish between the two embeddings.

If H is symmetric, we order the edges of the core in the following way. We start with the edges uv where $u = \varphi(u)$ and $v = \varphi(v)$ according to the traversal; We call such edges *blue*. Then we list the edges uv where $u = \varphi(v)$ and $v = \varphi(u)$ according to the traversal; We call such edges *red*. We continue with the edges that are not fixed by the nontrivial automorphism φ, and order them according to the above traversal. Edges and their images, which we call *corresponding edges*, are ordered arbitrarily.

To count the number of symmetric and asymmetric h-networks we repeatedly replace subnetworks in the above order. It is not difficult to fomulate corresponding recursive counting formulas; for details we refer to the full version of the paper [6]. We are finally left with the problems (a) to count and sample networks *with a pole-exchanging automorphism* – see Section 6, (b) to count and sample *3-connected symmetric* networks – see Section 4, and (c) to sample *3-connected asymmetric* networks. For this last task, we apply rejection sampling. That is, we first generate an arbitrary 3-connected network. We then check whether it has such a symmetry, which can be done in linear time [18]. If yes, we restart the algorithm. If no, we output the asymmetric network. Since almost all 3-connected networks do not have a sense-reversing automorphism (see [3] for a much stronger result), the expected number of restarts is constant, and we obtain an expected polynomial time algorithm.

4 Symmetric c-nets

This section contains one of the main ideas to deal with symmetries when counting unlabeled planar graphs. We want to count 3-connected planar networks with a distinguished directed edge, up to isomorphisms that fix this edge. There might be one or two embeddings where the root lies at the outer face. Embedded 3-connected networks are called *c-nets*. As mentioned in the introduction, counting formulas and sampling procedures for c-nets are known. If a network has a nontrivial automorphism that fixes the root edge, we call this automorphism *sense-reversing*, and say that the network is *symmetric*. Clearly, if we can compute the number of symmetric 3-connected networks, then we can also compute the number of asymmetric 3-connected networks.

Let H be a symmetric 3-connected planar network, and φ its nontrivial sense-reversing automorphism. A vertex v of H is called *blue* if $\varphi(v) = v$, and *red* if v is connected to $\varphi(v)$ by an edge. The edge $v\varphi(v)$ is also called red. An edge uv of a colored network is blue if both u and v are blue. (Red and blue edges were already defined in Section 3.) Thus a vertex or an edge is either blue, red, or uncolored, and the poles and the root are blue. We can think of H as being embedded in the plane in such a way that φ corresponds to a reflection, the blue vertices being aligned on the reflection axis, and the red vertices having an edge crossing this axis perpendicularly (see Fig. 3, left part). Our arguments, however, do not rely on such a representation.

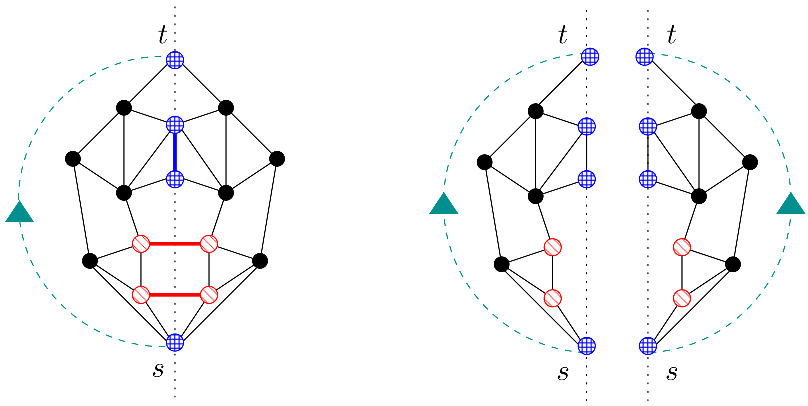

Fig. 3. Decomposition of a symmetric h-network

If we remove from H the blue vertices and their incident edges, and also remove the red edges (that is, we cut H along the symmetry axis), then the resulting graph has exactly two connected components (see Fig. 3). The graphs induced by these components and the blue vertices are isomorphic and will be called H_1 and H_2. We claim that H_1^* is 2-connected, and hence H_1 is a network rooted at s and t: Suppose there is a cut-vertex in H_1^*. Then this cut-vertex together with the corresponding cut-vertex in H_2^* is a 2-cut in H^*, contradicting the 3-connectivity of H^*. Now we extract some more properties of the graphs H_1 and H_2 and define *colored networks*. They are defined in such a way that we can recursively decompose them, and that we can establish a bijection between symmetric h-networks and certain colored networks.

Definition 1. *A* colored network *is a network N, where some vertices are colored* red *and* blue*, satisfying the following.*

(P1) *N^* has a plane embedding s.t. all colored vertices and the poles lie on the outer face.*
(P2) *N and every proper subnetwork of N contain at least one colored vertex.*
(P3) *No subnetwork of N has two blue poles.*

Then the bijection to symmetric 3-connected networks is as follows. The proof can be found in the long version of the paper [6].

Theorem 2. *For all m, b, r there is a bijection between the following two sets of objects:*

(i) *colored networks with $(m+b-r)/2$ edges and blue poles, where b is the number of blue edges and r the number of red vertices, and*
(ii) *3-connected networks with m edges having a nontrivial automorphism that fixes $b+r$ edges, and point-wise fixes the root and b other edges.*

5 Counting Colored Networks

The recurrences to count the number of colored networks follow very much the decomposition that we had in Section 3, but we have to control the possible colors of the poles. Another difficulty is that in the recursive decomposition we might or might not have a blue cut-vertex in the colored network without the root edge. However, we can handle this with the help of appropriately chosen counting functions. The details and counting formulas can be found in the full version of the paper [6]. We briefly comment on all the types of networks.

Colored *s*-networks. Let u be the cut-vertex in S that is closest to s. If at least one of the poles s, t is blue, then S can not have a blue cut-vertex (in particular, u is not blue). The cut-vertex u induces a colored p- or h-network with poles s, u, and a remaining part with poles u, t, which is an arbitrary colored network that has no blue cut-vertex.

Colored *p*-networks. Due to property $(\mathsf{P1}-\mathsf{P2})$ all the colored vertices of a colored p-network must lie in one of its parts, and the remaining networks must consist of a single edge. If at least one of the poles is blue, the colored part has no blue cut-vertex.

Colored *h*-networks. There is a unique embedding of the core of a colored h-network H into the plane where the root edge and the core edges replaced by colored networks lie on the outer face. To decompose H, we need to control the number of edges on the outer face. If an edge uv on the outer face of the core is not an edge in H, $\{u, v\}$ is a 2-cut in H and determines a subnetwork S. Due to property $(\mathsf{P3})$ it is not possible that both u, v are blue. If either u or v is blue, then $\{u, v\}$ induces a colored network with no blue cut-vertex. It might be the case that all colored vertices lie in S. Then the remaining network after the replacement of S is 3-connected with a specified number of edges on the outer face. The number of such graphs is counted in [5].

6 Pole-symmetric Networks

We saw in Section 3 that in a symmetric h-network with a sense-reversing automorphism φ a red edge uv of the core (i.e., $\varphi(u) = v$ and $\varphi(v) = u$) can only be replaced by a *pole-symmetric* subnetwork, that is, a subnetwork with an automorphism ψ that exchanges s and t. Such networks are further decomposed in this section.

Pole-symmetric s-networks. Here we split off the same p- or h-network at both poles simultaneously. What remains is either again a pole-symmetric network, or an edge, or a vertex.

Pole-symmetric p-networks. There may be several pole-symmetric s- or h-networks between s and t, and s and t may or may not be adjacent.

Pole-symmetric h-networks. We want to control the number of pole-symmetric h-networks with and without a sense-reversing automorphism φ satisfying $\varphi(s) = s$ and $\varphi(t) = t$. In the case where we do not have a sense-reversing automorphism, we order the edges of the core of H in such a way that blue edges uv where $\psi(u) = u$ and $\psi(v) = v$ come first, followed by the red edges uv where $\psi(u) = v$ and $\psi(v) = u$. Finally we have the uncolored edges, ordered in such a way that corresponding uncolored edges with respect to the pole-symmetry are consecutive – but we do not care about their order.

In the case that we have a sense-reversing automorphism φ, we order the edges of the core in such a way that we start with the blue edges with respect to φ, and then the blue edges with respect to ψ. Next we list the red edges with respect to φ and then the red edges with respect to ψ. Finally, we list corresponding edges with respect to φ consecutively, which are followed by the two corresponding edges with respect to ψ, respectively. Similarly as in Section 3 it is now possible to formulate recurrences for these functions, and a sampling procedure; again we have to refer to the full version of the paper [6] for details.

7 Pole-symmetric c-nets with a Sense-reversing Automorphism

To compute the number of pole-symmetric networks with a sense-reversing automorphism, we again use colored networks, but impose the additional constraint that the colored network has a pole-exchanging automorphism. Along the lines of Theorem 2 we have a bijection between these pole-symmetric colored networks and pole-symmetric networks with a sense-reversing automorphism. The decomposition of pole-symmetric colored networks is a straightforward combination of the ideas in Section 4 and 6.

When we remove the last colored subnetwork in a pole-symmetric colored h-network, we have an embedded 3-connected pole-symmetric network with l edges on the outer face. The dual of such an object is an embedded 3-connected network with a sense-reversing automorphism where the s-pole has degree l (blue edges correspond to red edges and vice versa). It is possible to modify the decomposition of colored networks in Section 4 to control also this parameter.

8 Conclusion

We presented a decomposition strategy for unlabeled 2-connected planar graphs along the connectivity structure. In order to count these objects we need a *unique* decomposition. Thus we used the well-known concept of a *root* and *planar networks*. For 3-connected networks, however, we also had to control whether there is a sense-reversing automorphism or not, which in turn required to control networks that have a pole-exchanging automorphism, and networks that have both a sense-reversing and a pole-exchanging automorphism. For this purpose we introduced the concept of *colored* networks, and proved several bijections. The decomposition together with the counting formulas can be used for a polynomial time sampling procedure for planar networks.

Theorem 3. *There is an algorithm that generates an unlabeled 2-connected planar graph with m edges uniformly at random in expected polynomial time.*

Proof. The algorithm first generates a planar network N with m edges, using the above decomposition and the values of the counting formulas that can be computed efficiently using dynamic programming. Note that the representation size of all the numbers in this paper is linear, since we deal with unlabeled structures. We use at most six-dimensional tables (in Section 6). The summation there runs over one parameter, and within the sum we have to perform a multiplication with large numbers, which can be done in quadratic time. Hence, the overall running time for the computation of the values is within $O(m^9)$.

With these values we can make the correct probabilistic decisions in a recursive construction of a planar network according to the presented decomposition – this method is standard and known as the *recursive method* for sampling [11, 14, 20]. Then the algorithm computes the number of orbits o in the automorphism group of the unrooted graph, which can be done in linear time, see e.g. [17], and outputs the graph with probability $1/o$. Since the number of edges in a planar graph is linear, the expected number of restarts is also linear. Thus the overall expected running time is in $O(m^9)$. If we do not charge for the costs for computing the values in the table and the partial sums of the formulas, e.g. because we performed a precomputation step, the generation can be done in cubic time. □

The counting formulas resulting from the ideas presented in this paper can easily be extended to graphs with a specified number of vertices. It is also easy to adapt the enumeration and the sampling procedure for multi-graphs with parallel edges and/or self-loops. The recursive formulas have a form that allows to formulate them with equations between the corresponding generating functions. It is sometimes possible to solve these equations and obtain closed formulas or asymptotic estimates from the solutions. However, due to the large number of parameters needed in the decomposition, it will not be easy to handle these equations. In the simpler case of labeled planar graphs the equations recently lead to asymptotic expressions for the number of labeled planar graphs [16].

References

1. 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.
2. A. Bender, Z. Gao, and N. Wormald. The number of labeled 2-connected planar graphs. *Electronic Journal of Combinatorics*, 9(43), 2002.
3. E. A. Bender and N. Wormald. Almost all convex polyhedra are asymmetric. *Can. J. Math.*, 27(5):854–871, 1985.

4. M. Bodirsky, O. Giménez, M. Kang, and M. Noy. On the number of series-parallel and outerplanar graphs. In *the Proceedings of European Conference on Combinatorics, Graph Theory, and Applications (EuroComb 2005)*, DMTCS Proceedings Series Volume AE, pages 383 – 388, 2005.
5. M. Bodirsky, C. Gröpl, D. Johannsen, and 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.
6. M. Bodirsky, C. Gröpl, and M. Kang. Sampling unlabeled biconnected planar graphs. Full version, available at `http://www.informatik.hu-berlin.de/Forschung_Lehre/algorithmen/en/forschung/planar/`.
7. M. Bodirsky, C. Gröpl, and M. Kang. Generating labeled planar graphs uniformly at random. In *Thirtieth International Colloquium on Automata, Languages and Programming (ICALP'03)*, pages 1095–1107, 2003.
8. M. Bodirsky and M. Kang. Generating outerplanar graphs uniformly at random. Accepted for publication in Combinatorics, Probability and Computing. Presented at the 1st workshop on Algorithms for Listing, Counting, and Enumeration (ALICE 03), 2003.
9. 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.
10. A. Denise, M. Vasconcellos, and D. Welsh. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.
11. A. Denise and P. Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. *Theoretical Computer Science*, 218:233–248, 1999.
12. R. Diestel. *Graph Theory*. Springer–Verlag, New York, 1997.
13. D. P. Eric Fusy and G. Schaeffer. Dissections and trees: applications to optimal mesh encoding and random sampling. In *Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms (SODA05)*, pages 690 – 699, 2005.
14. P. Flajolet, P. Zimmerman, and B. Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1-2):1–35, 1994.
15. S. Gerke and C. McDiarmid. On the number of edges in random planar graphs. *Comb. Prob. and Computing*, 13:358–402, 2004.
16. O. Giménez and M. Noy. Asymptotic enumeration and limit laws of planar graphs. preprint.
17. C. M. Hoffmann. *Group-Theoretic Algorithms and Graph Isomorphism*. Springer-Verlag, New York NY, 1982.
18. J. E. Hopcroft and J. K. Wong. Linear time algorithm for isomorphism of planar graphs. *Annual ACM Symposium on Theory of Computing*, pages 172–184, 1974.
19. C. McDiarmid, A. Steger, and D. Welsh. Random planar graphs. *Journal of Combinatorial Theory, Series B*, 93:187–205, 2005.
20. A. Nijenhuis and H. Wilf. *Combinatorial algorithms*. Academic Press Inc., 1979.
21. D. Osthus, H. J. Prömel, and A. Taraz. On random planar graphs, the number of planar graphs and their triangulations. *J. Combinatorial Theory, Series B*, pages 119–143, 2003.
22. G. Schaeffer. Random sampling of large planar maps and convex polyhedra. In *Proc. of the Thirty-first Annual ACM Symposium on the Theory of Computing (STOC'99)*, pages 760–769, Atlanta, Georgia, May 1999.
23. E. Steinitz. Polyeder und Raumeinteilungen. *Encyclopädie der mathematischen Wissenschaften*, Band III(9), 1922.
24. B. A. Trakhtenbrot. Towards a theory of non-repeating contact schemes. *Trudi Mat. Inst. Akad. Nauk SSSR*, 51:226–269, 1958. [In Russian].
25. W. T. Tutte. *Graph Theory*. Cambridge University Press, 1984.
26. T. Walsh. Counting labelled three-connected and homeomorphically irreducible two-connected graphs. *J. Combin. Theory*, 32:1–11, 1982.
27. T. Walsh and V. A. Liskovets. Ten steps to counting planar graphs. In *18th Southeastern International Conference on Combinatorics, Graph Theory, and Computing, Congr. Numer.*, volume 60, pages 269–277, 1987.