Random Sampling of Large Planar Maps and Convex Polyhedra

Gilles Schaeffer
Laboratoire Bordelais de Recherche en Informatique,
LaBRI, Université Bordeaux I,
351, cours de la libération, 33405 Talence, France
Email: Gilles.Schaeffer@labri.u-bordeaux.fr

Abstract

We present a versatile almost uniform random generator for planar embeddings of graphs (usually called *maps*). This simple pseudo-algorithm is linear on average and gives in a few seconds random maps with up to one million edges or vertices. The class of planar graphs that can be obtained includes graphs of convex polyhedra (or 3-connected planar graphs) and convex irreducible triangulations (or 4-connected maximal planar graphs). Our algorithm relies on a combinatorial approach. First, new simpler compact encodings are defined using canonical unlabelled covering trees. Second, a general *extraction/rejection* pseudo-algorithm is defined for composed structures. If correctly *tuned* (we provide the necessary analysis for maps), it applies efficiently to a much wider class of planar graphs than previously known methods.

1 Introduction

A *map* is an embedding of a graph in the plane, considered up to continuous deformations of the plane. Maps have been studied by mathematicians mainly because they give an angle of approach towards planar graphs. Indeed, unlike the random graph $G(n, m)$, random planar graphs are still quite mysterious objects. Basic questions, like “How many planar graphs can be constructed on m vertices ?” or “What is the average diameter of a planar graph ?” remain widely open. Random maps are somewhat easier to study and results for maps can help on planar graphs [3, 5]. For instance, using old formulae of Tutte for planar maps [23], it is possible to prove that the number c_m of planar graphs with m vertices satisfies [10]:

$$\frac{256}{27} \leq \lim_{m\to+\infty} (c_m)^{1/m} \leq 8\frac{256}{27}.$$

These deceptively poor bounds are the best ones known. Similarly, even for maps, the diameter is still intractable. The first motivation for our work is to provide, via random generation of maps, an intuition on statistics on planar graphs. As a first example we have been able to observe limit laws for the diameter of maps.

In the sequel we shall allow graphs to have multiple edges and loops, and call a graph or a map *simple* otherwise. Tutte has pointed out that *rooted maps* should be considered, *i.e.* maps with one oriented edge, called the *root*. The introduction of a root does not affect statistical properties because maps are known to have very few automorphisms [20]. On the other hand the root gives a canonical way to “catch” a map, and makes all computations much easier. *In the sequel all maps are rooted, unless explicitly stated.*

Remark that considering rooted maps introduces a bias equal to the size of the automorphism group of the map. This is precisely the notion of random maps which is considered in the physics community as a powerful tool to discretize random surfaces and study graphs or colorings on them: it sometimes allows easier exact resolutions than more rigid lattices (see e.g. [8]). In this context, a quite efficient random generator was given for cubic maps [1].

Recently there has been a renewed interest in using Markov chains to generate combinatorial objects, mainly because of new techniques to prove rapid mixing and obtain exact sampling (see [26]). This approach has not yet proved successful for planar graphs: although it is easy to define a Markov chain converging to uniformity on the set of all planar graphs with m vertices [10], its mixing time seems difficult to evaluate. Moreover one cannot hope to efficiently obtain random graphs with more than a few hundred vertices. A related approach could be based on exhaustive generators, as suggested in [16].

Reprint from the 31th Annual ACM Symposium on Theory of Computing (STOC), May 1999, Atlanta, Georgia.

main family	intermediate family	underlying trees	bit counts
general	quartic (all degree 4)	binary	$3.58n$
bipartite (or Eulerian)	bipartite cubic	binary	$3n$
m-constellations	m-Eulerian	general	$(m + \log_2 m)n$
non separable	quartic without 2-cocycle	ternary	$3n$
loopless triangulations	non separable cubic	ternary	$\frac{4}{3}n$

Table 1: Maps (with n edges, loops and multiple edges allowed) encoded or generated in linear time via conjugation.

maps	simple	smooth	no leaf	2-c	non separ.	3-c	4-c
all	ok	ok	ok	ok	ok	ok	no
bipartite	ok	no	ok	ok	ok	no	no
triangulations	ok	–	–	–	–	ok	ok

Table 2: Some extra properties reachable via extraction/rejection. (Non separable means loopless 2-connected, smooth means without vertices of degree 2.)

In this work, we take advantage of a good knowledge of enumerative properties. This approach has recently proved efficient for sampling trees [2], words of context-free grammars [13, 11], covering trees and other plane structures counted by determinants [25]. It is usually more complicated to apply than Markov chains but when it works, very efficient algorithms are obtained.

The first part of the present work, *conjugation of trees* can also be viewed as a constructive proof of some formulae of Tutte [23], that yields both simple compact encodings and optimal uniform random generators. Table 1 gives the basic families that benefit from this approach, together with bit counts for our encodings. We did not take care here to support queries, because the techniques of [18, 7] could very well apply. Our bit counts are comparable to theirs[1], but our encodings are significantly simpler. Moreover we derive from them extremely efficient random generators: we were able in less than one hour on a standard PC to sample 100,000 maps with 2000 edges, or to construct in a few seconds a random map with one million edges. The most efficient previously known algorithms [1, 9] were limited to a specific family and much slower.

Then we go beyond the families of Table 1, in particular to obtain 3-connected and maximal 4-connected planar graphs (which are graphs of convex polyhedra and irreducible convex triangulations). We achieve this main result, together with various others (see Table 2) via a new method: *extraction/rejection*. It combines the usual rejection principle: "sampling from a *not too big* superset of the target gives good chances to strike", with *composition schemes* that are common for maps (but exist also for other graphs). When properly *tuned*, this method is very effective.

Complete proofs can be found in [22]. Preliminaries and related works appear in [6, 21, 15].

[1]Typically, to our credit comes the distinction between bipartite and general maps, but we cannot achieve the $\frac{3}{2}(\log 3)n$ bits for simple 3-connected maps of [7].

2 Conjugation of trees

2.1 The cycle lemma

We shall need a folklore result on strings, called *the cycle lemma*. A *conjugate* of a string w is a string $w' = vu$ where $w = uv$ and v is not empty. We restrict our attention to the alphabet $\{x, \bar{x}\}$ where x is viewed as an opening parenthesis and $\bar{x}$ as a closing parenthesis.

Lemma 1 *(Cycle lemma) Let w be a string with n opening and $n+p$ closing brackets (n and p positive). There exists a unique factorization $w = u\bar{x}v_1\bar{x}v_2\bar{x}\dots v_{p-1}\bar{x}u'$ such that*

- *v_i is a balanced parenthesis string for all i*
- *$u'u$ is a balanced parenthesis string.*

Proof. Immediate. □

A string w has the *Lukacievicz property* if $w = w'\bar{x}$ where w' is a balanced parenthesis string.

The cycle lemma (in a more general form) has been used to enumerate generalised parenthesis strings and various kinds of elementary trees (like k-ary trees). It yields simple algorithms for linear time random generation of these objects (see [2] and references therein).

2.2 Maps, unlabelled trees and dual bsf-traversal, a first instance

We present our new encoding method. Due to space limitation, we deal here with a specific case, namely that of *bipartite cubic maps*[2]. General results are stated in the next section.

All our encoding trees share the following two properties:

[2]Bipartite cubic maps with $3n$ edges are in one-to-one correspondence with general bipartite maps with n edges (see [24]) so that all results also hold for the latter.

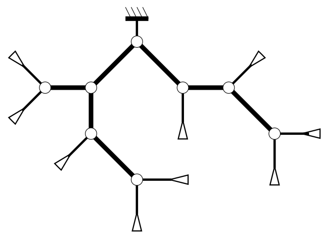


Figure 1: A planted complete binary tree.

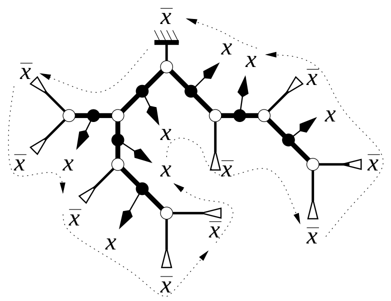
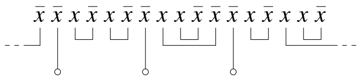


Figure 2: A blossom tree and its border.

- They are *planted plane trees*, *i.e.* the root is a vertex of degree one.
- The vertices of degree one are of two kinds, the *leaves* and the *buds*. The root is always a leaf. We refer to inner vertices as *nodes*.

It is more convenient to start with the decoding procedure. Let us consider the planted complete binary tree with $n = 8$ nodes and $n+2 = 10$ leaves of Figure 1. In the middle of each internal edge, add a black vertex with a *bud*, on one of the two sides (Figure 2). We call a tree that is obtained this way a *blossom tree*. There are $n-1$ internal edges so that the number of blossom trees with n white nodes is 2^{n-1} times the number of binary trees with n nodes, which is $\frac{1}{n+1}\binom{2n}{n}$ (as one can prove using the cycle lemma).

To a blossom tree with n nodes, we associate a string with n letters x and $n+3$ letters $\bar{x}$, called its *border*. It is obtained by writing down x for buds and $\bar{x}$ for leaves as one goes along a postfix traversal of the tree, finishing with the root leaf. For instance the border of the tree of Figure 2 is $\bar{x}\bar{x}x\bar{x}x\bar{x}\bar{x}xx\bar{x}\bar{x}\bar{x}x\bar{x}xx\bar{x}$. If we consider the border as a cyclic word, we can uniquely associate each opening bracket to a closing one by recursively matching each x with the closest available $\bar{x}$. In the latter example we obtain



By construction, three letters $\bar{x}$ remain single. Let us define accordingly the *partial closure* of a blossom tree:

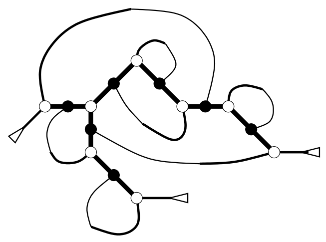


Figure 3: Partial closure of the tree of Figure 2.

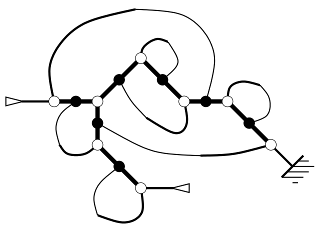
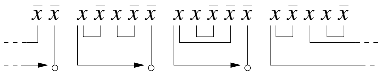


Figure 4: Partial closure of a balanced conjugate of the tree of Figure 2.

we simply match buds and leaves according to the previous pattern and form new edges (Figure 3). Each time an edge is closed, a new face is created. We call the leaves that remain free, *single leaves*. According to the cycle lemma, and as there are three more leaves than buds, the border of a tree factors, up to conjugation, into three words having the Lukacievicz property. In our example,



This leads us to the definition of *conjugation of trees*: two blossom trees are *conjugate* if one is obtained from the other by changing the root leaf. Of course one can object that conjugacy classes are nothing more than unplanted trees, yet we shall see that our definition provides a convenient analogy with conjugation on words.

Lemma 2 *The borders of two conjugated trees are conjugated. In particular the single leaves of an unplanted tree are well defined.*

Proof. Immediate. □

Carrying on with the analogy we define *balanced blossom trees*: they are blossom trees whose border is a product of three strings having the Lukacievicz property. Equivalently one can say that a blossom tree is balanced if its root remains single after the partial closure (Figure 4). Remark that the number of balanced blossom trees is simply $\frac{3}{n+2}\cdot\frac{2^{n-1}}{n+1}\binom{2n}{n}$, as 3 leaves among $n+2$ are single in any blossom tree[3]. If we restrict our

[3] One could object that there can be a symmetry of order three and hence, only one balanced tree in some conjugacy class. But then there are also only $(n+2)/3$ distinct trees in these.

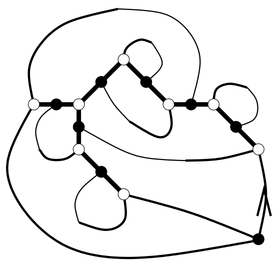


Figure 5: Complete closure.

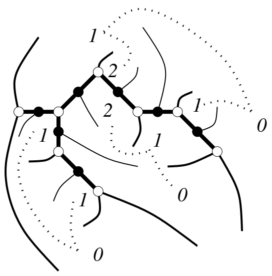


Figure 6: Counterclockwise dual bfs-traversal.

attention to these balanced trees, we can define the *complete closure* by joining the three single leaves to a new vertex and rooting the resulting map towards the root leaf (Figure 5).

Theorem 1 *The complete closure defines a one-to-one correspondence between balanced blossom trees with n nodes and bipartite cubic maps with $3n$ edges.*

Proof. Clearly, in order to reverse the complete closure, one should start with a bipartite cubic map and delete the origin of the root so as to form three single leaves. Then the difficult part is to reverse the partial closure, namely to find a covering tree of the remaining map such that,

(i) each black vertex is incident to one and only one *external edge* (*i.e.* an edge that does not belong to the covering tree),

(ii) when the map is drawn in the plane with single leaves in the infinite face, each external edge, oriented from its black end to its white one, defines with the covering tree a bounded component on its left.

The first condition ensures that *opening* external edges (*i.e.* replacing each external edge by a bud on its black end and a leaf on its white end) yields a blossom tree. The second condition ensures that the complete closure of the tree gives back the original map. Hence Theorem 1 follows if one can show that there exists a unique covering tree satisfying conditions (i) and (ii). This is done in forthcoming Proposition 1. □

Let us now describe the reverse mapping, *i.e.* the *encoding* of a bipartite cubic map by a blossom tree. Consider the bipartite cubic map of Figure 5. As exemplified by Figure 6, we shall do a specific *breadth first search* traversal of its faces, that we call *the counterclockwise dual bfs-traversal*: it is defined as follows, faces reached at step k being at *depth* k.

- at step 0, the three edges around the origin of the root edge are opened and three faces are reached.
- at step k, we reach all unvisited faces that are adjacent to a face at depth $k-1$ through an edge joining a black vertex to a white one in counterclockwise order around the remaining map. When several edges allow us to reach the same face, we open only *the first one after the root edge* in counterclockwise order.

Proposition 1 *Given a bipartite cubic map, the counterclockwise dual bfs-traversal leaves a tree that is the unique covering tree satisfying conditions (i) and (ii).*

Proof. Omitted. □

Remark that both encoding and decoding are linear in time and that the space used for the encoding is optimal up to lower order terms. Indeed a blossom tree can be encoded by a binary string of length $3n$: a parenthesis string of length $2n$ for the underlying binary tree and a 0-1 string of length $n-1$ to specify which bud is chosen on each edge. But according to Theorem 1, the binary logarithm of the number of bipartite cubic maps is $\log_2 \frac{3\cdot 2^{n-1}(2n)!}{n!(n+2)!} = 3n + o(n)$. In passing we have recovered Tutte's formula for the number of bipartite cubic maps [23].

2.3 General encoding results

The simplicity of our encoding of bipartite cubic maps reflects the elegance of Tutte's enumerative formula (although his proof was not at all based on conjugation of trees). Several other families of maps enjoy such an elegant enumerative formula (see [23]), and we give encodings accordingly:

Theorem 2 *All maps of Table 1 can be encoded and decoded in linear time by some elementary balanced blossom trees without labels.*

Proof. Omitted. □

The m-constellations that appear in Table 1 are maps that generalize bipartite maps. In [6] we use their encoding in terms of blossom trees in a different context to enumerate ramified coverings of the sphere by itself and a complete proof of the one-to-one correspondence is given there.

	i vertices/ j faces
all	$O((i+j)ij)$ + tables
bipartite	$O((i+j)ij)$ + tables
non separable	$O(i+j)$

Table 3: The cost of vertex number control.

	given	restricted to $\mathcal{D}$
Eulerian	$O(n \log n)$	$O^*(n^2\|\mathcal{D}\|^2)$+ tables
m-Eulerian	$O(n \log n)$	$O^*(n^2\|\mathcal{D}\|^2)$+ tables

Table 4: The cost of vertex degrees control.

2.4 Uniform random generation

Consider again the family of bipartite cubic maps and their balanced blossom trees.

Theorem 3 *The random generation of blossom trees with n internal nodes can be done in three independent steps:*

1. *choose uniformly at random a binary tree among all planted complete binary trees with n nodes,*
2. *toss a coin independently for each edge to place buds,*
3. *conjugate the tree in one of the three possible ways, with probability* $1/3$ *for each.*

This algorithm yields a uniformly chosen balanced blossom tree with n internal nodes in linear time. All computations are made using integers within $[0, 2n]$.

The same result is valid for all families of Table 1.

In order to generate a map with given numbers of edges and vertices, or with restricted or given degrees of faces, one must be able to keep track of these parameters on the trees. As indicated by Tables 3 and 4, this can often be done, but except for non separable maps, at the expand of a greater complexity (see Appendix).

3 The core of a map

3.1 Composition schemes

Let $\mathcal{M}$ and $\mathcal{C}$ be two families of maps, with $\mathcal{C}_n \subset \mathcal{M}_n$ where $\mathcal{M}_n$ and $\mathcal{C}_n$ denote the elements of $\mathcal{M}$ and $\mathcal{C}$ with n edges. The $\mathcal{C}$-core C of a map M of $\mathcal{M}_n$ is the largest map C of $\mathcal{C}$ *contained* in M, and whose root edge is the root of M. Typical examples include: the 3-connected core of a non separable map, where *contained* means being a 3-connected component; the non separable core of a map, where *contained* means being a submap. Let us consider more precisely the latter example: $\mathcal{M}$ is the family of all maps, and $\mathcal{C}$ the family of non separable maps. Then the following decomposition holds (see Figure 7):

$$\mathcal{M} \simeq \mathcal{C} \circ (\mathcal{X}\mathcal{M}^2).$$

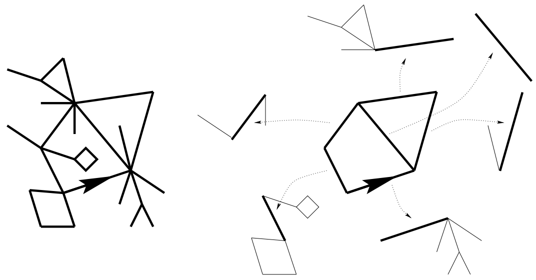


Figure 7: The non separable core of a map.

In this composition relation, $\mathcal{X}\mathcal{M}^2$ denote the family of maps whose root edge ($\mathcal{X}$) join two maps of $\mathcal{M}$. It means that a map M of $\mathcal{M}_n$ can be decomposed into a tuple $(C; H_1, H_2, \ldots, H_k)$ where C belongs to $\mathcal{C}_k$ and each H_i to the family $\mathcal{X}\mathcal{M}^2$: Given M, the core C is the largest non separable submap of M containing its root. Each H_i is composed of the ith edge e_i of C (in any given canonical order on edges) and of the pieces of $M \setminus C$ following e_i counterclockwise on both ends of e_i. Conversely, given $(C; H_1, \ldots, H_k)$ we reconstruct a map M by substituting H_i for the ith edge of C.

Such *composition schemes* happen to be quite common between maps. Table 5 gives various interesting examples and several others could easily be added to the list (see [22]). The first one and the three last were used by Tutte for enumerative purposes [23].

3.2 Distribution of the size of cores

Extraction/rejection algorithms of Section 4.1 are based on repeatedly taking cores of random maps until a large enough core is obtained. Their efficiency thus relies on the behaviour of this parameter. The following result enables us to control the number of edges:

Theorem 4 *For any scheme of Table 5, define the probability* $\Pr(\nu(C_n) = k)$ *that a random map* C_n *of size* n *has a core of size* $\nu(C_n) = k$. *Let* $\lambda(n) = o(n)$ *be any function going to infinity and denote by* $I_n(\alpha)$ *the interval* $\alpha n \pm n^{2/3}$. *Then there exist positive constants* α_0, p_f *and* p_ν *such that*

$$\lim_{n\to+\infty} \Pr(\nu(C_n) < \lambda(n)) = p_f,$$
$$\lim_{n\to+\infty} \Pr(\nu(C_n) \in I_n(\alpha_0)) = p_\nu = 1 - p_f.$$

We call α_0 *the* tuning factor *and* p_ν *the* hitting probability *of the composition scheme.*

Proof. These results are excerpts of a much more accurate description of the asymptotic distribution of the size of the core of a random map. The proof is based on generating functions and saddle point analysis. A first instance of this problem was studied in [4]. In [22]

maps ($\mathcal{F}$)	cores ($\mathcal{G}$)	scheme	tuning factor
general	non separable	$\mathcal{F} \simeq \mathcal{G} \circ (\mathcal{X}\mathcal{F}^2)$	1/3
general	loopless	$\mathcal{F} \simeq \mathcal{L} + \mathcal{G} \circ (\mathcal{X}((\mathcal{X}\mathcal{F})^*)^2)$	2/3
loopless	simple	$\mathcal{F} \simeq \mathcal{G} \circ (\mathcal{X}\mathcal{F})$	2/3
bipartite	bip. simples	$\mathcal{F} \simeq \mathcal{G} \circ (\mathcal{X}\mathcal{F})$	5/9
non separable	3-connected	$\mathcal{F} \simeq \mathcal{D} + \mathcal{G} \circ \mathcal{F}$	1/3
cubic n. s.	cubic 3-c	$\mathcal{F} \simeq \mathcal{G} \circ (\mathcal{X}(1+\mathcal{F})^3)$	1/2
cubic 3-c.	cubic 4-c	$\mathcal{F} \simeq \mathcal{F} \cdot \mathcal{G} \circ (\mathcal{X}\mathcal{F}^2)$	1/2

Table 5: A selection of composition schemes. ($\mathcal{X}$ is an edge, $\mathcal{L}$ and $\mathcal{D}$ auxiliary families.)

this bimodal distribution is shown to hold for various *critical* composition schemes like the ones of Table 5. □

Theorem 4 means that if we wish to obtain a map with a core having about n edges, we can maximize our chances by selecting a map with n/α_0 edges. However, in order to generate maps with given numbers of edges and vertices, we need more refined statistics.

3.3 Edges and vertices control

Suppose for instance that we seek a random 3-connected map with k edges and ℓ vertices. According to [5], random 3-connected maps with k edges are very likely to have about $k/2$ vertices; more precisely, for a map with k edges and ℓ vertices, let the *imbalance* x be defined by $\ell + 1 = k(1/2 + x)$. Then the number of faces is $k(1/2 - x) + 1$ and the probability that a random map with k edges has imbalance x is $O(k^{-3/2}\exp(-ckx^2))$ when k goes to infinity, uniformly for $|x| < \frac{1}{2} - \varepsilon$ (where $c > 0$ and $\varepsilon > 0$ are constant). This statement implies that the probability of getting a core with a significant imbalance from a random map with n edges is very small. (Recall that all cores with k edges are equally likely to appear.)

A natural idea to increase the probability that the core of a map be unbalanced is to pick a map that is itself unbalanced. The following theorem explains how the imbalance of a map is related to that of its core:

Theorem 5 *Given a non separable map M we denote x its imbalance, n its number of edges, y the imbalance of its core and k the number of edges of its core, and we let $\alpha = k/n$. Fix $-1/2 < x < 1/2$ and $0 < \varepsilon < 1/6$. Then if the map M is chosen uniformly among all non separable maps of size n and imbalance x we have,*

$$\lim_{n\to+\infty} \Pr(|y - x/3| < \varepsilon) = 1$$

$$\lim_{n\to+\infty} \Pr\left(|\alpha - \alpha_0(x)| < \frac{\lambda(n)}{n^{1/3}}\right) = \frac{16\alpha_0(x)^2}{9},$$

$$\lim_{n\to+\infty} \Pr(\alpha < \lambda(n)) = 1 - \frac{16\alpha_0(x)^2}{9},$$

where

$$\alpha_0(x) = \frac{(1-2x)(1+2x)}{3(1-2x/3)(1+2x/3)}.$$

and $\lambda(n) = o(n)$ denote any fixed function going to infinity with n.

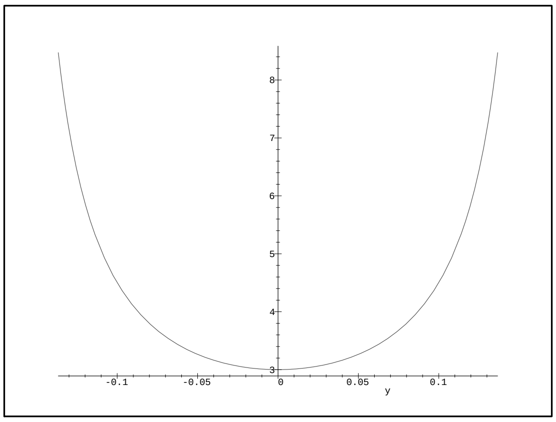


Figure 8: Plot of $1/\alpha_0(3y)$: optimal choice of n/k to get a core with k edges and imbalance y.

Proof. Omitted. Based on a quite technical bivariate saddle point analysis. □

This theorem means that to obtain a 3-connected core with k edges and imbalance y, we should select a random non separable map with $k/\alpha_0(3y)$ edges and imbalance $3y$. As clearly shown by Figures 8 and 9, when the imbalance gets close to 1/2, the tuning ratio becomes very low and the hitting probability decreases. In particular we see that this is no efficient way to obtain maximal 3-connected maps (hopefully these dualy cubic maps are given directly by Theorem 4).

4 More random maps and planar graphs

4.1 Extraction/Rejection algorithms

Let $\mathcal{M}$ be a family of objects for which we have a uniform random generator, and $\mathcal{C}$ a family related to $\mathcal{M}$ by a composition scheme ruled like in Theorem 4 by a tuning factor α_0 and a hitting probability p_ν. Then, given an expected size n and a tolerance Δ, we consider the following pseudo-algorithm whose output is an element of $\mathcal{C}$ with size in $[n \pm \Delta] = [n - \Delta, n + \Delta]$.

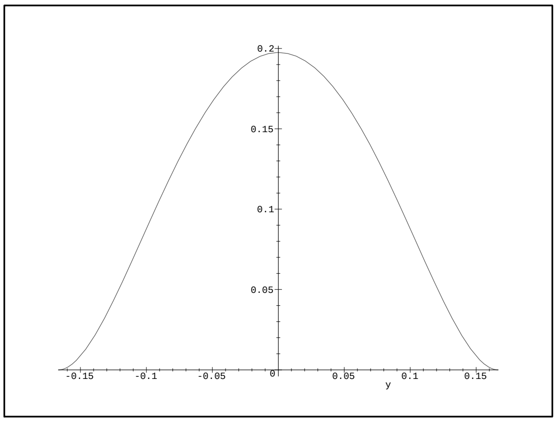


Figure 9: Plot of $\frac{16}{9}\alpha_0(3y)^2$: hitting probability versus wished imbalance of the core.

1. Select a random element of $\mathcal{M}$ of size n/α_0.
2. Extract its core C. If $|C| \notin [n \pm \Delta]$ then start again in 1, otherwise C is the output.

Theorem 6 *The output C of the extraction/rejection algorithm is chosen uniformly among all elements of $\mathcal{C}$ having the same size.*

The complexity of one iteration is the complexity of the random generation of an element of $\mathcal{M}$, plus that of the extraction, which is linear for all our schemes.

The average number of iterations is asymptotically for large n,

$$O\left(\frac{n^{2/3}}{\Delta} + \frac{1}{p_\nu}\right).$$

In particular, when the generation in $\mathcal{M}$ has linear complexity, the average complexity of the generation in $\mathcal{C}$ is

- *linear for $\Delta = \Omega(n^{2/3})$,*
- *$O(n^{5/3})$ for exact size random generation* i.e. *for $\Delta = O(1)$.*

Proof. Uniformity is a direct consequence of the definition of composition scheme: the possible choices of subcomponents depend only on the size of the core, so that all cores with the same size have the same probability to appear in a random element of $\mathcal{M}$.

The complexity of extracting cores is linear: in all our composition schemes, we only need to detect and suppress some short cycles. The most difficult extraction is that of the 3-connected core of a non separable map: we can use Hopcroft and Tarjan's algorithm for 3-connected components of graphs [14], or take advantage of the planarity to design a simpler algorithm, based on Mullin and Schellenberg's old remark that detecting 3-connected components in a map amounts to detecting 4-cycles in its *quadrangulation* [17].

The asymptotic evaluation of the average number of iterations is a direct corollary of Theorem 4. Remark that if the algorithm is not properly *tuned* using α_0, the number of iterations increases dramatically. □

Using conjugation of trees we have obtained all families of Table 1. Applying the extraction/rejection algorithm to the schemes that involve these maps we obtain several entries of Table 2. The remaining ones, like 4-connected triangulations or simple maps are obtained via a second application. When we compose two compatible schemes, there is no need for exact random generation on the first step, so that the complexity is only increased by the multiplication of tuning factors.

Using Theorem 5, a similar result holds for imbalanced cores. This is especially interesting for 3-connected maps as they are extracted from non separable maps whose generation is linear even for given numbers of vertices and edges.

4.2 Unrooted maps and convex polyhedra

One could object that our maps are *rooted* so that the distribution is not exactly uniform on unlabelled maps. However maps are known to have very few symmetries [20, 3] and this allows us to claim that our algorithms are *almost uniform* random generators for unrooted maps. We say that a generator is almost uniform on a family $\mathcal{M}_n$ if there exists an exponentially small subset $\mathcal{S}_n$ of $\mathcal{M}_n$ (*i.e.* there exist $\alpha > 0$ and $\lambda > 0$ such that $|\mathcal{S}_n| < |\mathcal{M}_n| e^{-\lambda n^\alpha}$) such that the generator is uniform on $\mathcal{M}_n \setminus \mathcal{S}_n$ but may neglect elements of $\mathcal{S}_n$ (*i.e.* they are selected with a probability smaller or equal to $1/|\mathcal{M}_n|$).

Moreover, thanks to Whitney's celebrated theorem on the uniqueness of embeddings of 3-connected planar graphs, we can claim that we have an almost uniform random generator for 3-connected planar graphs, and 4-connected maximal planar graphs.

Finally, using Tutte's theorem on graphs of convex polyhedra, we can claim that these algorithms generate almost uniformly graphs of convex polyhedra and of irreducible convex triangulations.

5 Conclusion

We have presented a new approach for encoding embedded graphs by trees. Using this approach together with a new extraction/rejection method, we have proposed extremely efficient random generators for various classes of maps.

Our encodings have nice statistical properties: For instance, the fact (see [12]) that binary trees have average height $O(\sqrt{n})$, implies directly that the average eccentricity (and thus the average diameter) of bipartite maps is bounded above by $O(\sqrt{n})$. Indeed each binary tree has the same probability to appear as covering tree in our encoding. This bound (valid for all maps in Table 1) may not seem very surprising, but it is, to our knowledge, the only bound available.

Moreover, preliminary experiments on the diameter using our random generator strongly suggests the existence of a limit law.

A complete version of the generator will be made available as an independent part of [19], designed with the specific purpose of practical testing of planarity and coloring algorithms on large random inputs.

References

[1] M. E. Agishtein and A. A. Migdal. Recursive sampling of planar graphs and fractal properties of a two-dimensional quantum gravity. *Internat. J. Modern Phys. C*, 1(1):165–179, 1990.

[2] L. Alonso, J. L. Rémy, and R. Schott. A linear-time algorithm for the generation of trees. *Algorithmica*, 17(2):162–182, 1997.

[3] E. A. Bender, E. R. Canfield, Z. Gao, and L. B. Richmond. Submap density and asymmetry results for two parameter map families. *Combin. Probab. Comput.*, 6(1):17–25, 1997.

[4] E. A. Bender, L. B. Richmond, and N. C. Wormald. Largest 4-connected components of 3-connected planar triangulations. *Random Structures Algorithms*, 7(4):273–285, 1995.

[5] E. A. Bender and N. C. Wormald. The number of rooted convex polyhedra. *Canad. Math. Bull.*, 31(1):99–102, 1988.

[6] M. Bousquet-Mélou and G. Schaeffer. Enumeration of planar constellations. Research Report 1209-98, LaBRI, Université Bordeaux I, July 1998. 21 pages. To appear in *Adv. in Applied Math.*

[7] R.C.-N. Chuang, A. Garg, X. He, M.-Y. Kao, and H.-I. Lu. Compact encodings of planar graphs via canonical orderings and multiple parentheses. In *25th ICALP*, number 1443 in LNCS, pages 118–129, New-York, 1998. Springer-Verlag.

[8] G. M. Cicuta, L. Molinari, E. Montaldi, and S. Stramaglia. A matrix model for random surfaces with dynamical holes. *J. Phys. A*, 29(14):3769–3785, 1996.

[9] A. Denise. Generation of random planar maps. In *Graph Drawing'93*, Paris, 1993.

[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] A. Denise and P. Zimmermann. Uniform random generation of decomposable structures using floating-point arithmetic. Technical Report 3242, INRIA, 1997. To appear in *Theoretical Computer Science.*

[12] P. Flajolet, Z. Gao, A. Odlyzko, and B. Richmond. The distribution of heights of binary trees and other simple trees. *Combin. Probab. Comput.*, 2(2):145–156, 1993.

[13] 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. Fundamental Study.

[14] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2:135–158, 1973.

[15] B. Jacquard and G. Schaeffer. A bijective census of nonseparable planar maps. *J. Combin. Theory Ser. A*, 83(1):1–20, 1998.

[16] B. D. McKay. Isomorph-free exhaustive generation. *J. Algorithms*, 26(2):306–324, 1998.

[17] R. C. Mullin and P. J. Schellenberg. The enumeration of c-nets via quadrangulations. *J. Combinatorial Theory*, 4:259–276, 1968.

[18] J.I. Munro and V. Raman. Succinct representation of balanced parentheses, static trees and planar graphs. In *38th IEEE FOCS*, pages 118–126, Miami Beach, Florida, Oct. 1997.

[19] *Pigale, An Automatic Graph Drawing Project.* Atelier de taxiplanie, CAMS, École des Hautes Études en Sciences Sociales, Paris. Part of ALCOM-IT, ESPRIT LTR Projet 20244.

[20] L. B. Richmond and N. C. Wormald. Almost all maps are asymmetric. *J. Combin. Theory Ser. B*, 63(1):1–7, 1995.

[21] 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. (electronic), 1997.

[22] G. Schaeffer. *Conjugaison d'arbres et cartes combinatoires aléatoires.* PhD thesis, Université Bordeaux I, 1998.

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

[24] W. T. Tutte. Duality and trinity. In *Infinite and finite sets, Vol. III*, pages 1459–1472. Colloq. Math. Soc. János Bolyai, Vol. 10, Amsterdam, 1975. North-Holland.

[25] D. B. Wilson. Determinant algorithms for random planar structures. In *8th Annual ACM-SIAM SODA*, pages 258–267, New Orleans, 1997.

[26] D. B. Wilson. Annotated bibliography of perfectly random sampling with Markov chains. In *Microsurveys in Discrete Probability*, volume 41 of *DIMACS*, pages 209–220, 1998. Updated at `http://dimacs.rutgers.edu/~dbwilson/exact`.

A The cost of controlling numbers of vertices and faces

In order to support the complexities indicated in Tables 3 and 4, let us first elaborate on the example of Section 2.2. There is a unique way to colour faces of a bipartite cubic maps with three colors in such a way that (i) they appear in counterclockwise order around white vertices and (ii) the root edge is bordered by faces of color 2 and 3 on its left and right sides. For instance, Figure 10 shows the unique 3-coloring of the map of Figure 5. As illustrated by Figures 10 and 11, this 3-coloring can be used to define a one-to-one correspondance between bipartite cubic maps with $3n$ edges and bipartite maps with n edges (we use color 1 and 2 as vertices). The bipartite map of Figure 11 has 6 vertices, 4 faces and 8 edges. Equivalently, we could say that the bipartite cubic map has 4 faces with color 3 and 24 edges. But recall that during the partial closure operation, each pairing of a bud and a leaf creates a new face on its right. We can thus label matched leaves of the blossom tree by colors of the corresponding faces. Finally, we also give to the three single leaves the color of the face on their right as illustrated by Figure 12. In fact, the cyclic order of colors around white and black vertices implies that the labelling of leaves is uniquely defined and can be computed directly from the blossom tree (starting from the root which is supposed to have label 2).

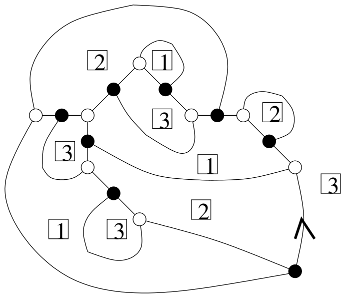


Figure 10: 3-coloring of the map of Figure 5.

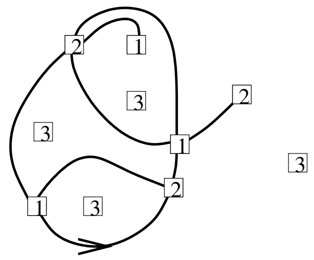


Figure 11: The bipartite map associated to the cubic map of Figure 10.

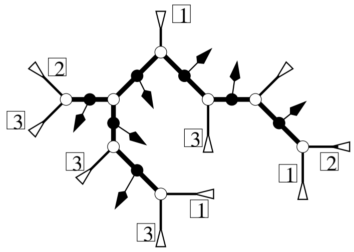


Figure 12: Labels of leaves.

According to this discussion, the generation of a random bipartite map with n edges and i faces is reduced to that of a random balanced blossom tree with n nodes and i labels 3. As a blossom tree with n nodes and i labels 3 has exactly one single leaf with label 2, we can as well generate a random blossom tree and then conjugate it into its only balanced conjugate with root labels 2.

Observe now that the method of Theorem 3 does not directly apply: the parameter i depends on positions of buds, and the number of ways to display buds on a binary tree with n nodes to obtain i labels 2 is not independant of the shape of the binary tree. Thus Step 1 and 2 of Theorem 3 cannot be done separately. One way to circumvent this difficulty is to observe that blossom trees can be encoded by words of a context-free grammar: as illustrated by Figure 13, the standard decomposition of rooted plane trees can be adapted to

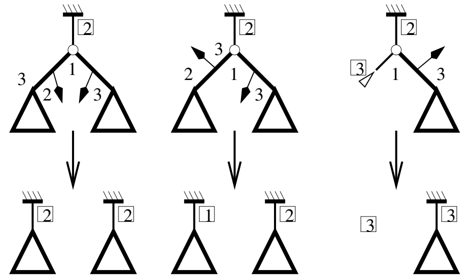


Figure 13: Three cases among the nine decompositions of a tree with root label 2 (both sons can be empty or have the bud on the left or on the right).

take labels into account. Via this decomposition, the blossom trees that we are looking for are in one-to-one correspondance with words with n letters a and i letters $\boxed{3}$ obtained from T_2 by the following derivation rules,

$$\begin{aligned} T_1 &\rightarrow \left(\boxed{2} + aT_3 + aT_1\right)\left(\boxed{3} + aT_1 + aT_2\right), \\ T_2 &\rightarrow \left(\boxed{3} + aT_1 + aT_2\right)\left(\boxed{1} + aT_2 + aT_3\right), \\ T_3 &\rightarrow \left(\boxed{1} + aT_2 + aT_3\right)\left(\boxed{2} + aT_3 + aT_1\right). \end{aligned}$$

The random generation of words given by context-free grammars is a well known problem. The methods of [13] can be extended to take into account the two parameters i and n. The $O^*(ij(i+j))$ then arises naturally. The star $*$ stresses the fact that the cost of arithmetic operations on large integers in not taken into account in [13], thus hidding higher bit complexities. Moreover, precomputed tables are needed.

Similar discussions hold for other $O^*(\ldots)$ entries of Tables 3 and 4; to conclude we explain the two entries $O(n \log n)$. In some of the generalised blossom trees that we use for other families of maps, degrees of nodes correspond to degrees of vertices. Then the random generation of blossom trees with prescribed vertex degrees is achieved using a generalised version of the cycle lemma. The problem is reduced to the generation of a random word on a given arbitrary multiset of letters, whose cost is $O(n \log n)$ (see [21] for a complete proof in the special case of Eulerian maps).

B Overview of the proof of Theorem 4 and 5

Theorem 4 and 5 are obtained together with a more precise description of the distribution of sizes of cores. A nice feature of our approach is that we can compare theoretical results given by saddle point analysis with experimental results obtained by random generation. As shown by Figures 14, and 15, results are in accordance. We sketch the successive steps of the proof of Theorem 4. The proof of Theorem 5 is similar, but more

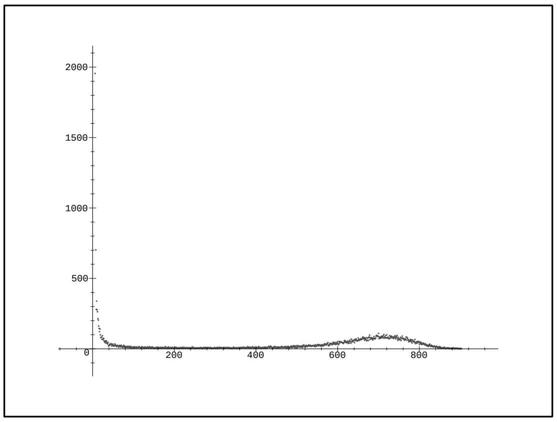


Figure 14: Numbers of 3-connected cores with k edges in 100,000 random non separable maps with 2000 edges.

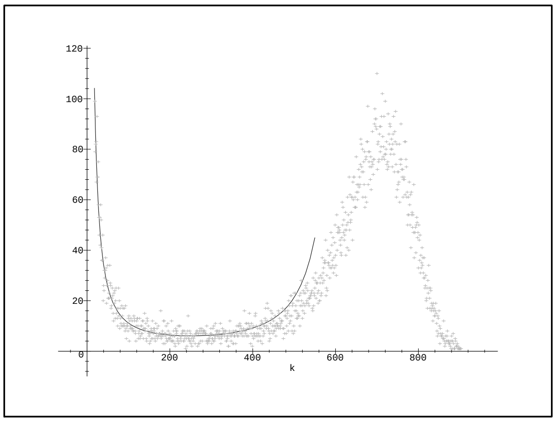


Figure 15: Comparison between experimental results of Figure 14 (for $k > 15$) and theorical values (given in the polynomially decreasing part between 0 and α_0). From left to right, observe the "discrete part", the polynomially decreasing one, the $n^{2/3}$ interval around α_0 and the exponential tail.

technical, mainly because there are four parameters instead of the two n and k of Theorem 4.

Consider the general composition scheme

$$\mathcal{M} \simeq \mathcal{C} \circ \mathcal{H},$$

where the three families contain no empty maps, and define the generating function of the family $\mathcal{M}$ by

$$M(x) = \sum_{n \geq 1} |\mathcal{M}_n| x^n$$

and similarly for $\mathcal{C}$ and $\mathcal{H}$. Then the proof goes along the following steps:

1. The composition scheme translates into a functional equation:

$$M(x) = C(H(x)).$$

In order to *mark* the size of the core, we introduce a bivaried generating function:

$$M(x, u) = C(uH(x)).$$

Observe that if we expand $M(x, u)$ (as a formal power serie)

$$M(x, u) = \sum_{n,k} f_{n,k} x^n u^k,$$

then $f_{n,k}$ is the number of maps from $\mathcal{M}$ having n edges and a core of with k edges. Therefore the probability $p_{n,k}$ that a map from $\mathcal{M}$ with n edges has a core with k edges is

$$\frac{f_{n,k}}{|M_n|} = \frac{[x^n u^k] M(x, u)}{|M_n|} = \frac{|C_k| \cdot [x^n] H^k(x)}{|M_n|}.$$

2. The generating functions of all the families of maps that we consider satisfy functional equations that allow to obtain asymptotic evaluations of $|M_n|$ and $|C_k|$. Moreover, there exists rational functions $\zeta(x)$, $\phi(x)$ and $\psi(x)$ such that

$$[x^n] H^k(x) = [x^n] \zeta(x) \phi(x)^k \phi(x)^n.$$

Using Cauchy formula this is easily expressed as

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \zeta(re^{i\vartheta}) \psi(re^{i\vartheta})^k \phi(re^{i\vartheta})^n r^{-n} e^{-i\vartheta n} d\vartheta$$

for proper choices of $r > 0$.

3. There exists a critical value α_0 such that the saddle point analysis applies when n tends to infinity with $k/n = x$ fixed in $]0, \alpha_0[$ or greater than α_0 (roughly speaking). In these invervals the probability happens to decrease toward zero, either polynomially or exponentially fast.

4. For fixed k, the limit probability $\lim_{n\infty} p_{n,k}$ can be computed and is positive for each k but the sum of these limit probabilities for all k is smaller than 1 (so that there is no discrete limit law).

5. By complementation, the probability of being in neither of the three previous domains, that is the hitting probability p_ν, is obtained.