Dissections and trees, with applications to optimal mesh encoding and to random sampling

Éric Fusy[*] Dominique Poulalhon[†] Gilles Schaeffer[*]

Abstract

We present a bijection between some quadrangular dissections of an hexagon and unrooted binary trees. This correspondence has interesting consequences for enumeration, mesh compression and random graph sampling.

It yields a succinct representation for the set $\mathcal{P}(n)$ of n-edge 3-connected planar graphs matching the entropy bound $\frac{1}{n}\log|\mathcal{P}(n)| = 2+o(1)$ bits per edge. This solves a theoretical problem in mesh compression, as these graphs abstract the combinatorial part of meshes with spherical topology.

Once the entropy bound is matched, the guaranteed compression rate can only be improved on subclasses: we achieve the optimal parametric rate $\frac{1}{n}\log|\mathcal{P}(n,i,j)|$ bits per edge for graphs of $\mathcal{P}(n)$ with i vertices and j faces. This effectively reduces the entropy as soon as $|i-j| \gg n^{1/2}$, and achieves the optimal rate for triangulations.

It also yields an efficient uniform random sampler for labeled 3-connected planar graphs. Using it, the amortized complexity of sampling labeled planar graphs is reduced from the best previously known $O(n^{6.5})$ to $O(n^3)$.

1 Introduction

One origin of this work can be traced back to an article of Ed Bender in the American Mathematical Monthly [3], where he asked for a simple explanation of the remarkable asymptotic formula

$$(1.1) \qquad |\mathcal{P}(n,i,j)| \sim \frac{1}{3^5 2^4 ijn}\binom{2i-2}{j+2}\binom{2j-2}{i+2}$$

for the cardinality of the set of 3-connected planar graphs[1] with i vertices, j faces and $n = i+j-2$ edges, n going to infinity. By a theorem of Whitney these graphs have essentially a unique embedding on the sphere up to homeomorphisms.

Graphs, dissections and trees Another well known property of 3-connected planar graphs with n edges is the fact that they are in direct one-to-one correspondence with dissections of the sphere into n quadrangles that have no non-facial 4-cycle faces. The heart of our paper lies in a further one-to-one correspondence.

Theorem 1.1. *There is a one-to-one correspondence between unrooted binary trees with n nodes and quadrangular dissections of an hexagon with n interior vertices and no non-facial 4-cycle.*

The application from binary trees to dissections, which we call *the closure*, is easily described and resembles constructions [27, 6, 25] that were recently proposed for simpler kinds of dissections. Conversely, recovering the tree from the dissection can be done in linear time by a traversal akin to Kant's canonical ordering [18, 9, 5, 8]. The proof that these algorithms are correct is rather sophisticated, relying on new properties of constrained orientations as used by Schnyder for triangulations [29] and by Felsner [11] for 3-connected planar graphs.

Theorem 1.1 leads directly to the implicit representation of the numbers $|P(n,i,j)|$ due to Mullin and Schellenberg and from which Formula 1.1 derives. It partially explains the combinatorics of the occurrence of the cross product of binomials, since these are typical of bicolored binary tree formulas.

Random sampling A first byproduct of Theorem 1.1 is an efficient uniform random sampler for labeled 3-connected planar graphs, that is an algorithm that given n – respectively, given (n,i,j) – outputs a random element of $\mathcal{P}(n)$ – respectively, of $\mathcal{P}(n,i,j)$ – with equal chances for all elements.

The uniform random generation of classes of dissections of the sphere like triangulations or 3-connected graphs was first considered in mathematical physics (see refs in [2, 25]), and various types of random planar graphs are commonly used for testing graph drawing algorithms (see [14]).

The best previously known algorithm [28] had expected complexity $O(n^{5/3})$ for $\mathcal{P}(n)$, and was much less efficient for $\mathcal{P}(n,i,j)$, having even exponential complexity for i/j or j/i tending to 2. In Section 7, we show that our generator for $\mathcal{P}(n)$ or $\mathcal{P}(n,i,j)$ performs in linear time except if i/j or j/i tends to 2 where it becomes at most quartic.

[*]LIX, École polytechnique, France. [Eric.Fusy, Gilles.Schaeffer]@lix.polytechnique.fr

[†]LIAFA, Université Paris 7, France. Dominique.Poulalhon@liafa.jussieu.fr

[1]Unless when explicitly mentioned, all graphs are unlabeled.

From the theoretical point of view, it is also desirable to work with the uniform distribution on planar graphs. However, random planar graphs appear to be challenging mathematical objects [24, 21]. A Markov chain converging to the uniform distribution on planar graphs with i vertices was given in [10], but it resists known approaches for perfect sampling [32], and has unknown mixing time. As opposed to this, a recursive scheme to sample planar graphs was proposed in [4], with amortized complexity $O(n^{6.5})$. This result is based on recursive sampling (aka the method of branching probabilities) [23, 13, 33]. Our new random generator for 3-connected planar graphs allows to reduce the amortized cost to $O(n^3)$.

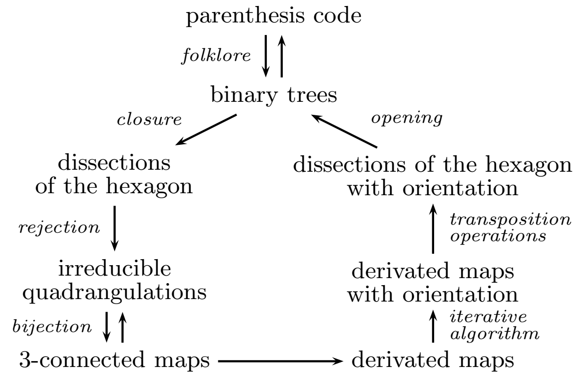


Figure 1: Relations between involved objects.

Succinct representations A second byproduct of Theorem 1.1 is the possibility to encode in linear time a 3-connected planar graph with n edges by a binary tree with n nodes. In turn the tree can be encoded by a balanced parenthesis word of $2n$ bits. This code is *optimal* in the information theoretic sense: the entropy per edge $\frac{1}{n}\log_2|\mathcal{P}(n)|$ of this class of graphs tends to 2 when n goes to infinity, so that a code for $\mathcal{P}(n)$ cannot give a better guarantee on the compression rate.

Applications calling for compact storage and fast transmission of 3D geometrical meshes have recently motivated a huge literature on compression, in particular for the combinatorial part of the meshes. The first compression algorithms dealt only with triangular faces [26, 30], but many meshes include larger faces, so that polygonal meshes have become prominent (see [1] for a recent survey).

The question of optimality of coders was raised in relation with exception codes produced by several successful heuristics when dealing with meshes with spherical topology [15, 19]. Since these meshes are exactly triangulations (for triangular meshes) and 3-connected planar graphs (for polyhedral ones), the coders of [25] and of the present paper respectively prove that traversal based algorithms can achieve optimality.

On the other hand, in the context of succinct data structures, almost optimal algorithms have been proposed [17, 20], that are based on separator theorems. However these algorithms are not truly optimal (they get ε close to the entropy but at the cost of an uncontrolled increase of the constants in the linear complexity). Moreover, although they rely on a sophisticated recursive structure, they do not support efficient adjacency requests.

As opposed to that, our algorithm shares with [16, 5] the fact that the code produced is essentially the code of a spanning tree. More precisely it is just the balanced parenthesis code of a binary tree, and adjacencies of the initial dissection that are not present in the tree can be recovered from the code by a simple variation on the interpretation of the symbols. It should thus be possible to deal with adjacency queries in time proportional to the degree of vertices using the approach of [22, 16].

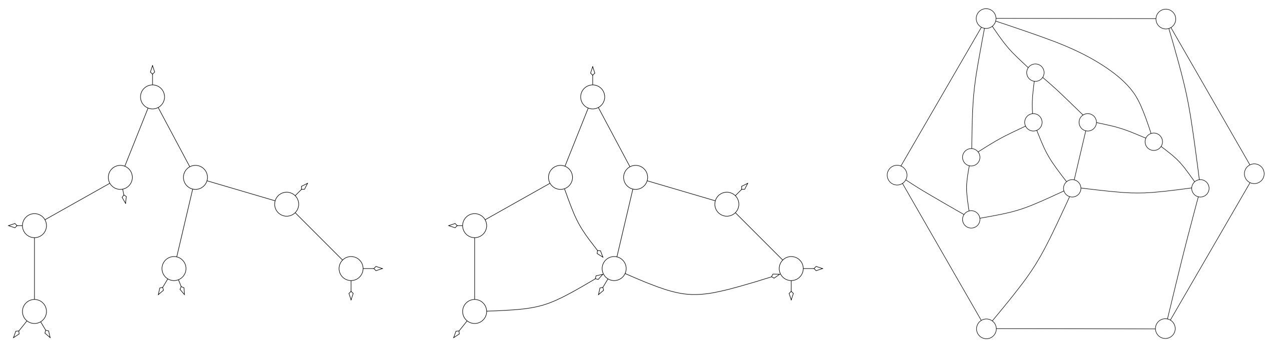
Finally we show that the code can be modified to be optimal on the class $\mathcal{P}(n,i,j)$. Since the entropy of this class is strictly smaller than that of $\mathcal{P}(n)$ as soon as $|i-n/2| \gg n^{1/2}$, the resulting parametric coder is more efficient in this range. In particular in the case $j = 2i-4$ our new algorithm specializes to an optimal coder for triangulations.

Outline of the paper We first present a bijective application between binary trees and some dissections of the hexagon by quadrangular faces; then we deduce from this bijection sampling algorithms for 3-connected planr maps and planar graphs, and finish with the description of a coding algorithm for 3-connected planar maps. The connections between the different families of objects we consider are shown on Figure 1.

2 Definitions

2.1 Planar maps A *planar map* is a proper embedding of a connected graph in the plane, where proper means that edges are smooth simple arcs which meet only at their endpoints. A planar map is said *rooted* if one edge of the outer face, called the *root-edge*, is marked and oriented such that the outer face is on the right of the edge. The origin of the root-edge is called *root-vertex*. Vertices are said *outer* or *inner* depending on whether they belong to the outer face or not.

A planar map is said *3-connected* if it can not be disconnected by the removal of two vertices. We denote by $\mathcal{P}'_n$ (resp. $\mathcal{P}'_{ij}$) the set of rooted 3-connected planar maps with n edges (resp.i vertices and j faces).



(a) A binary tree, (b) its partial closure, (c) and its complete closure.

Figure 2: The closure application.

2.2 Plane trees They are planar maps with a single face – the outer one. A vertex is called a *leaf* if it has degree 1, and *node* otherwise. Edges incident to a leaf are called *stems*, and the other *inner edges*. Observe that rooted plane trees are exactly usual trees, with the root-edge oriented from the usual root to its first son. It is convenient anyway to add a *root-leaf* "above" the root-vertex (for symmetry reasons in the case of regular trees). Hence in the following *rooted plane trees* are always rooted on a stem oriented toward its leaf.

Binary trees are plane trees whose nodes have degree 3. We denote respectively by $\mathcal{B}_n$ and $\mathcal{B}'_n$ the sets of binary and rooted binary trees with n nodes (and hence $(n+2)$ leaves by a recursive proof).

There is a unique bicoloration of the nodes of a rooted binary tree in black and white such that adjacent nodes have distinct colors and the node connected to the root-leaf is black. We denote by $\mathcal{B}_{ij}$ and $\mathcal{B}'_{ij}$ the set of (rooted) binary trees with i black nodes and j white nodes for this bicoloration (and hence $2i-j+1$ white leaves and $2j-i+1$ black ones by a recursive proof).

2.3 Quadrangulations and dissections A *quadrangulation* is a planar map whose faces (even the outer face) have degree 4. A *dissection of the hexagon by quadrangular faces* is a planar map whose outer face has degree 6 and inner faces have degree 4.

Any cycle of a map but the one that delimits the outer face is said *proper*. Proper cycles that do not delimit a face are said *separating*.

A quadrangulation or a dissection of the hexagon by quadrangular faces is said *irreducible* if it has no separating 4-cycle. We denote by $\mathcal{Q}_n$ ($\mathcal{Q}'_n$) the set of (rooted) irreducible quadrangulations with n faces (including the outer face, and hence $n+2$ vertices by Euler relation), and by $\mathcal{D}_n$ ($\mathcal{D}'_n$) the set of (rooted) irreducible dissections of the hexagon with n inner vertices (and hence $n+2$ quadrangular faces by Euler relation). From now on, irreducible dissections of the hexagon by quadrangular faces will be shortly called *irreducible dissections*.

As faces of dissections and quadrangulations have even degree, vertices of these maps can be bicolored (say, in black and white), such that each edge connects a black vertex to a white one, and such a bicoloration is unique up to the choice of the colors. We denote by $\mathcal{Q}'_{ij}$ the set of rooted bicolored irreducible quadrangulations with i black vertices and j white vertices and such that the root-vertex is black, and by $\mathcal{D}'_{ij}$ the set of rooted bicolored irreducible dissections with i black *inner* vertices and j white *inner* vertices and such that the root-vertex is black.

3 Closure application: from binary trees to irreducible dissections of the hexagon

3.1 Partial closure Let T be a binary tree. Let us define a *local closure* operation based on a counter-clockwise traversal of the contour: such a traversal walks alongside the edges, in an alternating series of stems and inner edges; if a stem is followed by three inner edges, its local closure consists in merging its leaf with the extremity of the third edge, so as to create (or *close*) a quadrangular face. The involved stem is then considered as an inner edge of the obtained map, on which further local closures may be performed greedily until no more is possible. It is easy to see that the final map, called *partial closure* of T, does not depend on the order of the local closures. An example is shown in Figure 2(b).

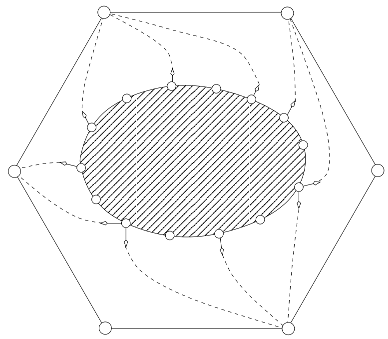


Figure 3: End of the complete closure when $r = s = 2$.

3.2 Complete closure We want here to complete the closure operation to obtain a dissection of the hexagon with quadrangular faces. Observe that a binary tree T with n nodes has $(n+2)$ leaves and $(n-1)$ inner edges. Hence there are $(n+2)$ stems and $(2n-2)$ sides of edges incident to the outer face at the beginning. As each local closure decreases by 1 the number of stems and by 2 the number of sides of inner edges which are incident to the outer face, if k denotes the number of unmatched stems in the partial closure of T, there are $(2k-6)$ sides of inner edges incident to the outer face. Moreover, stems delimit intervals of inner edges on the contour, with length at most 2, otherwise a local closure would be possible. Let r be the number of such intervals of length 1 and s be the number of intervals of length 0 (that is, the number of nodes incident to two unmatched stems). Then r and s are clearly related by the relation $r + 2s = 6$.

The *complete closure* consists in attaching all unmatched stems to vertices of the hexagon such that each pair of consecutive stems creates a quadrangular face. As illustrated by Figure 3, there is a unique way to do this up to rotation of the hexagon. An example is given on Figure 2(c).

LEMMA 3.1. *The closure of a binary tree is an irreducible dissection of the hexagon.*

Proof. Omitted. □

4 Tri-orientations

4.1 Definitions We define a kind of generalized orientation on maps, in which any edge may be non-oriented, or (*simply*) oriented in either direction, or even oriented in both directions (*bi-oriented edges*). If a map is endowed with such a generalized orientation, the *outdegree* of a vertex v is defined as the number of its incident edges that are either bi-oriented or simply oriented outward v.

The (unique) *tri-orientation* of a binary tree is defined as the generalized orientation in which inner edges are bi-oriented, and stems are simply oriented toward their leaf (see Figure 4(a) for an example). Hence each node has outdegree 3.

A *tri-orientation* of a dissection is a generalized orientation in which edges of the hexagon are not oriented and inner edges are either simply or bi-oriented in such a way that outer and inner vertices have respectively outdegree 0 and 3.

Let D be a dissection endowed with a tri-orientation. A *clockwise cycle* of D is a closed curve $\mathcal{C}$ consisting of edges of D that are either bi-oriented or simply oriented with the interior of $\mathcal{C}$ on their right.

PROPOSITION 4.1.
- *A tri-orientation of an element of $\mathcal{D}_n$ has $(n-1)$ bi-oriented edges and $(n+2)$ simply oriented edges.*
- *If a tri-orientation of a dissection has no clockwise cycle, then its bi-oriented edges form a spanning tree of the inner vertices.*

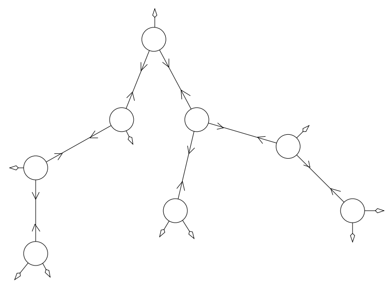
Proof. Let $D \in \mathcal{D}_n$ endowed with a tri-orientation, and let s and r denote the respective numbers of its simply and bi-oriented edges. Euler characteristic and the degree of the faces of the dissection ensure that the dissection has $2n+1$ inner edges. Hence $2n+1 = r+s$. Moreover, as all inner vertices have outdegree 3 for the tri-orientation, $3n = 2r + s$. Hence $r = n-1$ and $s = n+2$.

The subgraph T consisting of the bi-oriented edges and their extremities has $r = n-1$ edges, no cycle (otherwise such a cycle could be traversed clockwise as all its edges are bi-oriented), and at most n vertices since outer vertices can not belong to T. Hence T is a tree that spans exactly the n inner vertices. □

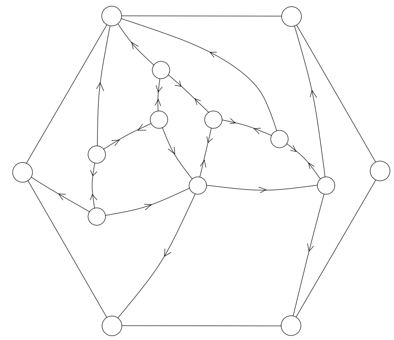
4.2 Closure-tri-orientation of a closure-dissection of the hexagon A dissection of the hexagon that can be obtained as the closure of a binary tree will be called a *closure-dissection*. Let D be such a closure-dissection, obtained by the closure of a binary tree T. The tri-orientation of T clearly induces *via* the closure operation a tri-orientation of D, called a *closure-tri-orientation*. On this tri-orientation, bi-oriented edges correspond to inner edges of the original binary tree, see Figure 4(b).

LEMMA 4.1. *There is no clockwise cycle in a closure-tri-orientation of a closure-dissection.*

Proof. No clockwise cycle can be created during the completion of the closure (illustrated in Figure 3), since extremities of stems are merged during this step with



(a) A binary tree with its tri-orientation,



(b) and the closure-tri-orientation of its closure.

Figure 4: Examples of tri-orientations.

vertices of the hexagon, which have outdegree 0 in the tri-orientation, and thus can not belong to any clockwise cycle. Hence clockwise cycles may only have been created during a local closure. However such closure edges have the outer face on their right, which contradicts the orientation of the cycle. □

5 The opening application

Proposition 4.1 and Section 4.2 give all necessary elements to describe the inverse application of the closure, which is called the *opening application*: let D be a closure-dissection endowed with its closure-tri-orientation; then the only binary tree whose closure is D is obtained by removing edges and vertices of the hexagon and detaching all simply oriented edges at their extremity.

As the closure-tri-orientation has no clockwise cycle, we define the general opening application of an irreducible dissection D as follows:

- endow D with a tri-orientation without clockwise cycle;
- remove the vertices and edges of the hexagon;
- disconnect all simply oriented edges of the tri-orientation at their extremity.

In order for this application to be correctly defined, we need the following theorem:

THEOREM 5.1. *Any irreducible dissection has a unique tri-orientation without clockwise cycle.*

Proof. Omitted. (See however Remark 9.1.) □

6 Statement of the main theorem

THEOREM 6.1. *The closure application is a bijection between the set $\mathcal{B}_n$ of binary trees with n nodes and the set $\mathcal{D}_n$ of irreducible dissections with n inner vertices. The inverse of the closure application is the opening application.*

Proof. Omitted. □

We can state two analogous versions of Theorem 6.1 for rooted objects:

THEOREM 6.2. *The closure application allows to define the following bijections between sets of rooted objects:*

$$\begin{aligned} \mathcal{B}'_n \times \{1,\dots,6\} &\equiv \mathcal{D}'_n \times \{1,\dots,n+2\}, \\ \mathcal{B}'_{ij} \times \{1,2,3\} &\equiv \mathcal{D}'_{ij} \times \{1,\dots,2i-j+1\}. \end{aligned}$$

Proof. The first point can easily be proved using Theorem 6.1 and the fact that a binary tree of $\mathcal{B}_n$ has $(n+2)$ leaves and a dissection of $\mathcal{D}_n$ has 6 edges to carry the root. The proof of the second point is similar. □

$|\mathcal{B}'_n|$ is well-known to be the nth Catalan number $\frac{1}{n+1}\binom{2n}{n}$, and refinements of the standard proofs yield $|\mathcal{B}'_{ij}| = \frac{1}{2j+1}\binom{2j+1}{i}\binom{2i}{j}$. Theorem 6.2 thus implies the following enumerative results:

COROLLARY 6.1.

$$|\mathcal{D}'_n| = \frac{6}{(n+2)(n+1)}\binom{2n}{n},$$

and

$$|\mathcal{D}'_{ij}| = \frac{3}{(2i+1)(2j+1)}\binom{2j+1}{i}\binom{2i+1}{j}.$$

7 Counting and sampling rooted 3-connected maps

7.1 Bijection between 3-connected maps and irreducible quadrangulations There is a well known application, which we shall refer to as Tutte's application, between $\mathcal{Q}'_n$ and $\mathcal{P}'_n$ (resp. between $\mathcal{Q}'_{ij}$ and $\mathcal{P}'_{ij}$): given a rooted quadrangulation $Q \in \mathcal{Q}'_n$ endowed with its bicoloration of vertices, we obtain a rooted map M by linking, for each face f of Q (even the outer face), the two diagonally opposed black vertices of f. M is canonically rooted on the edge corresponding to the outer face of Q with same root-vertex as Q.

This application is easily invertible: given a rooted map M with (black) vertices, it consists in adding a (white) vertex called a face-vertex in each face (even the outer face) of M and linking a vertex v and a face-vertex v_f by an edge if v is incident to the face f corresponding to v_f. By keeping only the edges of incidence face-vertex, we obtain a quadrangulation. We choose the root as the edge which follows the root of M in the counter-clockwise order around the origin of the root of M.

The following theorem is a classical result in the theory of maps.

THEOREM 7.1. *Tutte's application realizes a bijection between $\mathcal{P}'_n$ and $\mathcal{Q}'_n$, and between $\mathcal{P}'_{ij}$ and $\mathcal{Q}'_{ij}$.*

7.2 Injection of rooted irreducible quadrangulations in rooted irreducible dissections of the hexagon We can easily associate to a rooted irreducible quadrangulation Q a rooted dissection D of the hexagon in the following way: remove the root-edge of Q (the outer face becomes thus hexagonal) and carry the root on the edge of the hexagon that has the same origin as the origin of the root of Q and the outer face on its right.

It is easy to see that D is an irreducible dissection of the hexagon. Indeed the presence of a separating 4-cycle in D would clearly imply the presence of a separating 4-cycle in the quadrangulation Q from which we have removed the root edge.

Moreover it is immediate that this application is an injection (but not a bijection). More precisely, it is an injection from $\mathcal{Q}'_n$ to $\mathcal{D}'_{n-4}$, and from $\mathcal{Q}'_{ij}$ to $\mathcal{D}'_{i-3,j-3}$.

7.3 Algorithm to sample rooted 3-connected maps We deduce from the first application of Theorem 6.2 the following algorithm:

- sample an object T of $\mathcal{B}'_{n-4}$;
- perform the closure of T to obtain an irreducible dissection $D \in \mathcal{D}_{n-4}$;
- choose randomly the root-vertex v on the hexagon;
- add a root-edge e in the outer face of D going, with the outer face on its right, from v to the vertex diametrically opposed to v on the hexagon[2]; the obtained figure is a rooted quadrangulation Q with n faces;
- if Q is irreducible, return the rooted 3-connected map in $\mathcal{P}'_n$ associated to Q by Tutte's application; otherwise reject.

PROPOSITION 7.1. *We have the asymptotic result:*

$$\frac{|\mathcal{P}'_n|}{|\mathcal{D}'_{n-4}|} \sim \frac{2^8}{3^6}.$$

Hence the distribution for the number of rejects is a geometric law whose mean is $c = \frac{3^6}{2^8}$. Hence, as the closure application has a linear-time complexity, the sampling algorithm has a linear-time complexity.

Proof. We have obtained in section 6 the enumerative result $|\mathcal{D}'_n| = \frac{6}{n+2}|\mathcal{B}'_n| = \frac{6(2n)!}{(n+2)!n!}$. Using Stirling formula, we obtain $|\mathcal{D}'_{n-4}| \sim \frac{3}{128\sqrt{\pi}}\frac{4^n}{n^{5/2}}$. Moreover, according to [31], $|\mathcal{P}'_n| \sim \frac{2}{3^5\sqrt{\pi}}\frac{4^n}{n^{5/2}}$. □

Using the second application of Theorem 6.2, there is a similar algorithm to sample rooted 3-connected maps with i vertices and j faces. Using the asymptotic formula $|\mathcal{P}'_{ij}| \sim \frac{1}{3^5 2^2 ijn}\binom{2i-2}{j+2}\binom{2j-2}{i+2}$ and Corollary 6.1, it is easy to show that this algorithm has a linear-time complexity when the quantity $\alpha = i/j$ belongs to a closed interval included in $]1/2, 2[$. In addition, in the worst case of triangulations where $j = 2i - 4$, the time-complexity of this sampling algorithm is quartic.

7.4 The enumeration of rooted 3-connected maps In order to count rooted 3-connected maps, and ultimately to recover the asymptotic formula for $|\mathcal{P}'_{ij}|$, let us return to the fact that the injection of Section 7.2 is not a bijection. Observe that an irreducible dissection belongs to the image of this injection if and only if it does no contain a path of length 3 connecting the root vertex and the opposite vertex of the hexagon via an internal vertex. Such a dissection is called an *undecomposable* dissection.

The key of the enumeration is that any irreducible dissection can essentially be decomposed along the above mentioned paths of length 3 into undecomposable dissections. The analysis of this decomposition, as presented in the full paper, allows to give a simple expression of the generating function of undecomposable dissections (and hence of 3-connected planar maps) in

[2] This operation is the inverse of the injection of section 7.2.

terms of the generating function of all dissections of the hexagon (and hence in terms of the generating functions of binary trees).

This allows to give a derivation of Mullin and Schellenberg generating function expression for 3-connected planar maps. It explains the occurrence of binary trees generating functions and thus the form of the binomial coefficients in the asymptotic formula for $|\mathcal{P}'_{ij}|$.

8 Random planar graphs

The algorithm of [4] to generate labeled planar graphs is based on the decomposition of graphs into a tree-like structure of 3-connected components. In other terms there is a decomposition of any planar graph p into a pair $(t, (a_1, \ldots, a_\ell))$, where each a_k is a rooted 3-connected planar graph (*i.e.* the shape of the kth component) and t is the tree-like structure (*i.e.* the necessary information to glue components and put labels). A fundamental property of the uniform distribution on planar graphs is that given t (and in particular the numbers of vertices i_k and edges n_k in each connected components), the $(a_k)_k$ are independent r.v. and a_k is uniformly distributed on $\mathcal{P}(n_k, i_k, j_k)$.

The algorithm of [4] then essentially consists into generating the tree-like structure with the right probability, and then generating random 3-connected components with an external generator as [28]. The generation of the tree-like structure can be done by the recursive method [23] because all branching probabilities can be explicitly computed. The approach used in [4] consists in precomputing all the $O(n^4)$ entries involved in the branching probabilities by direct dynamical programming, *i.e.* applying convolutive recurrences, for a total cost of $O(n^6)$ arithmetic multiplications on numbers with $O(n \log n)$ bits.

Once the precomputation is done, the cost of generating the tree-like shape is small in front of the cost of generating the 3-connected graphs. In [4], the generation is done using the random sampler proposed in [28], and the cost of this part of the generation was estimated to $O(n^{13/2})$. However this analysis fails for extremal ranges of values, due to uniformity assumptions stated in the complexity results of [28], and ignored in [4]. For instance generating connected planar graphs with $2n$ edges and $i = n + 2$ vertices amounts to generating triangulations, which are 3-connected. In this case the whole process boils down to one call of the generator of [28], that has exponential complexity when used naively in this range. It is unclear whether this generator can be fine tuned to remain effective uniformly on the whole range needed for this application.

On the other hand, the new generator that we propose here allows both to correct this problem and to improve the complexity. Indeed, as we have seen in Section 7.3, the sampler appears to have quartic complexity in the worst case, and it is called to generate a linear number of edges. The final complexity of the generation is thus $O(n^3)$.

9 Optimal coding of 3-connected planar maps

9.1 Introduction We shall describe an algorithm to encode a 3-connected planar map whose outer face is triangular. Indeed, it can also be used to encode a general 3-connected planar map G: if the outer face of G is not triangular, it suffices to fix three consecutive vertices a_2, a_1 and a_3 on the outer face of G and link a_2 and a_3 by an edge to obtain a 3-connected planar map $\widetilde{G}$ whose outer face is triangular. Then the coding of G is obtained as the coding of $\widetilde{G}$ and a bit indicating that we had to add an edge.

To encode a 3-connected map G, the idea of the algorithm is first to compute a particular orientation, called minimal α_0-orientation, of the edges of the so-called derivated map G' of G. Then, using a simple application, we can associate an irreducible dissection D to G'. In addition, the minimal α_0-orientation of G' can be transposed in a tri-orientation of D without clockwise cycle. Finally, using the computed tri-orientation of D without clockwise cycle, it just remains to perform the opening of D and encode the obtained binary tree.

9.2 Derivated map and α-orientation Let G be a 3-connected planar map whose outer face is triangular. The *derivated map* G' of G is obtained by superposing G and its dual map G^* and by removing the dual vertex v^*_∞ corresponding to the outer face of G. Moreover, we add to each of the three border vertices a_1, a_2 and a_3 of G an half-edge directed toward the outer face.

The map G (resp. G^*) is called the *primal map* (resp. the *dual map*) of G'. To each edge e of G corresponds an *edge-vertex* of G' of degree 4: this edge-vertex is the intersection of e and its dual edge e^* in G^*. In addition, e and e^* respectively yield two edges in G'. An example of a derivated map can be seen on Figure 6(a), where the edges of G are darker than the edges of G^*.

In [12], Felsner introduces the notion of α-orientation. Let $G = (V, E)$ be a planar map and $\alpha : V \to \mathbb{N}$. An α-orientation of G is an orientation of the edges of G such that $\forall v \in V$, there are $\alpha(v)$ edges incident to v which are directed outward. If there exists an orientation of the edges of G which is an α-orientation, then the function α is said feasible in G. Felsner obtains in [12] the following results:

THEOREM 9.1. • *For a planar map G and a feasible* *function α, there exists a unique α-orientation of G which does not have any clockwise cycle. This α-orientation is said minimal.*

- *For the derivated map of a 3-connected planar map whose outer face is triangular, the function α_0 such that $\alpha_0(v) = 3$ for all primal and dual vertices and $\alpha_0(v) = 1$ for all edge-vertices is a feasible function.*

9.3 An algorithm to compute the minimal α_0-orientation of the derivated map

Introduction To compute the minimal α_0-orientation of G', we present an iterative linear-time algorithm which is very similar to Kant's algorithm of right-most canonical ordering. Our algorithm also generalizes an algorithm computing the Schnyder Woods of a triangulation presented in [7].

The idea is to maintain at each step k a simple cycle $\mathcal{C}_k$ of edges of G such that:

- the edge (a_2, a_3) is on $\mathcal{C}_k$;
- for each edge e of G outside $\mathcal{C}_k$, the four edges of G' incident to the edge-vertex v_e associated to e have been oriented at a step $j < k$ and v_e has outdegree 1;
- all other edges of G' are not oriented yet.

Let G_k denote the submap of G obtained by removing all vertices and edges outside of $\mathcal{C}_k$. In addition, we order the vertices of the cycle $\mathcal{C}_k$ from left to right, i.e. by doing a traversal of $\mathcal{C}_k$ from a_3 to a_2 without using the edge (a_2, a_3).

At each step k of the algorithm, the operation will consist in removing some vertices of $\mathcal{C}_k$ and their incident edges. This has the effect of merging the incident bounded faces with the outer face of G_k. For each removed edge e, we will also orient the 4 incident edges in G' of its associated edge-vertex v_e.

For this to produce a next planar map G_{k+1} such that the contour $\mathcal{C}_{k+1}$ of G_{k+1} is still a simple cycle containing (a_2, a_3), we need to carefully choose some vertices of $\mathcal{C}_k$: the proper way to do this is explained in the rest of this section.

Definitions A vertex of $\mathcal{C}_k$ is said *active* if it is incident to at least one edge of $G \backslash G_k$. Otherwise, the vertex is said *passive*. By convention, before the first step of the algorithm, the vertex a_1 is considered as active.

For each couple of vertices (v_1, v_2) of $\mathcal{C}_k$ ordered such that v_1 is on the left of v_2 on the contour of $\mathcal{C}_k$, the border path on $\mathcal{C}_k$ to go from v_1 to v_2 without passing by the edge (a_2, a_3) is denoted by $[v_1, v_2]$. We also write $]v_1, v_2[= [v_1, v_2] \backslash \{v_1, v_2\}$

A couple (v_1, v_2) of vertices of $\mathcal{C}_k$ is said *separating* if there exists an inner face f of G_k such that v_1 and v_2 are incident to f but the edges of $[v_1, v_2]$ are not all

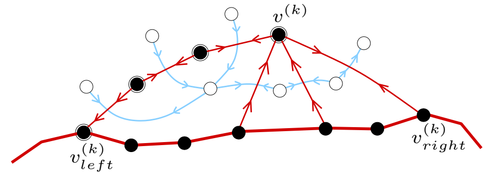


Figure 5: The operations performed at step k of the algorithm, when $v^{(k)}_{left}$ is active and $v^{(k)}_{right}$ is passive.

incident to f. Such a face is called a *separating face* and the triple (v_1, v_2, f) is called a *separator*.

A vertex v on $\mathcal{C}_k$ is said *blocked* if it is incident to a separating face of G_k. By convention the vertices a_2 and a_3 are always considered as blocked.

A vertex v on $\mathcal{C}_k$ is said *eligible* if it is active and not blocked.

Finally, for each vertex v of $\mathcal{C}_k$, we define its *left-connection vertex* v_{left} as the leftmost vertex on $\mathcal{C}_k$ such that the vertices of $]v_{left}, v[$ all have only two incident edges in G_k. The path $[v_{left}, v]$ is called the *left-chain* of v and the first edge of $[v_{left}, v]$ is called the *left-connection edge* of v. Similarly, we define the *right-connection vertex*, the *right-chain*, and the *right-connection edge* of v (see Figure 5).

Operations of step k First we choose the rightmost eligible vertex of $\mathcal{C}_k$ and we call $v^{(k)}$ this vertex [3]. Remark that this eligible vertex can not be a_2 or a_3 because a_2 and a_3 are blocked.

Let $f_1, \ldots, f_m$ be the bounded faces of G_k incident to $v^{(k)}$ from right to left. Let also $e_1, \ldots, e_{m+1}$ be the edges of G_k incident to $v^{(k)}$ from right to left. Hence, $\forall 1 \leq i \leq m$, f_i corresponds to the sector between e_i and e_{i+1}.

An important remark is that the right-chain of $v^{(k)}$ is reduced to one edge, which is easy to show using the fact that $v^{(k)}$ is the rightmost eligible vertex. Another remark is that all vertices of $]v^{(k)}_{left}, v^{(k)}[$ are active because each vertex of a 3-connected map has at least degree 3.

We perform the following operations: for each edge e belonging to the left-chain of $v^{(k)}$ or belonging to $\{e_1, \ldots e_m\}$, let v_e be its associated edge-vertex in G'. Then we choose an edge e' incident to v_e and orient e' outward of v_e. We orient the three other edge incident to v_e toward v_e, so that the edge-vertex v_e has outdegree 1. The choice of the outgoing edge is made as follows, see Figure 5, where the active vertices are surrounded:

- if e is an edge of the left-chain different from the

[3] We will see in Lemma 9.1 that there always exists an eligible vertex on $\mathcal{C}_k$ as long as G_k is not reduced to the edge (a_2, a_3).

left-connection edge, choose the edge connecting v_e to the dual vertex associated to f_m;

- if e is the left-connection edge, two cases can arise; if $v_{left}^{(k)}$ is passive, choose the outgoing edge of v_e as the edge going toward the exterior of $\mathcal{C}_k$; if $v_{left}^{(k)}$ is active, choose the outgoing edge of v_e as the edge going to $v_{left}^{(k)}$;
- similarly, if $e = e_1$ (that is the right-connection edge), two cases can arise; if $v_{right}^{(k)}$ is passive, choose the outgoing edge of v_e as the edge going toward the exterior of $\mathcal{C}_k$; if $v_{right}^{(k)}$ is active, choose the outgoing edge of v_e as the edge going to $v_{right}^{(k)}$.

Finally the graph G_{k+1} is obtained from G_k by removing the edges and vertices of the left-chain except $v_{left}^{(k)}$, and by removing the edges $e_1, \ldots, e_m$.

As a_2 and a_3 are blocked on $\mathcal{C}_k$, the contour of G_{k+1} still contains the edge (a_3, a_2).

In addition, if G_{k+1} is not reduced to (a_2, a_3), the 3-connectivity of G and the fact that the chosen vertex $v^{(k)}$ is not incident to any separating face can be used to prove easily that the contour $\mathcal{C}_{k+1}$ of G_{k+1} is a simple closed curve, i.e. it does not contain any separating vertex.

By construction, the three conditions stated in the introduction of Section 9.3 are still satisfied by $\mathcal{C}_{k+1}$.

The following lemma ensures that the algorithm terminates:

LEMMA 9.1. *There exists an eligible vertex on $\mathcal{C}_k$.*

Proof. Omitted (recursive proof based on the 3-connectivity of G). □

Last step of the algorithm Lemma 9.1 implies that there remains just the edge (a_2, a_3) at the end of the iteration. To conclude the construction, we bi-orient the edge (a_2, a_3), and the outgoing edge of the associated edge-vertex is chosen to be its incident edge directed toward the outer face.

The six first drawings of Figure 6 show the execution of the algorithm of orientation on an example. At each step the edges of $\mathcal{C}_k$ are black and wider, the active vertices on the border are circled and the next chosen vertex doubly circled. This algorithm is similar to the algorithm of [18], which can be implemented with a linear-time complexity. The adaptation of these methods of implementation ensures that our algorithm has also a linear-time complexity.

THEOREM 9.2. *The algorithm computes the minimal α_0-orientation of the derivated map.*

Proof. Omitted. □

9.4 Encode the 3-connected map using the minimal α_0-orientation of its derivated map

Associate an irreducible dissection to the derivated map We can associate to the derivated map G' a dissection D of the hexagon with quadrangular faces.

The vertices of D are the primal and dual vertices of G'. The edges of D are obtained as follows: each inner face f of G' is quadrangular and has two diagonally opposed edge-vertices and has a primal and a dual vertex which are diagonally opposed.

The edge of D associated to f is obtained by linking the primal vertex and the dual vertex of f by a new edge, see Figure 6(g), where the derivated map is superposed with its associated dissection of the hexagon (ignore the orientations here). Using 3-connectivity of G, it is easy to prove that D is irreducible.

Induce a tri-orientation of the irreducible dissection from the minimal α_0-orientation of the derivated map Given the minimal α_0-orientation X_0 of the derivated map G', we can associate to X_0 a tri-orientation of the associated dissection D of the hexagon in the following way, see Figure 6(g). For each face f of G', noting e_f its associated edge on the dissection, we distinguish the following cases:

- if e_f is incident to a vertex v of the outer face of the dissection, we orient e_f toward v;
- otherwise, let v (resp. v^*) be the primal (resp. dual) vertex of f; let e'_v (resp. e'_{v^*}) be the edge of f which has v (resp. v^*) as origin when we traverse the contour of f with the interior of f on our left; as we work with the α_0-orientation without clockwise cycle, only three cases can arise:
  - e'_v and e'_{v^*} are respectively directed outward of v and v^*; then we bi-orient e_f, i.e. we orient outward of their origin the two half-edges which form e_f;
  - e'_v is oriented outward of v and e'_{v^*} is directed toward v^*; then we orient e_f toward v^*;
  - e'_v is oriented toward v and e'_{v^*} is directed outward of v^*; then we orient e_f toward v.

LEMMA 9.2. *The orientation of the dissection obtained from the minimal α_0-orientation of the derivated map is a tri-orientation without clockwise cycle.*

Proof. Omitted. □

REMARK 9.1. *This lemma together with Theorem 9.1 are the main tools of the proof of the existence in Theorem 5.1.*

Open the dissection in a binary tree Having computed the tri-orientation without clockwise cycle of

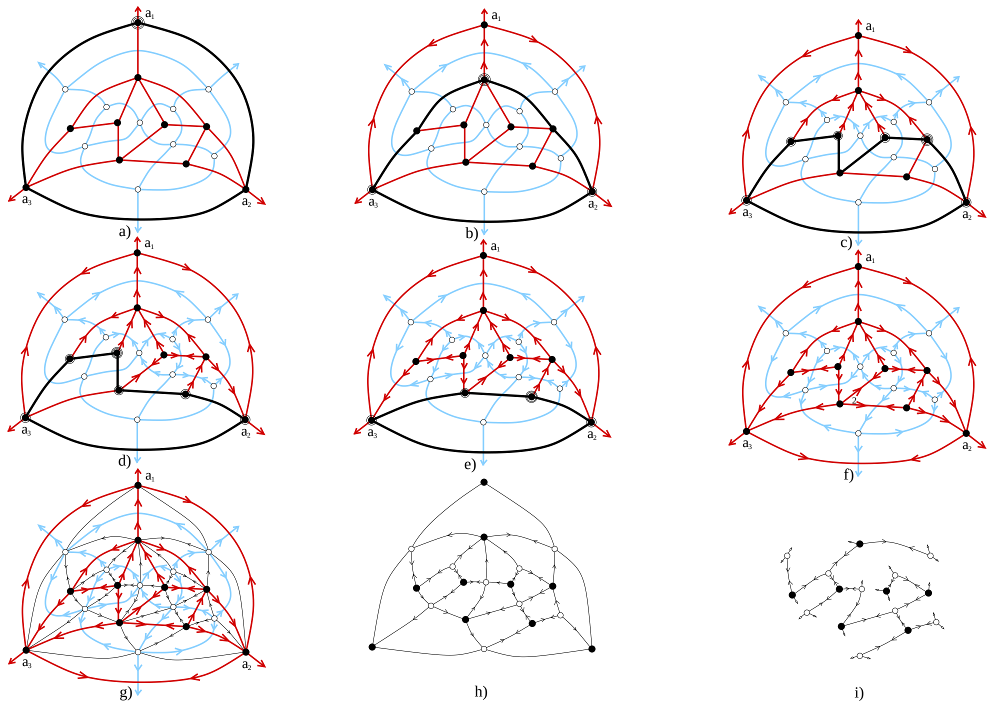


Figure 6: A complete example of the encoding of a 3-connected planar map.

the dissection D, we can perform the opening of D as indicated in Section 5. We obtain thus a binary tree T, see Figure 6(i). It is easy to show that T has always $n-5$ nodes if G has n edges. Similarly, if G has i vertices and j faces, then the bicoloration of T has always $i-3$ black vertices and $j-3$ white vertices or $i-3$ black vertices and $j-3$ white vertices (depending on the choice of the leaf to place a root).

In addition, the set of binary trees with n edges and the set of binary trees with i black vertices and j white vertices can both be easily optimally encoded using parenthesis words.

As $\frac{|\mathcal{B}'_n|}{|\mathcal{P}'_n|}$ is bounded by a constant and $\frac{|\mathcal{B}'_{ij}|}{|\mathcal{P}'_{ij}|}$ is bounded by a fixed (quadratic) polynomial, the asymptotic entropy of $\mathcal{B}'_n$ and $\mathcal{P}'_n$ (resp. of $\mathcal{B}'_{ij}$ and $\mathcal{P}'_{ij}$) are equal. Hence the encoding of $\mathcal{P}'_n$ (resp. of $\mathcal{P}'_{ij}$) is optimal.

Acknowledgments N. Bonichon and L. Castelli Aleardi are thanked for fruitful discussions.

References

[1] P. Alliez and C. Gotsman. Recent advances in compression of 3D meshes. In *Proceedings of the Symposium on Multiresolution in Geometric Modeling*, Cambridge, 2003. Available on `http://www.cs.technion.ac.il/~gotsman/`.

[2] J. Ambjørn, P. Białas, Z. Burda, J. Jurkiewicz, and B. Petersson. Effective sampling of random surfaces by baby universe surgery. *Phys. Lett. B*, 325:337–346, 1994.

[3] E. A. Bender. The number of three-dimensional convex polyhedra. *Amer. Math. Monthly*, 94(1):7–21, 1987.

[4] M. Bodirsky, C. Gröpl, and M. Kang. Generating labeled planar graphs uniformly at random. In *ICALP*, pp. 1095–1107, 2003.

[5] N. Bonichon, C. Gavoille, and N. Hanusse. An information-theoretic upper bound of planar graphs using triangulation. In *STACS*, pp. 499–510, 2003.

[6] J. Bouttier, P. di Francesco, and E. Guitter. Census of planar maps: from the one-matrix solution to a combinatorial proof. *Nucl. Phys. B*, 645:477–499, 2002.

[7] E. Brehm. 3-orientations and Schnyder 3-tree-decompositions. `http://page.mi.fu-berlin.de/~felsner/Diplomarbeiten/brehm.ps.gz`.

[8] L. Castelli-Aleardi and O. Devillers. Canonical triangulation of a graph, with a coding application. Preprint, 2004.

[9] R. C.-N. Chuang, A. Garg, X. He, M.-Y. Kao, and H.-I. Lu. Compact encodings of planar graphs via canonical orderings. In *ICALP*, pp. 118–129, 1998.

[10] A. Denise, M. Vasconcellos, and D. J. A. Welsh. The random planar graph. *Congr. Numer.*, 113:61–79, 1996. Festschrift for C. St. J. A. Nash-Williams.

[11] S. Felsner. Convex drawings of planar graphs and the order dimension of 3-polytopes. *Order*, 18:19–37, 2001.

[12] S. Felsner. Lattice structures for planar graphs. *Electron. J. Comb.*, 11(1):Research paper R15, 24p, 2004.

[13] P. Flajolet, P. Zimmermann, and B. Van Cutsem. A calculus for random generation of labelled combinatorial structures. *Theoret. Comput. Sci.*, 132(2):1–35, 1994.

[14] H. de Fraysseix, P. Ossona de Mendez, and P. Rosenstiehl. *Pigale, Automatic Graph Drawing*. Atelier de taxiplanie, CAMS, EHESS, Paris. ALCOM-IT, ESPRIT LTR Projet 20244.

[15] C. Gotsman. On the optimality of valence-based connectivity coding. *Computer Graphics Forum*, pp. 99–102, 2003.

[16] X. He, M.-Y. Kao, and H.-I. Lu. Linear-time succinct encodings of planar graphs via canonical orderings. *SIAM J. on Discrete Mathematics*, 12(3):317–325, 1999.

[17] X. He, M.-Y. Kao, and H.-I. Lu. A fast general methodology for information-theoretically optimal encodings of graphs. *SIAM J. Comput*, 30(3):838–846, 2000.

[18] G. Kant. Drawing planar graphs using the canonical ordering. *Algorithmica*, 16:4–32, 1996. (also *FOCS'92*).

[19] A. Khodakovsky, P. Alliez, M. Desbrun, and P. Schröder. Near-optimal connectivity encoding of 2-manifold polygon meshes. *Graphical Model*, 64(3–4), 2002.

[20] H.-I. Lu. Linear-time compression of bounded-genus graphs into information-theoretically optimal number of bits. In *SODA*, pp. 223–224, 2002.

[21] C. McDiarmid, A. Steger, and D. Welsh. Random planar graphs. 20pp, 2003.

[22] J. I. Munro and V. Raman. Succinct representation of balanced parentheses, static trees and planar graphs. In 38^{th} *Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pp. 118–126. IEEE Computer Society Press, Oct. 1997.

[23] A. Nijenhuis and H. S. Wilf. *Combinatorial Algorithms*. Academic Press, second edition, 1978.

[24] D. Osthus, H. J. Prömel, and A. Taraz. On random planar graphs, the number of planar graphs and their triangulations. *J. Combin. Theory Ser. B*, 88(1):119–134, 2003.

[25] D. Poulalhon and G. Schaeffer. Optimal coding and sampling of triangulations. In *ICALP*, pp. 1080–1094, 2003.

[26] J. Rossignac. Edgebreaker: Connectivity compression for triangle meshes. *IEEE Transactions on Visualization and Computer Graphics*, 5(1):47–61, 1999.

[27] G. Schaeffer. Bijective census and random generation of Eulerian planar maps with prescribed vertex degrees. *Electron. J. Combin.*, 4(1):Research Paper 20, 14 pp., 1997.

[28] G. Schaeffer. Random sampling of large planar maps and convex polyhedra. In *STOC*, pp. 760–769, 1999.

[29] W. Schnyder. Embedding planar graphs on the grid. In *SODA*, pp. 138–148, 1990.

[30] C. Touma and C. Gotsman. Triangle mesh compression. In *Graphic Interface Conference*, pp. 26–34, 1998.

[31] W. T. Tutte. A census of planar maps. *Canad. J. Math.*, 15:249–271, 1963.

[32] D. B. Wilson. An annotated bibliography of perfectly random sampling with markov chains. Maintained on `http://dimacs.rutgers.edu/~dbwilson/exact`.

[33] D. B. Wilson. Determinant algorithms for random planar structures. In *SODA*, pp. 258–267, 1997.