2005 International Conference on Analysis of Algorithms DMTCS proc. **AD**, 2005, 125–138

Quadratic exact-size and linear approximate-size random generation of planar graphs

Éric Fusy[1]

[1]*Algorithms project, INRIA Rocquencourt, France*

This extended abstract introduces a new algorithm for the random generation of labelled planar graphs. Its principles rely on Boltzmann samplers as recently developed by Duchon, Flajolet, Louchard, and Schaeffer. It combines the Boltzmann framework, a judicious use of rejection, a new combinatorial bijection found by Fusy, Poulalhon and Schaeffer, as well as a precise analytic description of the generating functions counting planar graphs, which was recently obtained by Giménez and Noy. This gives rise to an extremely efficient algorithm for the random generation of planar graphs. There is a preprocessing step of some fixed small cost. Then, for each generation, the time complexity is quadratic for exact-size uniform sampling and linear for approximate-size sampling. This greatly improves on the best previously known time complexity for exact-size uniform sampling of planar graphs with n vertices, which was a little over $\mathcal{O}(n^7)$.

Keywords: planar graphs, Boltzmann samplers, rejection sampling

1 Introduction

A graph is said to be planar if it can be embedded in the plane so that no two edges cross each other. In this article, we will consider labelled planar graphs, where vertices receive distinct labels. Statistic properties of planar graphs have been intensively studied [2, 7, 8]. Very recently, O. Giménez and M. Noy [8] solved *exactly* the difficult problem of asymptotical enumeration of planar graphs. They also provide exact analytic expressions for the asymptotic probability distribution of many parameters such as for example the number of edges and the number of connected components. Since many other statistics on random planar graphs remain analytically and combinatorially untractable, it is an important issue to find an efficient procedure to generate planar graphs at random. In addition, it makes it possible to validate algorithms and programs on planar graphs, for example planarity testing, embedding algorithms, efficient procedures for finding geometric cuts, etc...

A first algorithm for the random generation of planar graphs was proposed by Denise, Vasconcellos, and Welsh [3], where a Markov chain on the set $\mathcal{G}_n$ of planar graphs with n vertices is defined. By symmetry of the transition matrix of the Markov chain, the probability distribution converges to the uniform distribution on $\mathcal{G}_n$. This algorithm is very simple and seems to work well in practice. However, it only *converges* to the uniform distribution, so that any execution of the algorithm is bound to provide non-uniform results. This is aggravated by the fact that the rate of convergence is unknown.

A second approach for uniform random generation on $\mathcal{G}_n$ was developed by Bodirsky, Gröpl and Kang [1]. It relies on the *recursive method* introduced by Nijenhuis and Wilf [10] and formalized by Flajolet, Van Cutsem and Zimmermann [5]. The recursive method is a general framework that can be implemented for any class of objects admitting a recursive decomposition. Thus, producing an object of the class uniformly at random boils down to producing the *decomposition tree* corresponding to its recursive decomposition. Then, the branching probabilities that produce the decomposition tree with suitable probability are computed using the *coefficients* counting the objects involved in the decomposition. As a consequence, this method entails a preprocessing step where large tables of large coefficients are calculated using the recursive relations that they satisfy.

Bodirsky *et al* apply the recursive method for planar graphs, which admit a well known combinatorial decomposition according to successive levels of connectivity. The coefficients enumerating planar graphs do not seem to satisfy nice recursive relations, so that the time requirement of the preprocessing step is

	Auxiliary memory	Preprocessing time	Time per generation	
Markov chains	$\mathcal{O}(\log n)$	$\mathcal{O}(1)$	*unknown*	{exact size}
Recursive method	$\mathcal{O}(n^5 \log n)$	$\mathcal{O}\left(n^7 (\log n)^2 (\log\log n)\right)$	$\mathcal{O}(n^3)$	{exact size}
Boltzmann sampler	$\mathcal{O}((\log n)^k)$	$\mathcal{O}((\log n)^k)$	$\mathcal{O}(n^2)$	{exact size}
			$\mathcal{O}(n)$	{approx. size}

Tab. 1: Comparison between the complexities of the algorithms of random generation of planar graphs.

large. More precisely, for the random generation of planar graphs with n vertices (and possibly also a fixed number m of edges), the time and memory requirements of the preprocessing step are respectively $\mathcal{O}\left(n^7 (\log n)^2 (\log\log n)\right)$ and $\mathcal{O}(n^5 \log n)$. Once the tables are computed, the time requirement of each generation is $\mathcal{O}(n^3)$

In this article, we introduce a new algorithm for the random generation of planar graphs that combines the efficiency of Markov chains [3] and the uniformity property and precise complexity analysis of the recursive method [1]. It can be implemented to produce planar graphs with a fixed size uniformly at random. Furthermore it has an approximate-size version where a small relative range, say a few percents, is allowed for the size of the output. For practical purpose, approximate-size random sampling often suffices. The approximate-size algorithm we propose is very efficient as it has linear time complexity (see Theorem 1). With this algorithm, we estimate that a careful implementation should allow the random generation of planar graphs with several tens of thousands of vertices, whereas the recursive method of Bodirsky *et al* seems to be limited to sizes of about 100.

Our algorithm is based on the principle of Boltzmann samplers, a very powerful framework for random generation of combinatorial structures recently developed by Duchon, Flajolet, Louchard, and Schaeffer in [4]. The idea of Boltzmann samplers is to relax the constraint of exact size sampling. More precisely, given a combinatorial class, a Boltzmann sampler draws an object of size n with probability proportional to x^n (or proportional to $\frac{x^n}{n!}$ for labelled objects), where x is a certain *real* parameter that can be appropriately tuned. As a consequence, the probability distribution is spread over all objects of the class, but objects with the same size receive the same probability. In particular, the probability distribution is uniform when restricted to a fixed size. Like the recursive method, Boltzmann samplers can be found for any combinatorial class admitting a recursive decomposition. This time, the branching probabilities used to produce the decomposition tree of a random object are not based on the *coefficients* (recursive method) but on the *values* at x of the generating functions of the classes involved in the decomposition.

Theorem 1 *Let $n \in \mathbf{N}$ be a target size. There exists an* exact-size *algorithm A_n producing planar graphs of size n uniformly at random. Let $\epsilon > 0$ be a fixed size-tolerance. There also exists an* approximate-size *algorithm $A_{n,\epsilon}$ producing random planar graphs with size in $[n(1-\epsilon), n(1+\epsilon)]$, and such that the distribution of planar graphs is uniform on each size $k \in [n(1-\epsilon), n(1+\epsilon)]$.*

Algorithm A_n is of quadratic time-complexity*: the expected running time of each generation is asymptotically bounded by $C \cdot n^2$, for some constant C. Algorithm $A_{n,\epsilon}$ is of* linear time-complexity*: the expected running time for each generation is asymptotically bounded by $C_\epsilon \cdot n$, where the constant C_ϵ depends on ϵ as follows: $C_\epsilon \sim_{\epsilon\to 0} \frac{C}{\epsilon}$.*

In addition, the auxiliary memory and preprocessing time required by $A_{n,\epsilon}$ and A_n are small, being of order $\mathcal{O}(\log n)^k$.

Let us also comment on the preprocessing complexity. The implementation of $A_{n,\epsilon}$ and A_n requires the storage of a fixed number of real constants, which are special values of generating functions. Using adaptative methods discussed in [4], we in fact only need to store $\mathcal{O}\left(\log n\right)^k$ bits of these special values, where k is a fixed integer. The generating functions we need to evaluate are those of different families of planar graphs (connected, 2-connected, 3-connected). A very crucial result, recently established by O. Giménez and M. Noy [8], is that there exist exact analytic expressions for these generating functions. Hence their evaluation can be done efficiently, with a linear time complexity in the number of bits we need to compute.

The complexity model used for the analysis of the algorithm is that of the number of arithmetic operations over real numbers assumed to be known exactly. Fixed-size truncation of real numbers leads to algorithms with a probablity of failure (caused by the lack of precision) that can be made arbitrarily close to 0. No failure will arise with a precision of 20 digits in practice. To achieve a complete correctness, adaptative precision routines can be called in case of failure.

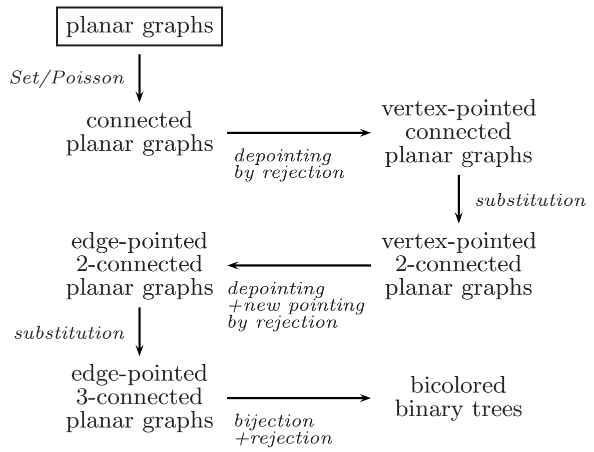

Fig. 1: The chain of reductions from planar graphs to binary trees.

The performances of the two previously existing algorithms for the random generation of planar graphs are compared with the performances of our algorithm in Table 1.

2 Overview

The algorithm we propose relies on several ideas. First we extend the classical construction rules for Boltzmann samplers, as detailed in [4], and develop the more complicated case of substitution constructions, see Section 3.2. We exploit in Section 4 the recursive decomposition of planar graphs according to successive levels of connectivity (already used in [1]) and adapt it to the Boltzmann framework. This decomposition reduces the realization of a Boltzmann sampler for planar graphs to the realization of a Boltzmann sampler for so-called 3-connected planar graphs (more precisely for edge-pointed ones). Contrary to classical recursive decompositions (e.g. binary trees) studied in [4], the transposition of the decomposition into Boltzmann samplers is not straightforward. It is also crucial to introduce new rejection techniques into the Boltzmann framework.

Then the second step, developed in Section 5, is to realize a complete Boltzmann sampler for edge-pointed 3-connected planar graphs. To do this, we use a very recent result of bijective combinatorics found by the author, D. Poulalhon and G. Schaeffer [6]: there exists a surprisingly simple correspondence (not detailed in this article) between binary trees and edge-pointed 3-connected planar graphs. The realization of a Boltzmann sampler for binary trees is straightforward and is transported by the correspondence of [6], combined with a careful rejection procedure (see Lemma 8 and Lemma 6), into a Boltzmann sampler for edge-pointed 3-connected planar graphs. The chain of reductions from planar graphs to binary trees and the techniques we will use to perform the reductions are illustrated on Figure 1.

However the size distribution of the Boltzmann sampler for planar graphs, obtained from Section 4 and Section 5, is too concentrated on objects of small size. To improve the size distribution, we *point* the objects, in a way inspired by [4]. The precise singularity analysis of the generating functions of planar graphs, recently done in [8], indicates to us that we have to point planar graphs three times to get a satisfying size distribution. In Section 6 we explain how to take the pointing operation into account in the decomposition of planar graphs. We obtain a Boltzmann sampler $\Gamma G^{\bullet\bullet\bullet}(x)$ for "triply pointed" planar graphs. The complexity of this Boltzmann sampler, for a well tuned value $x = x_n$, is analyzed in Section 7. This yields the complexity results stated in Theorem 1.

3 Boltzmann samplers

3.1 Definition

Boltzmann samplers, introduced and detailed by Duchon *et al* in [4], are a very general and powerful framework to perform random generation of objects of a combinatorial class $\mathcal{C}$. Instead of fixing a particular size for the random generation, objects are drawn under a probability distribution spread over the whole class. This distribution gives to each object of a combinatorial class $\mathcal{C}$ a weight essentially proportional to the exponential of its size. More precisely, if $\mathcal{C}$ is an unlabelled class, we consider the generating function $C(y) := \sum_{\gamma \in \mathcal{C}} y^{|\gamma|}$, where $|\gamma|$ stands for the size (e.g. the number of nodes in a tree) of γ. It is well known that the sum defining $C(y)$ converges if y is smaller than the radius of convergence ρ_C of $C(.)$. If it is the case, y is said *coherent*. Then, the probability distribution assigining to each object γ of $\mathcal{C}$ a weight,

$$\mathbf{P}_y(\gamma) = y^{|\gamma|}/C(y)$$

, is a well defined distribution, called Boltzmann distribution of parameter y. A Boltzmann sampler $\Gamma C(y)$ is simply a procedure that draws each object of $\mathcal{C}$ with probability $\frac{y^{|\gamma|}}{C(y)}$, i.e. the objects of $\mathcal{C}$ are drawn under a Boltzmann distribution. The authors of [4] give automatic recursive constructions of Boltzmann samplers for combinatorial classes that are assembled recursively using basic combinatorial constructions (union, product,...).

Boltzmann samplers can similarly be assembled in the framework of labelled objects (e.g. graphs with labelled vertices). This time the generating function of the class $\mathcal{C}$ is defined as $\widehat{C}(x) = \sum_{\gamma \in \mathcal{C}} \frac{x^{|\gamma|}}{|\gamma|!}$. The "labelled" Boltzmann distribution assigns to each object of $\mathcal{C}$ a weight

$$\mathbf{P}_x(\gamma) = \frac{x^{|\gamma|}}{|\gamma|!\widehat{C}(x)}.$$

Then, a Boltzmann sampler for the labelled class $\mathcal{C}$ is a procedure that draws objects of $\mathcal{C}$ at random under their "labelled" Boltzmann distribution. As in the unlabelled framework, the authors of [4] develop automatic rules of assemblage of Boltzmann samplers from basic combinatorial constructions

In this extended abstract, we detail in Section 3.2 the principle of assemblage of Boltzmann samplers for the case of a mixed combinatorial class. In a mixed class $\mathcal{C} = \cup_{n,m} \mathcal{C}_{n,m}$, an object has n labelled "atoms" and m unlabelled "atoms", for example a graph with n (labelled) vertices and m (unlabelled) edges. The associated generating function $C(x, y)$ is defined as $C(x, y) = \sum_{n,m} C_{n,m} \frac{x^n}{n!} y^m$ where $C_{n,m}$ is the number of objects of $\mathcal{C}$ with n labelled and m unlabelled atoms. For a fixed real value y_0, we denote by $\rho_C(y_0)$ the radius of convergence of $x \to C(x, y_0)$. A pair (x, y) is said to be *coherent* if $x \in (0, \rho_C(y))$, which means that $\sum_{n,m} C_{n,m} \frac{x^n}{n!} y^m$ converges and that $C(x, y)$ is well defined. Given a coherent pair (x, y), the *Boltzmann distribution* is the probability measure $\mathbf{P}_{x,y}$ such that an object γ with n labelled and m unlabelled atoms has probability

$$\mathbf{P}_{x,y}(\gamma) = \frac{1}{C(x, y)} \frac{x^n}{n!} y^m.$$

An important property of this measure is that two objects with the same size parameters have the same probability. A *Boltzmann sampler* $\Gamma C(x, y)$ is a program that produces objects of $\mathcal{C}$ at random under the Boltzmann distribution $\mathbf{P}_{x,y}$. Observe that the development of the Boltzmann framework for mixed classes is an extension of the two classical frameworks (i.e. labelled and unlabelled) studied in [4]. Indeed, the unlabelled case can be recovered by setting the variable x (marking labelled atoms) to 1, and the labelled case can be recovered by setting the variable y (marking unlabelled atoms) to 1.

3.2 Construction rules

A nice feature of Boltzmann samplers is that they can be obtained straightforwardly for finite sets, and that the basic combinatorial constructions (union, product, set) can be transposed into simple rules of construction for the associated Boltzmann samplers. Here we have to suppose that we know exactly the values of the generating functions at a given coherent pair. We will also need two basic distributions: given $0 < p < 1$, the Bernoulli Law $Bern(p)$ is given by the random variable X such that $\mathbf{P}(X = 1) = p$ and $\mathbf{P}(X = 0) = 1 - p$. Given $\lambda > 0$, the Poisson Law $Pois(\lambda)$ is given by the random variable X such that $\mathbf{P}(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}$.

Starting from combinatorial classes $\mathcal{A}$ and $\mathcal{B}$ for which we have valid Boltzmann samplers, we explain in Table 2 how to derive a valid Boltzmann sampler for a class $\mathcal{C}$ constructed from $\mathcal{A}$ and $\mathcal{B}$ using five fundamental rules (these are the rules we will use in this article):

Proposition 1 *For the five construction rules described in Table 2, the program $\Gamma C(x, y)$ is a valid Boltzmann sampler for the combinatorial class $\mathcal{C}$.*

Proof. Let us just detail the case of union. An object of $\mathcal{A}_{n,m}$ has probability $\frac{1}{A(x,y)} \frac{x^n}{n!} y^m$, by definition of $\Gamma A(x, y)$ multiplied by $\frac{A(x,y)}{C(x,y)}$, of being drawn by $\Gamma C(x, y)$. Hence it has probability $\frac{1}{C(x,y)} \frac{x^n}{n!} y^m$ of

	Construction	Generator $\Gamma C(x,y)$
union	$\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ $C(x,y) = A(x,y) + B(x,y)$	**if** *Bern* $\left(\frac{A(x,y)}{C(x,y)}\right)$ **then return** $\Gamma A(x,y)$ **else return** $\Gamma B(x,y)$ **endif**
product	$\mathcal{C} = \mathcal{A} \star \mathcal{B}$ $C(x,y) = A(x,y)B(x,y)$	$\gamma \leftarrow (\Gamma A(x,y), \Gamma B(x,y))$ {independent calls} DISTRIBUTERANDOMDISTINCTLABELS(γ) **return** γ
set	$\mathcal{C} = Set(\mathcal{A})$ $C(x,y) = \exp(A(x,y))$	$k \leftarrow$ *Pois*$(A(x,y))$ $\gamma \leftarrow (\Gamma A(x,y), \ldots, \Gamma A(x,y))$ {k independent calls} DISTRIBUTERANDOMDISTINCTLABELS(γ) **return** γ

x-substitution	$\mathcal{C} = \mathcal{A} \circ_x \mathcal{B}$
	$C(x,y) = A(B(x,y),y)$
		$\gamma \leftarrow \Gamma A(B(x,y),y)$
		for $v \in$ *labelled_atoms*(γ) **do**
		$\gamma_v \leftarrow \Gamma B(x,y)$ {independent calls}
		substitute v by γ_v in γ
		endfor
		DISTRIBUTERANDOMDISTINCTLABELS(γ)
		return γ

y-substitution	$\mathcal{C} = \mathcal{A} \circ_y \mathcal{B}$
	$C(x,y) = A(x,B(x,y))$
		$\gamma \leftarrow \Gamma A(x,B(x,y))$
		for $e \in$ *unlabelled_atoms*(γ) **do**
		$\gamma_e \leftarrow \Gamma B(x,y)$ {independent calls}
		substitute e by γ_e in γ
		endfor
		DISTRIBUTERANDOMDISTINCTLABELS(γ)
		return γ

Tab. 2: The transposition into Boltzmann samplers of 5 classical rules of construction of combinatorial classes.

being drawn. Similarly, an object of $\mathcal{B}_{n,m}$ has probability $\frac{1}{B(x,y)}\frac{x^n}{n!}y^m \cdot \left(1 - \frac{A(x,y)}{C(x,y)}\right) = \frac{1}{C(x,y)}\frac{x^n}{n!}y^m$ of being drawn. Hence $\Gamma C(x,y)$ is a valid Boltzmann sampler for $\mathcal{C}$. The proof for the four other cases is similar, still more intricate (the two substitution constructions, that are not in [4], are new). □

Example We take the example of the (unlabelled) class $\mathcal{C}$ of binary trees where the atoms are the inner nodes. The class $\mathcal{C}$ has the following decomposition grammar:

$$\mathcal{C} = (\mathcal{C} \cup \varnothing) \star \{\bullet\} \star (\mathcal{C} \cup \varnothing)$$

Hence the series $C(y)$ counting binary trees is given by $C(y) = y\,(1 + C(y))^2$. Thus $C(y)$ can be easily evaluated for a fixed real parameter $y < \frac{1}{4}$.

Using the construction rules for union and product given in Table 2, we obtain the following Boltzmann sampler for binary trees:

$\Gamma C(y)$: return $(\Gamma_{leafOrTree}(y), \{\bullet\}, \Gamma_{leafOrTree}(y))$ {independent calls}

$\Gamma_{leafOrTree}(y)$: if $Bern\left(\frac{1}{1+C(y)}\right)$ return $\varnothing$
else return $\Gamma C(y)$

Remark The function DISTRIBUTERANDOMDISTINCTLABELS(γ) throws distinct (uniformly) randomly permuted labels on the labellable atoms of γ. It is necessary to call this procedure on top of the combinatorial construction (for example "$return\ (\Gamma A(x), \Gamma B(x))$" for the cartesian product) to ensure that the atoms of the returned object bear distinct labels. If we consider a combinatorial class whose construction involves the 5 rules given in Table 2, the call to DISTRIBUTERANDOMDISTINCTLABELS can be postponed to the end of the algorithm, i.e. we can apply the labelling to the finally output object (this is also mentioned by Flajolet *et al* [5, Sec3]). Hence the labels do not really matter and introduce no additional complexity in the Boltzmann samplers: for a class $\mathcal{C}$ whose combinatorial decomposition involves these five construction rules, we just have to generate the (unlabelled) *shape* of an object γ produced by $\Gamma C(x,y)$; then we call DISTRIBUTERANDOMDISTINCTLABELS(γ).

Pointing In the following sections, we will make much use of the *pointing* operation: Given a mixed (or labelled) combinatorial class $\mathcal{C} = \cup_{n,m}\mathcal{C}_{n,m}$, the pointed class $\mathcal{C}^\bullet$ is defined as the class of objects of $\mathcal{C}$ with a marked labelled atom. As a consequence, the generating function of $\mathcal{C}^\bullet$ is $\sum_{n,m} nC_{n,m}\frac{x^n}{n!}y^m = x\frac{\partial C}{\partial x}(x,y)$. For the particular case of a class of planar graphs, we will also consider objects with a marked unlabelled atom, i.e. planar graphs with a pointed edge. This time the corresponding generating function is given by $\sum_{n,m} mC_{n,m}\frac{x^n}{n!}y^m = y\frac{\partial C}{\partial y}(x,y)$.

4 Decomposition of planar graphs and Boltzmann samplers

We present here a well known combinatorial decomposition of planar graphs (also used by Bodirsky *et al* [1]) according to successive levels of connectivity, and we adapt it to Boltzmann sampling. We recall that a graph is said to be 2-connected (resp. 3-connected) if at least 2 (resp. 3) of its vertices have to be removed to disconnect it. The decomposition can be summarized as follows: a planar graph can be decomposed into its connected components; and a connected planar graph can be seen as a decomposition tree in which the nodes are occupied by 3-connected planar graphs with a marked edge. Using the rules stated in Table 2, a topdown approach yields a chain of reductions. In this section, each reduction of the chain has a corresponding lemma, from Lemma 1 to Lemma 5. The concatenation of the reduction-lemmas finally gives the following Proposition:

Proposition 2 *Finding a Boltzmann sampler for labelled planar graphs comes down to finding a Boltzmann sampler for edge-pointed 3-connected planar graphs.*

4.1 Planar graphs from connected planar graphs

In this section, we consider Boltzmann samplers in one variable x marking the (labelled) vertices of the graphs. We recall that the rules of Table 2 are still valid when setting $y = 1$. We write $G(x) = \sum_n g_n\frac{x^n}{n!}$ and $C(x) = \sum_n c_n\frac{x^n}{n!}$ for the series counting respectively labelled planar graphs and connected labelled planar graphs by their number of vertices. A planar graph can be decomposed into the set of its connected components, which yields the equation $G(x) = \exp(C(x))$.

Lemma 1 *Finding a Boltzmann sampler $\Gamma G(x)$ for planar graphs reduces to finding a Boltzmann sampler $\Gamma C(x)$ for connected planar graphs.*

Proof. We use Rule 3 (set construction) of Table 2: a Poisson law of parameter $C(x)$ is used to draw the number k of connected components. Then we return a planar graph made of k independent calls to $\Gamma C(x)$. □

4.2 Connected from 2-connected planar graphs

We describe here a well-known decomposition, detailed in [9, p10]. It is called *block-decomposition* and establishes a relation between pointed connected and pointed 2-connected planar graphs. Each vertex-pointed connected planar graph can be uniquely constructed by composition in the following way: take a set of vertex-pointed 2-connected planar graphs and attach them, by merging their marked vertices into a unique marked vertex. Then for each non marked vertex v of each 2-connected component, take a vertex-pointed connected planar graph γ_v and merge the marked vertex of γ_v with v (this operation corresponds to an x-substitution). This construction implies the relation $C^\bullet(x) = x\exp(B'(C^\bullet))$ where $C^\bullet(x) := xC'(x)$ is the series counting vertex-pointed connected planar graphs.

Lemma 2 *Finding a Boltzmann sampler $\Gamma C(x)$ for connected planar graphs reduces to finding a Boltzmann sampler $\Gamma C^\bullet(x)$ for vertex-pointed connected planar graphs.*

Proof. We use the following algorithm with rejection, where we write $|\gamma|$ for the number of vertices of a graph γ:

$\Gamma C(x)$: $\quad \gamma \leftarrow \Gamma C^\bullet(x)$
if $Bern\left(\frac{1}{|\gamma|}\right)$ return γ else reject and restart

The probability for a graph γ to be drawn with $\Gamma C(x)$ is proportional to $|\gamma|\frac{x^{|\gamma|}}{|\gamma|!}$ (because of $\Gamma C^\bullet(x)$) multiplied by $\frac{1}{|\gamma|}$ (because of rejection). Hence it is proportional to $\frac{x^{|\gamma|}}{|\gamma|!}$, which ensures that $\Gamma C(x)$ is a valid Boltzmann sampler for connected planar graphs. □

Lemma 3 *Finding a Boltzmann sampler $\Gamma C^\bullet(x)$ for vertex-pointed connected planar graphs reduces to finding a Boltzmann sampler $\Gamma B^\bullet(x)$ for vertex-pointed 2-connected planar graphs.*

Proof. Using construction rules set and x-substitution of Table 2, the block decomposition explained above is directly transposed into the following Boltzmann sampler for vertex-pointed connected planar graphs:

$\Gamma C^{\bullet}(x)$: $k \leftarrow Pois(B'(C^{\bullet}(x))$
$\gamma \leftarrow (\Gamma B^{\bullet}(C^{\bullet}(x)), \ldots, \Gamma B^{\bullet}(C^{\bullet}(x)))$ $\{k$ independent calls$\}$
merge the k components of γ at their marked vertices
for each non marked vertex v of γ replace v by $\gamma_v \leftarrow \Gamma C^{\bullet}(x)$ {independent calls}
return γ.

4.3 2-connected from 3-connected planar graphs

A second well-known decomposition due to Trakhtenbrot [11], that we call *network-decomposition*, ensures that a 2-connected planar graph can be decomposed into 3-connected planar components. This combinatorial decomposition allows us to reduce the definition of a Boltzmann sampler for 2-connected planar graphs to the definition of a Boltzmann sampler for 3-connected planar graphs. We rely on [12] for the description of the decomposition. A *network* is a connected graph N with two poles labelled 0 and ∞, such that the graph N^* obtained by adding an edge between 0 and ∞ is a 2-connected planar graph. A *series-network* or s-network is a network made of at least 2 networks connected *in chain* at their poles. A *parallel network* or p-network is a network made of at least 2 networks connected *in parallel*, so that their respective ∞-poles and 0-poles coincide. A network N such that N^* is 3-connected is called a *pseudo-brick*. A *polyhedral network* or h-network is a network that can be obtained by substituting a network N_e in each edge e of a pseudo-brick (these networks will put the bridge between 2-connected and 3-connected planar graphs).

Proposition 3 (Trakhtenbrot) *Networks with at least 2 edges are partitioned into s-networks, p-networks and h-networks.*

Now we explain how to obtain a precise recursive decomposition and exact equations for the different families of networks. We write $D(x, y)$, $S(x, y)$, $P(x, y)$, $H(x, y)$ for the series counting respectively networks, s-networks, p-networks, h-networks by their number of non-pole vertices (variable x) and their number of edges (variable y). Proposition 3 ensures that:

$$D(x, y) = y + S(x, y) + P(x, y) + H(x, y) \tag{1}$$

An s-network can be uniquely decomposed into a non-s-network (the head of the chain) followed by a network (the trail of the chain):

$$S(x, y) = (y + P(x, y) + H(x, y))\, xD(x, y) \tag{2}$$

A p-network has a unique *maximal* parallel decomposition into a set of parallel components which are not p-networks. Observe that we consider here graphs without multiple edges, so that at most one of these components is an edge. Whether there is one or no such edge-component gives:

$$P(x, y) = y \exp_{\geq 1}(S(x, y) + H(x, y)) + \exp_{\geq 2}(S(x, y) + H(x, y)) \tag{3}$$

where $\exp_d(z) = \sum_{k \geq d} \frac{z^k}{k!}$.

Finally, the series for h-networks clearly corresponds to an y-substitution. If we write $G_3(x, y)$ for the series counting 3-connected labelled planar graphs, then the series counting pseudo-bricks is $\frac{2}{x^2}\frac{\partial}{\partial y}G_3(x, y)$, and we have:

$$H(x, y) = \frac{2}{x^2}\frac{\partial G_3}{\partial y}(x, D(x, y)) \tag{4}$$

Lemma 4 *Using rejection, a Boltzmann sampler $\Gamma B^{\bullet}(x)$ for (vertex-) pointed 2-connected planar graphs can be "efficiently" obtained, in an $\mathcal{O}(1)$ expected number of trials, from a Boltzmann sampler $\Gamma\frac{\partial B}{\partial y}(x, y)$ for edge-pointed 2-connected planar graphs.*

Proof. Once again, we use rejection. A Boltzmann sampler $\Gamma B^{\bullet}(x)$ is obtained as follows, where we write respectively i and j for the number of vertices and edges of a graph γ:

$\Gamma B^{\bullet}(x)$: $\gamma \leftarrow \Gamma\frac{\partial B}{\partial y}(x, 1)$
if $Bern\left(\frac{i}{j}\right)$ return γ else reject and restart

By construction, $\Gamma B^{\bullet}(x)$ draws a 2-connected planar graph γ with probability proportional to $j\frac{x^i}{i!}$ (because of $\Gamma\frac{\partial B}{\partial y}(x, 1)$) multiplied by $\frac{i}{j}$ (because of rejection). Hence it draws a 2-connected planar graph with probability proportional to $i\frac{x^i}{i!}$, which corresponds to a valid Boltzmann sampler for (vertex-) pointed 2-connected planar graphs. Let us now comment the word "efficient". The crucial point is that the graphs we consider are planar, so that Euler relation applies and gives $\frac{i}{j} \geq \frac{1}{3}$. Hence the probability of success at each trial is bounded away from 0 (this was not the case for the transition from $\Gamma C^{\bullet}(x)$ to $\Gamma C(x)$ described in Lemma 2). □

Lemma 5 *Finding a Boltzmann sampler* $\Gamma\frac{\partial B}{\partial y}(x,y)$ *for edge-pointed 2-connected planar graphs reduces to finding a Boltzmann sampler* $\Gamma\frac{\partial G_3}{\partial y}(x,y)$ *for edge-pointed 3-connected planar graphs.*

Proof. If we write $K(x,y)$ for the series counting networks where poles are not connected by an edge, we have both $\frac{x^2}{2}K(x,y) = \frac{\partial B}{\partial y}(x,y)$ and $(1+y)K(x,y) = 1 + D(x,y)$, so that $(1+y)\frac{\partial B}{\partial y}(x,y) = \frac{x^2}{2}(1+D(x,y))$. Hence, finding a Boltzmann sampler $\Gamma\frac{\partial B}{\partial y}(x,y)$ reduces to finding a Boltzmann sampler $\Gamma D(x,y)$ for networks.

Then, the combinatorial decomposition of networks, summarized by Equations 1-4, can be directly transposed into a Boltzmann sampler $\Gamma D(x,y)$ for networks, using the rules of construction of Table 2. The only terminal nodes of this decomposition grammar are the so-called pseudo-bricks. As we have seen, these objects correspond to edge-pointed 3-connected planar graphs, which concludes the proof. □

5 Boltzmann sampler for 3-connected planar graphs

The preceding section has ensured that the realization of a Boltzmann sampler for planar graphs comes down to the realization of a Boltzmann sampler for edge-pointed 3-connected planar graphs. This last task is possible since 3-connected planar graphs are combinatorially tractable.

A first well known result, due to Whitney, ensures that such graphs have a unique topological embedding (in general a planar graph can have many embeddings in the plane). More precisely, we define a *rooted 3-connected map* as an unlabelled 3-connected planar graph embedded in the plane, together with the choice of a marked and oriented edge, called the *root*. Writing $M(x,y) = \sum_{i,j} M_{i,j}x^iy^j$ for the series counting rooted 3-connected maps by their number of vertices and edges, Whitney's Theorem yields: $M(x,y) = 4y\frac{\partial G_3}{\partial y}(x,y)$. Hence, rooted 3-connected maps correspond to the unlabelled shape of edge-pointed 3-connected labelled planar graphs. In addition, according to the remark of Section 3.2, it is sufficient to draw only the unlabelled shape of the objects, so that we have the following lemma:

Lemma 6 *Finding a Boltzmann sampler* $\Gamma\frac{\partial G_3}{\partial y}(x,y)$ *for edge-pointed 3-connected planar graphs reduces to finding a Boltzmann sampler* $\Gamma M(x,y)$ *for rooted 3-connected maps.*

Now we use a combinatorial result (which relies on an explicit bijection) found by the author, D. Poulalhon and G. Schaeffer [6]. This result establishes a surprising correspondence between binary trees and rooted 3-connected maps. We define a *bicolored* binary tree as a binary tree (each node has a left son and a right son that are possibly empty) whose nodes are colored in black or white so that two adjacent nodes are always differently colored. These trees are partitioned into black-rooted and white-rooted depending on the color of their root node. Writing $T(x_\bullet, x_\circ)$, $T_\bullet(x_\bullet, x_\circ)$ and $T_\circ(x_\bullet, x_\circ)$ for the series counting respectively bicolored, black-rooted and white-rooted binary trees, we have:

$$\left\{\begin{array}{lcl} \mathcal{T} & = & \mathcal{T}_\bullet \cup \mathcal{T}_\circ \\ \mathcal{T}_\bullet & = & \{\bullet\} \star (\varnothing \cup \mathcal{T}_\circ)^2 \\ \mathcal{T}_\circ & = & \{\circ\} \star (\varnothing \cup \mathcal{T}_\bullet)^2 \end{array}\right. \qquad \left\{\begin{array}{lcl} T(x_\bullet, x_\circ) & = & T_\bullet(x_\bullet, x_\circ) + T_\circ(x_\bullet, x_\circ) \\ T_\bullet(x_\bullet, x_\circ) & = & x_\bullet\,(1 + T_\circ(x_\bullet, x_\circ))^2 \\ T_\circ(x_\bullet, x_\circ) & = & x_\circ\,(1 + T_\bullet(x_\bullet, x_\circ))^2 \end{array}\right.$$

Remark The classes of rooted planar maps and bicolored binary trees are unlabelled classes with two parameters (vertices and edges for maps, black and white nodes for binary trees). This case is not treated in Section 3, but it is clear that a Boltzmann sampler of parameter (x,y) has to be defined as a program that draws an object with i atoms of the first kind and j atoms of the second kind with probability $\frac{x^iy^j}{C(x,y)}$. With this definition, it is easy to establish that the construction rules given in Table 2 for union and product are valid.

Lemma 7 *The decomposition grammar for bicolored binary trees yields a* complete *Boltzmann sampler for bicolored binary trees, where "complete" means that no auxiliary Boltzmann sampler is needed.*

Now we state the combinatorial correspondence with rooted 3-connected maps, detailed in [6]:

Proposition 4 *Let $\mathcal{T}_{i,k}$ be the set of bicolored binary trees with i black nodes and k white nodes. There is a mapping, called closure-mapping, that establishes a bijection between $\mathcal{T}_{i,k} \times \{1,2,3,4,5,6\}$ and $\mathcal{D}_{i+3,k+3} \times \{1 \ldots i+k+2\}$, where $\mathcal{D}_{i+3,k+3}$ is a "small" superset of the set $\mathcal{M}_{i+3,i+k+4}$ of rooted 3-connected maps with $i+3$ vertices and $i+k+4$ edges.*

Lemma 8 *Setting $x_\bullet = x \cdot y$ and $x_\circ = y$, the correspondence of Proposition 4 transports a Boltzmann sampler $\Gamma T(x_\bullet, x_\circ)$, as defined in Lemma 7, into a sampler for rooted 3-connected maps such that the probability of drawing an object of $\mathcal{M}_{i,j}$ is proportional to $(j-2)x^i y^j$.*

Adding a rejection step with probability of success equal to $\frac{1}{j-2}$ on top of this sampler, we obtain a Boltzmann sampler for rooted 3-connected maps.

This lemma ends our decomposition chain and also ends the corresponding chain of lemmas, which indicate to us how to realize a Boltzmann sampler for planar graphs. However more is needed to achieve the complexity stated in Theorem 1, as explained in the next section.

6 Size distribution

In the last section, we have described a procedure for finding a Boltzmann sampler $\Gamma G(x)$ for labelled planar graphs. We are interested in the distribution of size of the planar graphs output by $\Gamma G(x)$. Typically, we need to *tune* the real parameter x in order to ensure that the distribution of the size of the object produced is concentrated around a specific target value n. However, this tuning operation does not always apply depending on the *singularity type* of $G(x)$.

Definition Given $\alpha \in \mathbf{R}\backslash\mathbf{N}$, a generating function $F(x)$ is said α-singular if the following expansion holds in a Camembert-neighbourhood of its dominant singularity ρ_F (see [4] for technical conditions of such neighbourhoods),

$$F(x) \underset{x \to \rho_F}{=} P(x) + c_\alpha \left(1 - \frac{x}{\rho_F}\right)^\alpha + o\left(1 - \frac{x}{\rho_F}\right)^\alpha ,$$

where $P(x)$ is a polynomial.

The following lemma, Theorem 6.3 of [4], ensures that the tuning operation mentioned above applies well for α-singular functions with $\alpha < 0$.

Lemma 9 [4] *Let there be given a Boltzmann sampler $\Gamma F(x)$ for a combinatorial class and assume that the associated generating function $F(x)$ is α-singular with $\alpha < 0$. For each integer n, define $x_n = \rho_F\left(1 + \frac{\alpha}{n}\right)$ and denote by X the size of an object output by $\Gamma F(x_n)$.*

Then, for each fixed tolerance-ratio $\epsilon > 0$, we have

$$\mathbf{P}\left(X \in [n(1-\epsilon), n(1+\epsilon)]\right) \to_{n\to\infty} p_\epsilon ,$$

where the positive constant $p_\epsilon \in]0,1]$ varies in proportion to ϵ: $p_\epsilon \sim_{\epsilon\to 0} \sigma \cdot \epsilon$ for some constant σ.

Moreover, we have

$$\mathbf{P}\left(\Sigma_n = n\right) \sim_{n\to\infty} \frac{\sigma}{n} .$$

Now the following lemma indicates how to modify the Boltzmann sampler $\Gamma G(x)$ for planar graphs so that the size distribution of the output gets the behaviour required by Lemma 9.

Lemma 10 *Given a combinatorial class whose associated generating function $F(x)$ is α-singular, the generating function $F^\bullet(x) = xF'(x)$ associated to the pointed class $F^\bullet$ is $(\alpha - 1)$-singular.*

The generating function $G(x)$ counting planar graphs is $\frac{5}{2}$-singular, see [8].

As a consequence, the generating function $G^{\bullet\bullet\bullet}(x)$ is $\left(-\frac{1}{2}\right)$-singular. Hence, Lemma 9 applies for the size distribution of the output of $\Gamma G^{\bullet\bullet\bullet}(x)$.

Observe that the pointing operator can be easily injected in the basic rules of construction that we use for the decomposition of planar graphs. For example, if $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ then $\mathcal{C}^\bullet = \mathcal{A}^\bullet \cup \mathcal{B}^\bullet$; if $\mathcal{C} = \mathcal{A} \star \mathcal{B}$ then $\mathcal{C}^\bullet = \mathcal{A}^\bullet \star \mathcal{B} \cup \mathcal{A} \star \mathcal{B}^\bullet$; if $\mathcal{C} = Set(\mathcal{A})$ then $\mathcal{C}^\bullet = \mathcal{A}^\bullet \star Set(\mathcal{A})$. As a consequence, the pointing operator can be injected in the chain of reductions from planar graphs to 3-connected planar graphs. For example, Lemma 3 becomes:

Lemma 11 *Finding Boltzmann samplers $\Gamma G^{\bullet\bullet\bullet}(x)$, $\Gamma G^{\bullet\bullet}(x)$, $\Gamma G^\bullet(x)$, $\Gamma G(x)$ for planar graphs reduces to finding Boltzmann samplers $\Gamma C^{\bullet\bullet\bullet}(x)$, $\Gamma C^{\bullet\bullet}(x)$, $\Gamma C^\bullet(x)$, $\Gamma C(x)$ for connected planar graphs.*

Proof. Starting from $G(x) = \exp(C(x))$ (i.e. $\mathcal{G} = Set(\mathcal{C})$), we find successively $G^\bullet(x) = C^\bullet(x)\exp(C(x))$, $G^{\bullet\bullet}(x) = C^{\bullet\bullet}(x)\exp(C(x)) + C^\bullet(x)^2\exp(C(x))$, $G^{\bullet\bullet\bullet}(x) = C^{\bullet\bullet\bullet}(x)\exp(C(x)) + C^{\bullet\bullet}(x)C^\bullet(x)\exp(C(x)) + 2C^{\bullet\bullet}(x)C^\bullet(x)\exp(C(x)) + C^\bullet(x)^3\exp(C(x))$. The corresponding combinatorial decompositions are then directly transposed into Boltzmann samplers, using the rules of Table 2. □

In the same way, the pointing operator can be injected into the decomposition from connected to 2-connected and into the decomposition from 2-connected to 3-connected planar graphs. This yields:

Proposition 5 *Finding a Boltzmann sampler $\Gamma G^{\bullet\bullet\bullet}(x)$ reduces to finding Boltzmann samplers $\Gamma \frac{\partial G_3}{\partial y}(x, y)$, $\Gamma \frac{\partial G_3}{\partial y}^\bullet(x, y)$, $\Gamma \frac{\partial G_3}{\partial y}^{\bullet\bullet}(x, y)$. According to Whitney's Theorem (see Section 5), this comes down to finding Boltzmann samplers $\Gamma M(x, y)$, $\Gamma M^\bullet(x, y)$, $\Gamma M^{\bullet\bullet}(x, y)$ for non-pointed, vertex-pointed and vertex-bi-pointed rooted 3-connected maps.*

From Lemma 8, we already have a Boltzmann sampler $\Gamma M(x, y)$, and the following lemma completes the construction for pointed and bi-pointed objects:

Proposition 6 *Using the correspondence* "binary-trees↔rooted-3-connected-maps" *stated in Proposition 4 and using rejection, Boltzmann samplers $\Gamma M^\bullet(x, y)$ and $\Gamma M^{\bullet\bullet}(x, y)$ can be "efficiently" obtained, in an $\mathcal{O}(1)$ number of trials, from Boltzmann samplers $\Gamma T(xy, y)$ and $\Gamma T^\bullet(xy, y)$ of bicolored and black-node-pointed bicolored binary trees. As T and $T^\bullet$ have simple complete decomposition grammars, complete Boltzmann samplers can be directly derived for these two classes.*

Hence, Proposition 5 ensures that a Boltzmann sampler $\Gamma G^{\bullet\bullet\bullet}(x)$ for triply pointed planar graphs can be obtained.

Proof. Let us detail the case $\Gamma T(xy, y) \to \Gamma M^\bullet(x, y)$. As stated in Lemma 8, the correspondence *binary trees↔rooted 3-connected maps* yields a sampler for rooted 3-connected maps where each object with i vertices and j edges has probability proportional to $(j-2)x^i y^j$. It just remains to pile up on top of this sampler a rejection step with success-probability $\frac{i}{j-2}$. As opposed to Lemma 8, the probability of success is bounded away from 0 because Euler relation ensures that $\frac{i}{j-2} \geq \frac{1}{3}$. □

To conclude, we have to point the objects so that the size distribution of the outputs of Boltzmann samplers has rather good concentration properties. Then it is possible to inject the pointing operator into the decomposition of planar graphs and to obtain a Boltzmann sampler $\Gamma G^{\bullet\bullet\bullet}(x)$ for triply vertex-pointed planar graphs, which have a satisfactory size-distribution. We have also seen that the rejection step that we add on top of our samplers (in particular for rooted 3-connected maps in Proposition 6) works better for pointed objects than for non-pointed objects.

7 Algorithm scheme and Complexity results

The sampler we finally propose in order to produce planar graphs is the "triply pointed" Boltzmann sampler $\Gamma G^{\bullet\bullet\bullet}(x_n)$ with the value $x_n = \rho_G\left(1 - \frac{1}{2n}\right)$ tuned as indicated in Lemma 9. The complete scheme, from binary trees to triply vertex-pointed planar graphs, is recapitulated on Figure 2 and Figure 3. The following proposition implies directly the time complexity stated in Theorem 1.

Proposition 7 *Let Λ_n be the expected running time of $\Gamma G^{\bullet\bullet\bullet}(x_n)$ where $x_n = \rho_G\left(1 - \frac{1}{2n}\right)$. Let $\epsilon > 0$ be a fixed size-tolerance parameter.*

- *The quantity Λ_n is linearly bounded: $\Lambda_n = \mathcal{O}(n)$ as $n \to \infty$.*
- *The expected running time of $\Gamma G^{\bullet\bullet\bullet}(x_n)$ conditioned (by rejection) to output an object of size $\Sigma = n$ is quadratic. More precisely, it is asymptotically $\frac{n}{\sigma}\Lambda_n$, where the constant σ is introduced in Lemma 9.*
- *The expected running time of $\Gamma G^{\bullet\bullet\bullet}(x_n)$ conditioned (by rejection) to output an object of size $\Sigma \in [n(1-\epsilon, n(1+\epsilon)]$ is linear. More precisely, it is asymptotically $\frac{1}{p_\epsilon}\Lambda_n$, where the positive constant p_ϵ is introduced in Lemma 9.*

Proof. The second and third points are trivial using Lemma 9 and the following easy technical result: if a rejection algorithm A has expected running time τ and success-probability p at each trial, then the expected running time till success is $\frac{\tau}{p}$. The third point is more difficult. A complete proof requires to develop calculation rules for the expected number $\Lambda C(x)$ of operations performed during one call to a Boltzmann sampler $\Gamma C(x)$. For a class $\mathcal{C}$ assembled recursively using classical combinatorial constructions (like

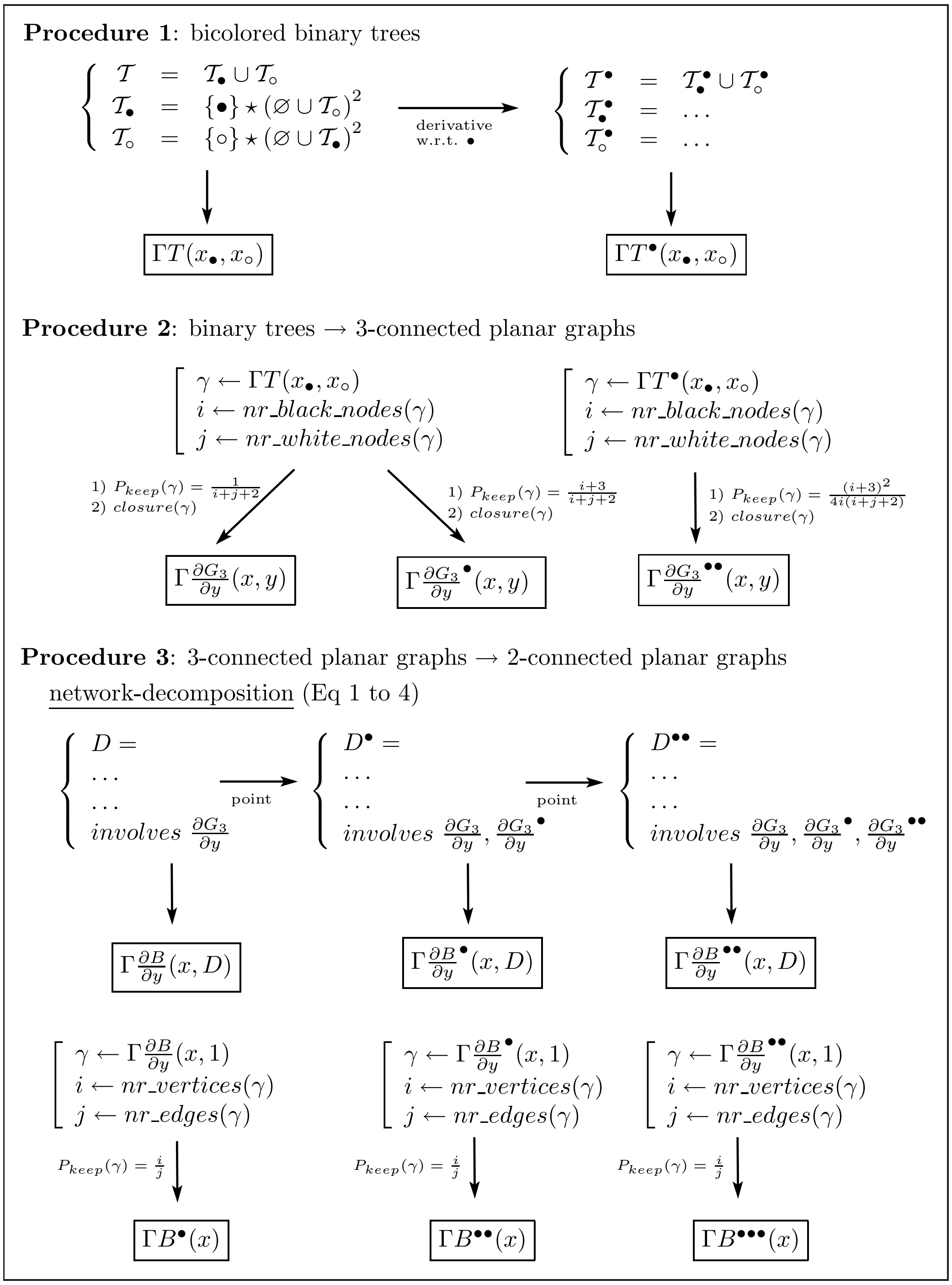

Fig. 2: The algorithmic scheme producing Boltzmann samplers for 2-connected planar graphs from Boltzmann samplers for bicolored binary trees.

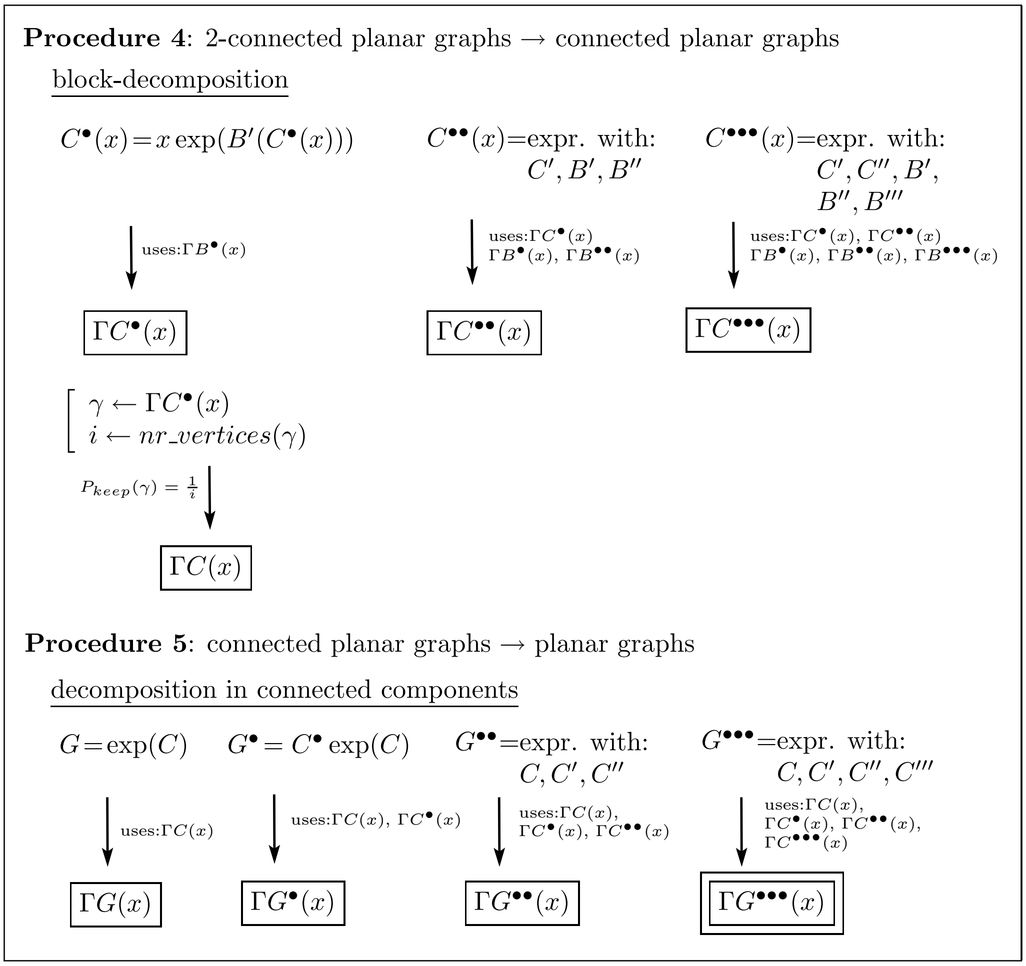

Fig. 3: The algorithmic scheme producing a Boltzmann sampler for triply vertex-pointed planar graphs from Boltzmann samplers for 2-connected planar graphs.

union, product), simple rules of composition can be developed for the calculation of $\Lambda C(x)$. For example, if $\mathcal{C} = \mathcal{A} \star \mathcal{B}$ then $\Lambda C(x) = \Lambda A(x) + \Lambda B(x)$, if $\mathcal{C} = \mathcal{A} \cup \mathcal{B}$ then $\Lambda C(x) = \frac{A(x)}{C(x)} \Lambda A(x) + \frac{B(x)}{C(x)} \Lambda B(x)$. Other simple rules can be derived for the three other construction (set, x-substitution, y-substitution) used by our algorithm. Injecting these rules of calculation into the successive decomposition grammars used by our algorithm (see Figure 2 for a summary), we finally obtain that $\Lambda G^{\bullet\bullet\bullet}(x_n)$ is $\mathcal{O}(n)$.

Let us now give an intuitive explanation of the linear complexity result. All operations used in the algorithm (assemblage of connected components, closure-mapping ...) have linear cost. Hence the steps that can make the complexity of the algorithm increase are the rejection steps. For example, the transition from $\Gamma C^{\bullet}(x)$ to $\Gamma C(x)$ requires a rejection step where the accepting-probability is $\frac{1}{i}$ with i the number of vertices of the object. It seems that rejection arises very often if the expected number of vertices Σ_n of an output of $\Gamma C^{\bullet}(x_n)$ is of order n. Fortunately this is not the case because the size distribution of the output of $\Gamma C^{\bullet}(x_n)$ is concentrated on objects of small size, so that we have $\Sigma_n = \mathcal{O}(1)$. In our algorithm, there are also rejection steps where the expected size of the objects to reject is large, in $\mathcal{O}(n)$. This is for example the case for the transition from $\Gamma \frac{\partial B}{\partial y}^{\bullet\bullet}(x, 1)$ to $\Gamma B^{\bullet\bullet\bullet}(x)$. However, see the proof of Lemma 4, the acceptance probability is greater than $\frac{1}{3}$, so that this rejection step does not make the complexity order increase. To sum up, the rejection steps involving large objects are always such that the acceptance probability is bounded away from 0. □

Acknowledgements

I am very grateful to my two advisors Philippe Flajolet and Gilles Schaeffer for their encouragements, very instructive discussions, and useful corrections and suggestions. I also thank Omer Giménez and Marc Noy for fruitful interactions and great help in starting the implementation of the algorithm.

References

[1] Manuel Bodirsky, Clemens Gröpl, and Mihyun Kang. Generating labeled planar graphs uniformly at random. In *Thirtieth International Colloquium on Automata, Languages and Programming, Springer Verlag, LNCS 2719*, pages 1095–1107, 2003.

[2] N. Bonichon, C. Gavoille, N. Hanusse, D. Poulalhon, and Schaeffer G. Planar graphs, via well-orderly maps and trees. In 30^{th} *International Workshop, Graph - Theoretic Concepts in Computer Science (WG)*, volume 3353 of *Lecture Notes in Computer Science*, pages 270–284. Springer-Verlag, 2004.

[3] Alain Denise, Marcio Vasconcellos, and Dominic J.A. Welsh. The random planar graph. *Congressus Numerantium*, 113:61–79, 1996.

[4] Philippe Duchon, Philippe Flajolet, Guy Louchard, and Gilles Schaeffer. Boltzmann samplers for the random generation of combinatorial structures. *Combinatorics, Probability and Computing*, 13(4–5):577–625, 2004. Special issue on Analysis of Algorithms.

[5] Philippe Flajolet, Paul Zimmerman, and Bernard Van Cutsem. A calculus for the random generation of labelled combinatorial structures. *Theoretical Computer Science*, 132(1-2):1–35, 1994.

[6] Éric Fusy, D. Poulalhon, and G. Schaeffer. Dissections and trees, with applications to optimal mesh encoding and to random sampling. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2005.

[7] S. Gerke, C. McDiarmid, A. Steger, and A. Weissl. Random planar graphs with a fixed number of edges. In *16th Annual ACM-SIAM Symposium on Discrete Algorithms*, January 2005.

[8] Omer Gimenez and Marc Noy. Asymptotic enumeration and limit laws of planar graphs, 2004. 14 pages, hrefhttp://arxiv.org/abs/math.CO/0501269math.CO/0501269.

[9] F. Harary and E. Palmer. *Graphical Enumeration*. Academic Press, New York, 1973.

[10] A. Nijenhuis and Herbert S. Wilf. *Combinatorial algorithms*. Academic Press Inc., 1979.

[11] B. A. Trakhtenbrot. Towards a theory of non–repeating contact schemes (russian). In *Trudi Mat. Inst. Akad. Nauk SSSR 51*, pages 226–269, 1958.

[12] T. R. S. Walsh. Counting labelled three-connected and homeomorphically irreducible two-connected graphs. *J. Combin. Theory*, 32(B):1–11, 1982.