RANDOM SAMPLING OF PLANE PARTITIONS

OLIVIER BODINI, ÉRIC FUSY, AND CARINE PIVOTEAU

ABSTRACT. This article introduces random generators of plane partitions. Combining a bijection of Pak with the framework of Boltzmann samplers, we obtain random samplers that are slightly superlinear: the complexity is \( O(n \ln(n)^3) \) in approximate-size sampling and \( O(n^{4/3}) \) in exact-size sampling (under a real-arithmetic computation model). To our knowledge, these are the first polynomial-time samplers for plane partitions according to the size (there exist polynomial-time samplers of another type, which draw plane partitions that fit inside a fixed bounding box). The same principles yield efficient samplers for \((a \times b)\)-boxed plane partitions (plane partitions with two dimensions bounded), and for skew plane partitions. The random samplers allow us to perform simulations and observe limit shapes and frozen boundaries, which have been analysed recently by Cerf and Kenyon for plane partitions, and by Okounkov and Reshetikhin for skew plane partitions.

INTRODUCTION

Plane partitions, originally introduced by A. Young [24], constitute a natural generalization of integer partitions in the plane, as they consist of a matrix of integers that are non-increasing in both dimensions (whereas an integer partition is an array of non-increasing integers). In addition, they also have a nice interpretation in 3D-space as a heap of cubes (see Figure 2). Plane partitions have motivated a huge literature in numerous fields of mathematics [2, 10, 11, 15, 23] and statistical physics [14, 21], and have provided crucial insight for solving challenging problems in combinatorics [25], see [3] for a detailed historical account. The problem of enumerating plane partitions was solved by Mac Mahon, who proved the beautiful formula

\[
P(x) = \prod_{r \geq 1} (1 - x^r)^{-r}
\]

for the generating function. The simplicity of the formula asks for a combinatorial interpretation. A first completely bijective proof has been given by Krattenthaler [12]. The principle is inspired by the seminal bijection of Novelli-Pak-Stoyanovskii [16] giving an interpretation of the hook-length formula. In [12], Krattenthaler also discusses as application of his bijection a polynomial-time algorithm for the random generation of plane partitions in a given box \(a \times b \times c\). Upon looking at the heap of cubes in the \((1, 1, 1)\) direction, this task is equivalent to sampling hexagon tilings by rhombi for a hexagon of side lengths \((a, b, c, a, b, c)\); there also exist random samplers for such tilings, which rely either on “coupling from the past principles” [20] or on “determinant algorithms” [22]. In contrast, we are interested here in sampling plane partitions uniformly at random with respect to the size, defined as the sum of the matrix entries. For this purpose, we use another bijective interpretation of Mac Mahon’s formula recently given by Pak [19].

Let us briefly mention the motivations for having a random sampler for plane partitions according to the size. The size is a natural parameter, as it corresponds to the volume of the plane partition (number of cubes) in the 3D interpretation. Recently, several authors have studied with great attention the statistical properties of plane partitions with respect to the size. In particular, Cerf and Kenyon [4]

\[\text{Date: June 9 2006.}\]
have determined the asymptotic shape under fixed-size distribution (the asymptotic shape in the boxed framework—hexagon tilings—is due to Cohn, Larsen, and Propp [5]). Even more recently, Okounkov and Reshetikhin, using a method based on Schur processes, have rediscovered the limit shape of Cerf and Kenyon [17]. They have studied in a subsequent article [18] the local correlations and limit shapes for plane partitions under a mixed model: the plane partition is constrained to a 2-coordinate box \(a \times b\) and is drawn under the Boltzmann model with respect to the size. (We will also describe random samplers for this mixed model.) In addition, physicists have developed new models relying on plane partitions, giving rise to a simplified version of the 3-dimensional models of lattice vesicles [13]. Plane partitions are also related to the 3-dimensional Ising model in the cubic lattice [4]. In general, physicists are interested in checking experimentally or conjecturing some limit properties of these models, by generating very large random objects.

For this purpose, this paper introduces efficient samplers for plane partitions. Our approach consists in combining methods from bijective combinatorics and symbolic combinatorics. Precisely, we derive from the bijection of Pak [19]—upon minor simplifications and adopting a specific terminology— an explicit algorithm that maps a multiset of integer pairs (the class is denoted by \(M\)) to a plane partition with the same size. Pak’s bijection reduces the task of finding a sampler for plane partitions to the task of finding a sampler for \(M\). As the class \(M\) has an explicit simple combinatorial decomposition, it is suitable for methods of symbolic combinatorics: we adopt the framework of Boltzmann samplers, introduced in [6] and further developed in [7].

The specificity of Boltzmann samplers is that the probability is spread over all objects of the class: an object of size \(n\) has probability proportional to \(x^n\), where \(x\) is a fixed real parameter. In particular, as two objects having the same size have equal probability, the probability distribution restricted to a given size \(n\) is uniform. As we are interested in generating very large plane partitions, the Boltzmann framework is suitable, due to the complexity gain obtained by relaxing the exact-size constraint. The articles [6] and [7] provide a collection of rules for building a sampler for a class admitting a decomposition involving classical constructions. Using these rules, the decomposition of \(M\) is readily translated to a Boltzmann sampler. This yields, via Pak’s bijection, a Boltzmann sampler for plane partitions. In addition, as the size distribution of plane partitions—under the Boltzmann model—has good concentration properties, it is possible to “tune” the parameter \(x\) so as to draw objects of size around (or exactly at) a given target value \(n\). With the parameter \(x\) suitably tuned and a rejection loop targeted at the size, we obtain a quasi-linear time approximate-size sampler for plane partitions: for any tolerance-ratio \(\epsilon > 0\), our sampler draws a plane partition of size in \([n(1 - \epsilon), n(1 + \epsilon)]\) with an expected complexity \(O(n \ln(n)^3)\). The same principles, with the rejection loop running until a given size \(n\) is attained, yields an exact-size sampler for plane partitions, with expected complexity \(O(n^{4/3})\). To our knowledge, our algorithm is the first exact-size sampler for plane partitions with expected polynomial complexity\(^1\). This allows us to generate objects of size up to \(10^7\) in a few minutes on a PC. The same principles (i.e., Pak’s bijection + Boltzmann sampler) yield efficient Boltzmann samplers for \((a \times b)\)-boxed plane partitions (plane partitions whose non-zero entries lie in a \((a \times b)\) rectangle), which are those considered by Okounkov and Reshetikhin. We obtain for boxed plane partitions an approximate-size sampler with expected

\(^1\)A polynomial sampler might also be obtained by applying the recursive method of sampling instead of Boltzmann sampling, yet this would require further extension of the recursive framework, and the complexity should be at least quadratic.
constant complexity — the constant depends on the rectangle and tolerance $\epsilon$ but not on the size $n$ — and a linear exact-size sampler.

Proving the correct complexity order of the samplers is the major technical difficulty we have to deal with. This leads us to analyse the distribution of a natural length-parameter of a plane partition, which is the maximum hook length (abscissa + ordinate + 1) over all nonzero entries of the matrix. Let us mention that the asymptotic behaviour of such a parameter can not be derived from the convergence results of Cerf and Kenyon [4]; the main reason is that the length parameter we analyse tends to infinity even when the plane partitions is normalised by its volume. Also our methods of analysis differ greatly from [4]. We take advantage of two powerful techniques of analytic combinatorics [8]: the saddle-point method for finding the asymptotic order of the length parameter under fixed size distribution; and Mellin transform for analysing the expected complexities of the samplers under Boltzmann probability, which are naturally expressed as harmonic sums.

Outline of the paper. After some definitions (Section 1) about combinatorial classes and plane partitions, we introduce in Section 2 the algorithmic version of Pak’s bijection, which induces a combinatorial isomorphism between the set of plane partitions and the class $\mathcal{M} := \text{MSet}(\mathbb{Z} \times \text{Seq}(\mathbb{Z}))^2$. Section 3 recalls basic principles of Boltzmann sampling, in particular the sampling rules associated to the constructions appearing in the specification of $\mathcal{M}$. The Boltzmann sampler for $\mathcal{M}$, as well as $\mathcal{M}_{a,b} := \text{MSet}(\mathbb{Z} \times \text{Seq}_{<a}(\mathbb{Z}) \times \text{Seq}_{<b}(\mathbb{Z}))$, is derived in Section 4, giving rise to Boltzmann samplers for plane partitions and $(a \times b)$-boxed plane partitions. We explain then briefly how the principles extend to obtain Boltzmann samplers for so-called skew plane partitions. As explained in Section 4.4, suitable choices of the parameter $x$ in the Boltzmann samplers yield in turn efficient samplers for plane partitions targetted exactly or approximately at a given size $n$ (precise statements are given in Theorems 4.4 and 4.5). The expected complexities of the Boltzmann samplers and targetted samplers are then analysed in Section 5.

1. Definitions

A combinatorial class is a pair $(\mathcal{A}, |.|)$ where $\mathcal{A}$ is a set and $|.|$ is a function from $\mathcal{A}$ to $\mathbb{N}$, called size function, such that the number of elements of any given size is finite. Observe that the property of $|.|$ implies that the set $\mathcal{A}$ is finite or denumerable. Using the size function, we can graduate $\mathcal{A}$ as $\mathcal{A} = \bigcup_n A_n$, where $A_n$ is the set of objects of $\mathcal{A}$ that have size $n$. In the sequel, we denote by $A_n$ the cardinality of $A_n$. To each combinatorial class $\mathcal{A}$, we associate the generating function $A(z) = \sum A_n z^n$.

Two combinatorial classes $(\mathcal{A}, |.|_{\mathcal{A}})$ and $(\mathcal{B}, |.|_{\mathcal{B}})$ are said to be combinatorially isomorphic $(\mathcal{A} \simeq \mathcal{B})$, if and only if there exists a one-to-one map from $\mathcal{A}$ to $\mathcal{B}$ that preserves the size. Let us notice that two classes $\mathcal{A}$ and $\mathcal{B}$ are isomorphic if and only if their generating functions are equal.

Here are some classical constructions on combinatorial classes that will be used in this paper. Notations and rules are summarized in Figure 1 (a more general presentation can be found in [8]):

- $\mathcal{E}$ and $\mathcal{Z}$ are atoms of size 0 and 1.
- Disjoint union $\mathcal{A} + \mathcal{B}$: the union of two distinct copies of $\mathcal{A}$ and $\mathcal{B}$.
- Cartesian product $\mathcal{A} \times \mathcal{B}$: the set of pairs $(\alpha, \beta)$ where $\alpha \in \mathcal{A}$ and $\beta \in \mathcal{B}$.

Given a class $\mathcal{A}$ not containing empty atoms,

- Sequence: $\text{Seq}(\mathcal{A})$ is the class of finite sequences of objects of $\mathcal{A}$.
- Multiset: $\text{MSet}(\mathcal{A})$ is the class of finite sets of objects of $\mathcal{A}$, with repetitions allowed.
Observe that, in a multiset $\mu \in \text{MSet}(A)$, each element $\alpha \in A$ has a multiplicity $c_\alpha \geq 0$. Hence, if $A$ is a finite set,

$$\text{MSet}(A) \simeq \prod_{\alpha \in A} \text{Seq}(\alpha).$$

### Table: Some constructions on combinatorial classes.

<table>
<thead>
<tr>
<th>Class</th>
<th>Generating function</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C = \emptyset$</td>
<td>$C(z) = 1$</td>
<td>neutral object of size 0</td>
</tr>
<tr>
<td>$C = Z$</td>
<td>$C(z) = z$</td>
<td>atom of size 1</td>
</tr>
<tr>
<td>$C = A + B$</td>
<td>$C(z) = A(z) + B(z)$</td>
<td>disjoint union</td>
</tr>
<tr>
<td>$C = A \times B$</td>
<td>$C(z) = A(z) \times B(z)$</td>
<td>cartesian product</td>
</tr>
<tr>
<td>$C = \text{Seq}(A)$</td>
<td>$C(z) = (1 - A(z))^{-1}$</td>
<td>$\mathcal{E} + A + A \times A + A \times A + \ldots$</td>
</tr>
<tr>
<td>$C = \text{MSet}(A)$</td>
<td>$C(z) = \exp(\sum(1/k)A(z^k))$</td>
<td>a multiset of elements of $A$</td>
</tr>
</tbody>
</table>

**Figure 1.** Some constructions on combinatorial classes.

A **plane partition** (Figure 2) of $n$ is a two-dimensional array of integers $(a_{i,j})_{\mathbb{N}^2}$ that are non-increasing both from left to right and bottom to top and that add up to $n$. In other words,

$$a_{i,j} \geq a_{i,j+1}, \ a_{i,j} \geq a_{i+1,j} \ \forall (i,j) \in \mathbb{N}^2 \ \text{and} \ \sum_{i,j} a_{i,j} = n.$$  

We denote by $\mathcal{P}$ the combinatorial class of plane partitions, endowed with the size function $|\langle (a_{i,j})_{\mathbb{N}^2} \rangle| = \sum_{i,j} a_{i,j}$. Plane partitions have a natural representation in 3D-space as a heap of cubes with non-increasing height in the direction of the $x$-axis and $y$-axis, see Figure 2. Observe that the size of the plane partition exactly corresponds to the number of cubes in the 3D-representation.

The **bounding rectangle** of a plane partition $(a_{i,j})_{\mathbb{N}^2}$ is the smallest double range $R = [0..\ell-1] \times [0..w-1]$ such that $a_{i,j} = 0$ for all index pairs $(i,j)$ outside of $R$. A $(a \times b)$-boxed plane partition is a plane partition whose bounding rectangle is at most $a \times b$. Equivalently, $a_{i,j}$ is null for any $(i,j)$ such that $i \geq a$ or $j \geq b$. We denote by $\mathcal{P}_{a,b}$ the class of $(a \times b)$-boxed plane partitions. Define the two combinatorial classes $\mathcal{M}$ and $\mathcal{M}_{a,b}$ as follows, where $\text{SEQ}_{<d}(A)$ denotes the class of sequences of at most $d - 1$ elements of $A$.

$$\mathcal{M} := \text{MSet}(\mathcal{Z} \times \text{SEQ}(\mathcal{Z})^2)$$

$$\mathcal{M}_{a,b} := \text{MSet}(\mathcal{Z} \times \text{SEQ}_{<a}(\mathcal{Z}) \times \text{SEQ}_{<b}(\mathcal{Z})).$$

Classically, $\text{SEQ}(\mathcal{Z})$ is identified with the class of nonnegative integers, so that we can specify $\mathcal{M}$ with the following simplified notation,

$$\mathcal{M} \simeq \text{MSet}(\mathcal{Z} \times \mathbb{N}^2).$$

Our main goal in the next section is to show how Pak’s bijection yields an explicit combinatorial isomorphism between $\mathcal{P}$ and $\mathcal{M}$. For this purpose, we introduce some more terminology. The **diagram** $D$ of an element $M \in \mathcal{M}$ is a two-dimensional array $(m_{i,j})_{\mathbb{N}^2}$ (with $(0,0)$ at the bottom left) where $m_{i,j}$ is the multiplicity of $(\mathcal{Z}, i,j)$
in $M$ (see the transition between the first two pictures of Figure 3). The size of
$D$ is defined as $|D| = \sum_{i,j} m_{i,j}(i + j + 1)$, so that it corresponds to the size
of the multiset in $M$. The bounding rectangle of $M$ is defined similarly as for plane
partitions: it is the smallest double range $R = [0, \ell - 1] \times [0, w - 1]$ such that all
entries of $D$ outside of $R$ are zero. The integers $\ell$ and $w$ are respectively called the
length and the width of $D$.

2. Pak’s bijection

In [19], Pak presents a bijection between a plane partition bounded in a shape $\mu$
($\mu$ being a Ferrer diagram) and a filling of the entries of $\mu$ with nonnegative inte-
gers. We give a slightly revisited version of this bijection as an algorithm realising
explicitly the combinatorial isomorphism $M \simeq P$, see Figure 3 for an example.

Algorithm 1: Pak’s algorithm

Input : a diagram $D$ of a multiset in $M$.
Output: a plane partition.
Let $\ell$ be the length and $w$ be the width of $D$.
for $i := \ell - 1$ downto $0$ do
for $j := w - 1$ downto $0$ do
$D[i,j] \leftarrow D[i,j] + \max(D[j+1,i], D[i,j+1]);$
for $c := 1$ to $\min(w - 1 - j, \ell - 1 - i)$ do
$x \leftarrow i + c; y \leftarrow j + c;$
$D[x,y] \leftarrow \max(D[x+1,y], D[x,y+1]) + \min(D[x-1,y], D[x,y-1]) - D[x,y];$
Return $D$.

Proposition 2.1 (Pak [19]). Algorithm 1 yields an explicit size-preserving bijection
between the class $M_{a,b}$ and the class of $(a \times b)$-plane partition. In other words, the
algorithm realizes the combinatorial isomorphism

\begin{equation}
\mathcal{P}_{a,b} \simeq \text{MSet}(\mathcal{Z} \times \text{SEQ}_{<a}(\mathcal{Z}) \times \text{SEQ}_{<b}(\mathcal{Z})).
\end{equation}

Proof. With the terminology of Pak, for any fixed shape $\mu$, there exists a bijection
between multisets of $M$ whose non-zero entries are in $\mu$ and plane partitions whose
non-zero entries are in $\mu$. Algorithm 1 starts from the diagram of the multiset and
then decides on the bounded shape $\mu$, which is chosen to be the bounding rectangle
of the multiset. Then our algorithm exactly applies Pak’s bijection for the diagram
$\mu$, by scanning the corners of $\mu$ from right to left and bottom to top. The two
important properties of Pak’s bijection are the independence of the resulting plane
partition from the shape $\mu$ bounding the diagram of the multiset, and from the
order in which the corners are treated. As a consequence, for a multiset in $M_{a,b}$,
our algorithm yields the same result as Pak’s bijection would give with the fixed
shape $\mu = (a \times b)$. This concludes the proof. \hfill \Box

Proposition 2.2. Algorithm 1 realizes the combinatorial isomorphism

\begin{equation}
\mathcal{P} \simeq \text{MSet}(\mathcal{Z} \times \text{SEQ}(\mathcal{Z})^2).
\end{equation}

Proof. Take the limit $a \to \infty$ and $b \to \infty$ in Proposition 2.1. \hfill \Box

3. Boltzmann sampling

This section recalls basic principles of approximate-size sampling under Boltz-
mann model ([6, 7]).
**Definition 3.1** (Boltzmann model). Let $\mathcal{C}$ be a combinatorial class and $C(x) := \sum_{\gamma \in \mathcal{C}} x^{\frac{1}{|\gamma|}}$ its generating function. Given a coherent positive real value of $x$ (i.e., chosen within the radius of convergence of $C(x)$), the Boltzmann model of parameter $x \in \mathbb{R}^+$ assigns to any element $\gamma \in \mathcal{C}$ the following probability, where $C(x) := \sum_{\gamma \in \mathcal{C}} x^{1/|\gamma|}$ is the generating function of $\mathcal{C}$,

$$P_x(\gamma) = \frac{x^{1/|\gamma|}}{C(x)}.$$ 

A Boltzmann sampler $\Gamma_C(x)$ for $\mathcal{C}$ is an algorithm that produces objects of $\mathcal{C}$ at random under the Boltzmann model. As the probabilities assigned to elements of the same size are equal, the probability induced by a Boltzmann sampler on any given size $n$ is uniform. The size of the output is a random variable $N_x$ satisfying

$$\mathbb{P}(N_x = n) = \frac{C_n x^n}{C(x)}.$$ 

Figure 4 shows this probability distribution for plane partitions. When a target size $n$ has to be achieved, the idea is to tune the parameter $x$ so that $\mathbb{E}(N_x) = n$ (see Section 5).

![Probability distribution of sizes for plane partitions](image)

**Figure 4.** Probability distribution of sizes for plane partitions under Boltzmann model, with different values of the parameter $x$.

Figure 5 briefly summarizes how to obtain samplers corresponding to the constructions used in the specification of $\mathcal{M}$ (see details in [7]); the rules can be combined to build a generator for any class specified with these constructions, in particular the class $\mathcal{M}$. The sampling rules make use of simple auxiliary generators:
\( \Gamma C(x) := (\Gamma A(x), \Gamma B(x)) \)

\( \Gamma C(x) := \text{Geom}(A(x)) \to \Gamma A(x) \)

where \( G \mapsto \gamma Y \) means "return \( G \) independent calls to \( \gamma Y \)".

Define the probability distribution relative to \( A \) and \( x \):

\[
\Pr(K \leq k) = \prod_{j>k} \exp \left( \frac{1}{j} A(x^j) \right).
\]

Let \( \text{MAX_INDEX}(A; x) \) be a generator according to this distribution.

\[
\Gamma C(x) : \gamma \leftarrow \emptyset; k_0 \leftarrow \text{MAX_INDEX}(A; x);
\]

\[
\text{if } k_0 \neq 0 \text{ then}
\]

\[
\text{for } j \text{ from } 1 \text{ to } k_0 - 1 \text{ do}
\]

\[
p \leftarrow \text{Pois} \left( \frac{A(x)}{j} \right);
\]

\[
\text{for } i \text{ from } 1 \text{ to } p \text{ do}
\]

\[
g \leftarrow \gamma, \text{ copy}(j, \Gamma A(x^j))
\]

\[
p \leftarrow \text{Pois} \left( \frac{A(x^{k_0})}{k_0} \right);
\]

\[
\text{for } i \text{ from } 1 \text{ to } p \text{ do}
\]

\[
g \leftarrow \gamma, \text{ copy}(k_0, \Gamma A(x^{k_0}))
\]

\[
\text{return } \gamma.
\]

**Figure 5.** Sampling rules associated to Boltzmann samplers for some combinatorial constructions.

**Proposition 3.2** (Flajolet et al [7]). *Given two combinatorial classes \( A \) and \( B \) endowed with Boltzmann samplers \( \Gamma A(x) \) and \( \Gamma B(x) \), the sampler \( \Gamma C(x) \), as defined in the first entry of Figure 5, is a Boltzmann sampler for \( A \times B \). Given a class \( A \) not containing the empty atom and endowed with a Boltzmann sampler \( \Gamma A(x) \), the samplers \( \Gamma C(x) \), as defined in the second and third entry of Figure 5, are respectively Boltzmann samplers for \( \text{SEQ}(A) \) and for \( \text{MSET}(A) \).*

From these sampling rules, a class \( \mathcal{C} \) recursively specified from atomic sets in terms of these constructions can be endowed with a Boltzmann sampler \( \Gamma C(x) \).

The complexity of generating an object \( \gamma \in \mathcal{C} \) is linearly bounded by the size of \( \gamma \).

**Complexity model.** Let us say a few words on the complexity model that is referred to when stating the linear complexity of Boltzmann sampling. First, notice that the samplers given in Figure 5 draw integers according to distributions (Geom, Pois, \( \text{MAX_INDEX} \)) that involve the exact values of some generating functions. Hence, such generating functions should be evaluated. The theoretical complexity model we adopt, as already specified in [6], relies on the oracle assumption. This assumption makes it possible to separate the combinatorial complexity of the sampler from the complexity of evaluating the generating functions, and thus to delay the detailed complexity study of evaluating generating functions (there are works in progress dedicated to this issue). Given any combinatorial class \( \mathcal{C} \) specified recursively using the constructions of Figure 5, and any given coherent value \( x > 0 \) of \( C(x) \), we assume that an oracle provides, at unit cost, the exact values at \( x \) of the generating functions for all classes intervening in the decomposition of \( \mathcal{C} \).

---

2We hereby correct an omission in the definition of the sampler for \( \text{MSET}(A) \) given in [7], namely the test \( k_0 \neq 0 \).
practice, we work with a fixed precision (e.g., 20 digits) and precompute the values of generating functions used by the Boltzmann sampler.

4. Samplers for plane partitions

4.1. Boltzmann sampler for plane partitions. The explicit bijection between \( \mathcal{M} \) and \( \mathcal{P} \) allows us to design a simple Boltzmann sampler for plane partitions, made of two steps: (i) generate a multiset in \( \mathcal{M} \) under Boltzmann model, (ii) apply Pak’s algorithm to the diagram of the multiset generated.

Lemma 4.1. Algorithm 2 is a Boltzmann sampler \( \Gamma_M(x) \) for the class \( \mathcal{M} \).

Proof. The specification of \( \mathcal{M} \), given in Equation ((5)), is translated to a Boltzmann sampler using the rules of Figure 5. The translation is carried out directly on the diagram of the multiset (recall that the entry \((i, j)\) of the diagram corresponds to the multiplicity of \((Z, i, j)\) in the multiset). \( \square \)

Proposition 4.2. Given \( x < 1 \), the process \( \Gamma_P(x) \) that calls the algorithm \( \Gamma_M(x) \) —as given in Algorithm 2— and applies Pak’s algorithm —as given in Algorithm 1— to the multiset generated, is a Boltzmann sampler for plane partitions.

Proof. As Pak’s algorithm yields a bijection between \( \mathcal{M} \) and \( \mathcal{P} \) preserving the size, it also preserves the Boltzmann distribution. Hence, the Boltzmann sampler for \( \mathcal{M} \) gives rise, via the bijection, to a Boltzmann sampler for \( \mathcal{P} \). \( \square \)

Algorithm 2: \( \Gamma_M(x) \) [Boltzmann sampler for \( \mathcal{M} \)]

\[
\begin{aligned}
M & \text{ is the diagram of the multiset to be generated} \\
\forall (x, y) & \ M[x, y] \leftarrow 0; \\
k_0 & \leftarrow \text{MAX\_INDEX}(A; x); \\
\text{if } k_0 \neq 0 & \text{ then} \\
\quad \text{for } k := 1 \text{ to } k_0 - 1 \text{ do} \\
\quad \quad & \text{p} \leftarrow \text{Pois}(\frac{x^{k}}{1-x^{k}}); \\
\quad \quad \text{for } i := 1 \text{ to } p \text{ do} \\
\quad \quad \quad & x \leftarrow \text{Geom}(x^{k}); \ y \leftarrow \text{Geom}(x^{k}); \\
\quad \quad \quad & M[x, y] \leftarrow M[x, y] + k; \\
\quad \quad & p \leftarrow \text{Pois}(\frac{x^{k}}{x_0(1-x^{k})}); \\
\quad \quad \text{for } i := 1 \text{ to } p \text{ do} \\
\quad \quad \quad & x \leftarrow \text{Geom}(x^{k}); \ y \leftarrow \text{Geom}(x^{k}); \\
\quad \quad \quad & M[x, y] \leftarrow M[x, y] + k_0; \\
\text{return } M; \\
\end{aligned}
\]

Figure 6 shows computation times\(^3\) of \( \Gamma_P(x) \) for sizes up to \( 10^7 \): the first line gives the time of generation of the multiset (\( \Gamma_M(x) \)) and the second line gives the computation time of Pak’s algorithm. The sampler has been implemented in Maple and the bijection in OCaml. As we can see, the complexity is dominated by Pak’s algorithm for objects of large size. This is confirmed by the analysis to be given in Section 5: the complexity of drawing a multiset of size \( n \) is of order \( n^{2/3} \), while the expected complexity of Pak’s algorithm applied to a random multiset of size \( n \) is of order \( n \ln(n)^3 \). Figure 7(a) shows a large plane partition generated by \( \Gamma_P(x) \) for \( x \) close to 1, \( x = 0.947 \).

\(^3\)Computations have been performed on a Mac OS X Power PC G4 1,42GHz, with 1Go of RAM and 512 Ko of cache.
<table>
<thead>
<tr>
<th>approx. size</th>
<th>$10^3$</th>
<th>$10^4$</th>
<th>$10^5$</th>
<th>$10^6$</th>
<th>$10^7$</th>
</tr>
</thead>
<tbody>
<tr>
<td>generation</td>
<td>$\sim 0.1$ sec.</td>
<td>$\sim 0.5$ sec.</td>
<td>$\sim 2$-3 sec.</td>
<td>$\sim 10$ sec.</td>
<td>$\sim 60$ sec.</td>
</tr>
<tr>
<td>Pak’s transform</td>
<td>$\sim 0.1$ sec.</td>
<td>$\sim 0.3$ sec.</td>
<td>$\sim 2$ sec.</td>
<td>$\sim 20$-30 sec.</td>
<td>$\sim 8$-9 min</td>
</tr>
</tbody>
</table>

Figure 6. Time per generation for different sizes of plane partitions.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure6.png}
\caption{(a) A random plane partition of size 15256, generated by $\Gamma P(0.947)$. (b) A random plane partition of size 1005749 generated by $\Gamma P(0.9866)$, seen from the direction $(1,1,1)$.}
\end{figure}

4.2. Boltzmann sampler for $(a \times b)$-boxed plane partitions. According to the equivalence with the definition in terms of diagrams, an element of $\mathcal{M}_{a,b}$ is a multiset of pairs $(i,j)$, with $0 \leq i < a$ and $0 \leq j < b$, each element $(i,j)$ having size $(i+j+1)$. The set of such pairs being finite, Equation (1) yields

\begin{equation}
\mathcal{M}_{a,b} = \prod_{0 \leq i < a} \prod_{0 \leq j < b} \text{SEQ}(Z^{i+j+1})
\end{equation}

Lemma 4.3. Algorithm 3 is a Boltzmann sampler for $\mathcal{M}_{a,b}$.

Proof. Translate the specification (8) to a Boltzmann sampler for $\mathcal{M}_{a,b}$ using the rules of Figure 5. \qed

\begin{algorithm}
\textbf{Algorithm 3: $\Gamma \mathcal{M}_{a,b}(x)$ [Boltzmann sampler for $\mathcal{M}_{a,b}$]}
\begin{algorithmic}
\State $M$ is the diagram of the multiset to be generated
\For {i $\leftarrow 0$ to $a - 1$}
\For {j $\leftarrow 0$ to $b - 1$}
\State $M[i,j] \leftarrow \text{Geom}(x^{i+j+1})$;
\EndFor
\EndFor
\State return $M$;
\end{algorithmic}
\end{algorithm}

4.3. Extension to skew plane partitions. We consider here a natural generalisation of $(a \times b)$-boxed plane partitions, called $(a \times b)$-boxed skew plane partitions. A $(a \times b)$-boxed skew plane partition is given by an index-domain $D \subset [0..a-1] \times [0..b-1]$ satisfying the monotonicity-property \{(i,j) $\in D \Rightarrow (i',j') \in D$ if $i' \in [i,a[$ and $j' \in [j,b[$\}; and by an array $\pi_{i,j}$ of nonnegative integers indexed
by the domain $D$, such that $\pi_{i,j}$ is weakly decreasing for $i$ increasing and for $j$ increasing (see Figure 8). Let us denote by $\mathcal{P}_D$ the class of all such partitions for a given domain $D$. For this new class of partitions, we need to define the hook length of a pair $(i, j)$ in the domain $D$. Let $\ell(i)$ be the minimum abscissa such that $(\ell(i), j) \in D$ and $d(j)$ the minimum ordinate such that $(i, d(j)) \in D$. The hook length of $(i, j)$ in $D$ is then $h(i, j) = (i - \ell(i)) + (j - d(j)) + 1$, which is exactly $i + j + 1$ when $D$ is $[0..a-1] \times [0..b-1]$. In [19], Pak’s bijection is most generally described for skew plane partitions, which leads to the following combinatorial isomorphism:

$$\mathcal{P}_D \simeq \prod_{(i,j) \in D} \mathbb{Z}^{h(i,j)}$$

The Boltzmann sampler for $(a \times b)$-boxed plane partition extends directly to a Boltzmann sampler for skew plane partitions as follows: to sample a diagram, draw the value at each point $(i, j)$ in $D$ according to a geometric law of parameter $x^{h(i,j)}$; then apply Algorithm 1 (Pak’s bijection) to the multiset generated, with the difference that the domain scanned by $(i, j)$ is $D$.

Okounkov and Reshetekin have studied the limit shape of a skew plane partition under Boltzmann distribution, with the Boltzmann parameter $x$ tending to 1 [18]. Typically, they consider an index domain of the form $[0..a-1] \times [0..b-1]$, possibly truncated by smaller rectangles, each truncation by a smaller rectangle introducing an inner corner in the index domain (e.g., the partition of Figure 8 has 3 inner corners). If the lengths of the rectangles are of order $(1 - x)^{-1}$, some interesting phenomena are to be observed on the typical shape of a random skew plane partition. Using a technique based on Schur processes, the authors of [18] provide a precise analysis of these phenomena. They prove that the (rescaled) limit shape of a skew plane partition has a frozen boundary that satisfies explicit equations, and they classify the non-smooth points of the boundary as turning points and cusps. Turning points always appear, even for a boxed domain $(a \times b)$; they correspond to points of tangency of the frozen boundary with the delimiting 3D-box $(x, y, z) \in \mathbb{R}^3_+$, s.t. $(x, y) \in D$. If the index domain has inner corners, some cusp points possibly appear at each of the inner corners.

Our random sampler for skew plane partitions makes it possible to perform simulations and observe these asymptotic phenomena. Figure 9(a) shows a $(100 \times 100)$-boxed plane partition of size 999400. Due to the constraint of lying inside the rectangular box, the shape is quite different from the one of an unconstrained plane partition (Figure 7): in particular a frozen boundary appears that meets the delimiting 3D-box in a tangential way (these points of tangency are precisely the turning points in the terminology of Okounkov and Reshetekin). The difference in the limit shapes —compare Figure 7(b) to Figure 9(a)— is due to the fact that the bounding rectangle of a plane partition under Boltzmann distribution has length larger than $(1 - x)^{-1}$. Precisely, this length parameter is typically of order $(1 - x)^{-1} \log(1 - x)$, as we will see in Section 5.3. We can indeed observe
in Figure 7 that a large random plane partition has 3 “legs”—one in each axis-direction—whose lengths tend to infinity even when the plane partition is rescaled to have unit volume, which essentially corresponds to rescaling by a factor \((1-x)^{-1}\) in each dimension.

Finally, Figure 9(b) shows a skew plane partition of size 1005532 on the index-domain \((100 \times 100)\setminus(50 \times 50)\), which has an inner corner at \((50,50)\). As can be observed, the boundary of the limit shape has a cusp point at the inner corner of the domain, which agrees with the frozen boundary analysis given in [18].

![A (100 x 100)-boxed plane partition of size 999400 drawn under Boltzmann distribution at x = 0.9931.](image1)

![A skew plane partition of size 1005532 on the index-domain \([0..99] \times [0..99] \setminus [0..49] \times [0..49]\), drawn under Boltzmann distribution at x = 0.9942.](image2)

**Figure 9**

4.4. **Samplers targeted around a given size.** An *approximate-size sampler* for a combinatorial class \(\mathcal{A}\) is a procedure that, given \(n \geq 1\) (the target-size) and \(\epsilon > 0\) (the tolerance ratio), samples objects of \(\mathcal{A}\) so that the size of the objects generated is in the range \([n(1-\epsilon), n(1+\epsilon)]\); and the probability distribution of the sampler is uniform on each size \(k \in [n(1-\epsilon), n(1+\epsilon)]\). An *exact-size sampler* for \(\mathcal{A}\) is a procedure that, given \(n \geq 1\), generates objects of \(\mathcal{A}_n\) uniformly at random.

**Theorem 4.4** (Plane Partitions). Define

\[
\xi_n := 1 - (2\zeta(3)/n)^{1/3},
\]

where \(\zeta(s) := \sum_{m \geq 1} m^{-s}\) is Riemann’s zeta function. Then, for \(n \geq 1\) and \(\epsilon > 0\), the process that calls \(\Gamma M(\xi_n)\) until the output size is in \([n(1-\epsilon), n(1+\epsilon)]\) (is exactly \(n\), resp.), and then applies Pak’s algorithm, is an approximate-size sampler (an exact-size sampler, resp.) for plane partitions. Under the oracle assumption, the expected complexity of the approximate-size sampler is \(O(n \ln(n)^3)\) as \(n \to \infty\),
the constant in the big $O$ being independent of $\epsilon$. The expected complexity of the exact-size sampler is $O(n^{4/3})$.

**Theorem 4.5** $((a \times b)$-boxed Plane Partitions). Define

$$\xi_{a,b}^n := 1 - ab/n.$$ 

Then, for $n \geq 1$ and $\epsilon > 0$, the process that calls $\Gamma_M(\xi_{a,b}^n)$ until the output size is in $[n(1-\epsilon), n(1+\epsilon)]$ (is exactly $n$, resp.), and then applies Pak’s algorithm, is an approximate-size sampler (an exact-size sampler, resp.) for $(a \times b)$-boxed plane partitions. There exists a fixed constant $C > 0$ such that, under the oracle assumption, the expected complexity of the approximate-size sampler is bounded by $C ab/\epsilon$, and the expected complexity of the exact-size sampler is bounded by $C abn$.

Let us mention that the targetted samplers for $(a \times b)$-boxed plane partitions are easily extended to the framework of $(a \times b)$-boxed skew plane partitions. For a fixed index-domain $D \subseteq [0..a-1] \times [0..b-1]$, the appropriate value to reach a target size $n$ is $\xi_{D}^n := 1 - |D|/n$, where $|D|$ is the cardinality of $D$ (in particular $|D| = ab$ for $(a \times b)$-boxed plane partitions).

5. Analysis of the complexity

This section is dedicated to proving the expected complexities of the random samplers, as stated in Theorem 4.4 and Theorem 4.5. Recall that the random samplers consist of two steps: generate a diagram using the Boltzmann framework, and apply Pak’s algorithm to the diagram so as to output a random plane partition. Accordingly, the complexity of generation results from adding up the cost of generating a diagram and the cost of Pak’s algorithm.

The expected costs of generating diagrams under Boltzmann model are naturally expressed as harmonic sums, so that the asymptotic analysis is best handled by the Mellin transform, which we recall next. The complexity of Pak’s algorithm is easily shown to be cubic with respect to the maximal value of the hook-length (absciss+ordinate+1) over all nonzero entries. Consequently, in order to study the expected complexity of Pak’s algorithm according to the size, we need to find the asymptotic order of the maximum hook-length under Boltzmann distribution for analysing the approximate-size sampler, and under fixed-size uniform distribution for analysing the exact-size sampler. The analysis under Boltzmann model is done using simple bounds, while we resort to the more elaborated saddle point method in the fixed-size model.

5.1. The Mellin transform. The Mellin transform is a powerful technique to derive asymptotic estimates of expressions involving specific “harmonic” infinite sums, (see [9] for a detailed survey), which occur several times in the analysis of our samplers. Given a continuous function $f(t)$ defined on $\mathbb{R}^+$, the Mellin transform of $f(t)$ is the function

$$f^*(s) := \int_0^\infty f(t)t^{s-1}dt.$$ 

For instance, the Euler Gamma function $\Gamma(s) := \int_0^\infty e^{-t}t^{s-1}dt$ is the Mellin transform of $e^{-t}$. If $f(t) = O(t^{-a})$ as $t \to 0^+$ and $f(t) = O(t^{-b})$ as $t \to +\infty$, then $f^*(s)$ is an analytic function defined on the fundamental domain $a < \text{Re}(s) < b$. In addition, $f^*(s)$ is in most cases continueable to a meromorphic function in the whole complex plane (e.g. $\Gamma(s)$ is continueable to a meromorphic function having its poles at negative integers). In a similar way as the Fourier transform, the Mellin
transform is almost involutive, the function \( f(t) \) being recovered from \( f^*(s) \) using the inversion formula

\[
(10) \quad f(t) = \int_{c-i\infty}^{c+i\infty} f^*(s)t^{-s}ds \quad \text{for any } c \in (a, b).
\]

From the inversion formula and the residue theorem, the asymptotic expansion of \( f(t) \) as \( t \to 0^- \) can be derived from the poles of \( f^*(s) \) on the left of the fundamental domain, the rightmost such pole giving the dominant term of the asymptotic development of \( f(t) \). If \( f^*(s) \) is decreasing very fast as \( \text{Im}(s) \to \infty \) (which occurs in all our examples, based on the fact that \( \Gamma(s) \) has this property and \( \zeta(s) \) is of moderate growth), then the following transfer rules hold,

\[
(11) \quad \text{pole of order } k+1, \quad f^*(s) \sim \lambda_n \frac{(-1)^k k!}{\alpha^{k+1}} (k \geq 0) \Rightarrow \text{term } \ln(t)^k \text{ in dev}_{t \to 0} f(t).
\]

In particular, a simple pole \( \lambda_n/(s-\alpha) \) translates to a term \( \lambda_n/t^\alpha \) in the development of \( f(t) \) as \( t \to 0^+ \).

Another fundamental property of Mellin transforms is to factorize harmonic sums,

\[
(12) \quad g(t) = \sum_{k \geq 1} a_k f(\mu_k t) \Rightarrow g^*(s) = \left( \sum_{k \geq 1} a_k \mu_k^{-s} \right) f^*(s).
\]

5.2. Complexity of Pak’s algorithm. For a multiset \( \mu \in \mathcal{M} := \text{MSet}(\mathbb{Z} \times \text{Seq}(\mathbb{Z}))^2 \) represented by its diagram, we recall that \( w \) and \( h \) stand for the width and height of the bounding rectangle of \( \mu \). Pak’s algorithm scans the double range \([0 \leq i \leq w-1, 0 \leq j \leq h-1]\); when a square \((i, j)\) is treated, the squares that are updated are those on the up-right diagonal \(\{(i+c, j+c) \mid i+c \leq w, j+c \leq h\}\); each update of an entry consists of a fixed number of operations involving \(\{+, -, \text{max}, \text{min}\}\). A simple calculation shows that the sum of the lengths of the up-right diagonals over the squares of the bounding rectangle is

\[
\sum_{i=1}^{m} i(w-i+h-i+1) = \frac{1}{2} Mm^2 + \frac{1}{2} Mm - \frac{1}{6} m^3 + \frac{1}{6} m, \quad \text{where } M = \text{max}(w, h), \quad m = \text{min}(w, h).
\]

In particular, the sum is bounded by \(M^3\), so that the complexity of the algorithm is cubic in \(M\).

Let us introduce a parameter that will crucially simplify the analysis. Given a multiset \( \mu \in \mathcal{M} \) represented as a diagram, the hook length of an entry \((i, j)\) of the diagram is defined as \(h(i, j) := i + j + 1\), i.e., \(h(i, j)\) is the size of \((i, j)\) seen as an element of \(A = \mathbb{Z} \times \text{Seq}(\mathbb{Z})^2\). The maximum hook length of \(\mu\), denoted by \(k(\mu)\), is the maximum value of the hook length over all non-zero entries of the diagram of \(\mu\) (i.e., entries occupied by a non-zero value).

**Lemma 5.1.** Given an element \(\mu\) of the class \(\mathcal{M} := \text{MSet}(\mathbb{Z} \times \text{Seq}(\mathbb{Z}))^2\), the complexity of Pak’s algorithm applied to \(\mu\) is linearly bounded by \(k(\mu)^3\), where \(k(\mu)\) is the maximal hook length of \(\mu\).

**Proof.** The maximal hook length \(k(\mu)\) is at least equal to the width and to the height of the bounding rectangle of \(\mu\), so that the complexity of Pak’s algorithm, which is cubic in \(M\), is also cubic in \(k(\mu)\). \(\square\)

5.3. Complexity of the free Boltzmann sampler for plane partitions. In this section, we analyse the complexity of the free Boltzmann sampler \(\Gamma P(x)\), operating under the sole effect of its parameter \(x\). Given a combinatorial class \(\mathcal{C}\) for which an explicit Boltzmann sampler \(\Gamma \mathcal{C}(x)\) is designed, we write \(\Lambda \mathcal{C}(x)\) for the expected complexity of a call to \(\Gamma \mathcal{C}(x)\). As we are interested in drawing large plane
The factorization property of the Mellin transform, Equation 12, yields

\[ \Lambda P(x) = \Lambda M(x) + E[\text{PakAlgo}](x), \]

where \( \Lambda M(x) \) is the expected complexity of drawing the multiset using the Boltzmann sampler \( \Gamma M(x) \) for \( \mathcal{M} = M\text{SET}(A), \ A = \mathbb{Z} \times \text{SEQ}(\mathbb{Z})^2; \) and \( E[\text{PakAlgo}](x) \) is the expected complexity of Pak’s algorithm under Boltzmann distribution at \( x \).

We first analyse the asymptotic of \( \Lambda M(x) \) as \( x \to 1^- \). By definition of the Boltzmann sampler \( \Gamma M(x) \), for each \( i \geq 1 \), the number of calls to \( \Gamma A(x^i) \) follows a Poisson law \( \text{Pois}(A(x^i)/i) \), where \( A(x) = x/(1 - x)^2 \) is the generating function of \( A \). As a consequence,

\[
\Lambda M(x) = \sum_{i \geq 1} E \left( \text{Pois} \left( \frac{A(x^i)}{i} \right) \right) \Lambda A(x^i) = \sum_{i \geq 1} \frac{A(x^i)}{i} \Lambda A(x^i).
\]

**Lemma 5.2.** The expected complexity of the Boltzmann sampler \( \Gamma M(x) \) for the class \( \mathcal{M} \) satisfies

\[
\Lambda M(x) = \mathcal{O}_{x \to 1^-} \left( \frac{1}{(1 - x)^{\frac{3}{2}}} \right).
\]

**Proof.** A first observation is that, for \( x < 1 \), the cost of a call to \( \Gamma A(x) \) is a constant. Indeed, it consists of two independent calls to \( \Gamma (\text{SEQ}(x)) \), i.e., two independent calls to a geometric law \( \text{Geom}(x) \). A simple calculation shows that the operation of drawing a real number \( u \) uniformly in \( (0, 1) \) and returning \( \lceil \ln(u)/\log(x) \rceil \) produces, at constant cost, a geometric law \( \text{Geom}(x) \).

As the cost of \( \Gamma A(x) \) is a constant uniformly in \( x < 1 \), Equation (13) becomes \( \Lambda M(x) = \mathcal{O} \left( \sum_{i \geq 1} A(x^i)/i \right) \). As a consequence, we have to study the asymptotic of \( F(x) := \sum_{i \geq 1} A(x^i)/i \) as \( x \to 1^- \). This is a first instance where the Mellin transform can be successfully applied (more elementary approaches would apply in this simple case). Define \( L(t) := F(e^{-t}) \). Then

\[
L(t) = \sum_{r \geq 1} \frac{e^{-rt}}{r(1 - e^{-rt})^2} = \sum_{r \geq 1} \frac{1}{r} f(rt), \quad \text{where} \quad f(t) := \frac{e^{-t}}{(1 - e^{-t})^2}.
\]

The factorization property of the Mellin transform, Equation 12, yields

\[
L^*(s) = \left( \sum_{r \geq 1} \frac{1}{r} r^{-s} \right) f^*(s) = \zeta(s + 1) f^*(s),
\]

where \( \zeta(s) := \sum_{r \geq 1} r^{-s} \) is the Riemann zeta function. In addition, \( f(t) = \sum_{n \geq 1} ne^{-nt} \), so that the factorization property yields \( f^*(s) = \left( \sum_{n \geq 1} mn^{-s} \right) \Gamma(s) = \zeta(s-1) \Gamma(s) \).

Thus, \( L^*(s) = \zeta(s + 1) \zeta(s-1) \Gamma(s) \). It is easily checked that \( L(t) = \mathcal{O}(1/t^2) \) as \( t \to 0^+ \) and \( L(t) = \mathcal{O}(e^{-t}) \) as \( t \to \infty \), so that the fundamental domain of \( L^*(s) \) is \( \text{Re}(s) > 2 \). Hence, to determine the asymptotic behavior of \( L(t) \) as \( t \to 0^+ \), we have to find the rightmost poles of \( L^*(s) \) such that \( \text{Re}(s) < 2 \). The function \( \zeta(s) \) has a unique pole at \( s = 1 \) with coefficient 1, and the function \( \Gamma(s) \) has its poles at nonpositive integers. Hence \( L^*(s) \) has a simple pole at \( s = 2 \), with coefficient \( \zeta(3) \), and no other pole for \( \text{Re}(s) \geq 1 \), so that the transfer property of Mellin transform yields

\[
L(t) = \frac{\zeta(3)}{t^2} + \mathcal{O} \left( \frac{1}{t^2} \right) \quad \text{as} \quad t \to 0^+.
\]
The change of variable \( t = -\ln(x) \) yields

\[
F(x) = \frac{\zeta(3)}{(1 - x)^2} + O\left(\frac{1}{1 - x}\right) \quad \text{as } x \to 1^-.
\]

As a consequence, \( \Lambda M(x) = O((1 - x)^{-2}) \) as \( x \to 1^- \).

We now turn to the overall complexity of \( \Gamma P(x) \). As the analysis shows hereafter, the complexity order of the bijective procedure is higher than the complexity of drawing the multiset, due to Pak’s algorithm.

**Lemma 5.3.** The expected complexity \( \Lambda P(x) \) of the Boltzmann sampler \( \Gamma P(x) \) for plane partitions satisfies

\[
\Lambda P(x) = O\left(\frac{1}{(1 - x)^3} \ln\left(\frac{1}{1 - x}\right)^3\right).
\]

**Proof.** The Boltzmann sampler \( \Gamma P(x) \) first calls the sampler \( \Gamma M(x) \) to draw a multiset \( \mu \), and then applies Pak’s bijection to \( \mu \). Hence, the overall complexity of \( \Gamma P(x) \) results from adding up the costs of the two procedures. Lemma 5.2 yields \( \Lambda M(x) = O((1 - x)^{-2}) \) and Lemma 5.1 ensures that the complexity of Pak’s algorithm is linearly bounded by \( k(\mu)^3 \) — with \( k(\mu) \) the maximal hook length of \( \mu \). Hence

\[
\Lambda P(x) = \Lambda M(x) + \mathbb{E}[\text{PakAlgo}](x) = O\left(\frac{1}{(1 - x)^2}\right) + O\left(\sum_{\mu \in \mathcal{M}} k(\mu)^3 \frac{x|\mu|}{M(x)}\right) \quad \text{as } x \to 1^-.
\]

We now study the asymptotic order of the second term. For \( k \geq 1 \), define \( \mathcal{M}_k \) (resp. \( \mathcal{M}_{\leq k} \)) as the subset of \( \mathcal{M} \) whose elements have maximal hook length equal to \( k \) (resp. at most \( k \)), and let \( M_k(x) \) (resp. \( M_{\leq k}(x) \)) be the associated generating function; observe that \( M_k(x) = M_{\leq k}(x) - M_{\leq k-1}(x) \). By definition,

\[
\mathcal{M}_{\leq k} \simeq \prod_{i+j+1 \leq k} \text{SEQ}(Z^{i+j+1}) \simeq \prod_{r \geq 1} \text{SEQ}(Z^r)^r, \quad \text{so } M_{\leq k}(x) = \prod_{r=1}^k \frac{1}{(1 - x^r)^r}.
\]

Define \( R(x) := \sum_{\mu \in \mathcal{M}} k(\mu)^3 \frac{x|\mu|}{M(x)} \). For \( k \geq 1 \), define \( a_k(x) := M_{\leq k}(x) / M(x) \), so that \( a_k(x) \) is the probability that \( \Gamma M(x) \) outputs an object of maximal hook length at most \( k \). Then

\[
R(x) = \sum_{k \geq 1} k^3 \frac{M_k(x)}{M(x)} = \sum_{k \geq 1} k^3((1-a_{k-1}(x)) - (1-a_k(x))) = 1 + \sum_{k \geq 1} ((k+1)^3 - k^3)(1-a_k(x)).
\]

First, we aim at bounding \( 1 - a_k(x) \) for large \( k \). For this purpose, we bound the (larger) quantity \(-\ln(a_k(x))\), as follows. Observe that

\[
a_k(x) = \prod_{r=1}^k \frac{(1-x^{-r})} {\prod_{r \geq 1} (1 - x^{-r})} = \prod_{r \geq k} (1 - x^r),
\]

hence

\[-\ln(a_k(x)) = -\sum_{r > k} r \ln(1 - x^r).\]

We now use the bound \(-\ln(1-u) \leq 2u\) for \( u \leq 1/2 \). Hence, for \( k \) such that \( x^k \leq 1/2 \), i.e., \( k \geq -\ln(2)/\ln(x) \), we have

\[-\ln(a_k(x)) \leq 2 \sum_{r > k} r x^r \leq 2k x^k \frac{x^k}{1-x} + 2 \frac{x^k}{(1-x)^2}.
\]

For \( K \geq 1 \), define the \( K \)th error term of \( R(x) \) as

\[
R_K(x) := \sum_{k > K} ((k+1)^3 - k^3)(1-a_k(x)).
\]
We now aim at finding an integer \( K := K(x) \) such that the \( K \)th error term is small, typically \( R_K(x) \leq 1 \). The inequality \( 1 - u \leq -\ln(u) \) for \( 0 < u \leq 1 \) implies \( 1 - a_k(x) \leq -\ln(a_k(x)) \), so that the bound for \( -\ln(a_k(x)) \) yields, for \( K \geq -\ln(2)/\ln(x) \),

\[
R_K(x) \leq 2 \sum_{k \geq K} ((k+1)^3 - k^3) \left( k \cdot \frac{x^k}{1-x} + \frac{x^k}{(1-x)^2} \right).
\]

As a consequence, for \( K \geq 1/(1-x) \) (which implies \( K \geq -\ln(2)/\ln(x) \), as \( -\ln(2) \leq 1 \) and \( -\ln(x) \geq 1 - x \) for \( 0 < x < 1 \)), we have

\[
R_K(x) \leq \frac{4}{1-x} \sum_{k \geq K} ((k+1)^3 - k^3)x^k \leq \frac{28}{1-x} \sum_{k \geq K} k^3x^k \leq \frac{28}{1-x} \sum_{i > 0} ((2K)^3 + (2i)^3)x^{K+i} \leq \frac{244K^3x^K}{(1-x)^2} + \frac{224xK}{(1-x)^5} \leq \frac{448K^3x^K}{(1-x)^5}.
\]

For a fixed \( x < 1 \), the quantity \( 448K^3x^K/(1-x)^5 \) converges to 0 as \( K \to \infty \). Hence, there exists a minimal integer \( K(x) \geq 1/(1-x) \) such that \( 448K^3x^K/(1-x)^5 \leq 1 \). Moreover, a simple calculation ensures that \( K(x) = O((1-x)^{-1} \ln((1-x)^{-1})) \) as \( x \to 1^- \); indeed, for \( K = [-9(1-x)^{-1} \ln(1-x)] \), \( K^3x^K/(1-x)^5 \to 0 \) as \( x \to 1^- \). Thus,

\[
\sum_{k \geq 1} ((k+1)^3 - k^3)(1 - a_k(x)) \leq \left( \sum_{k=1}^{K(x)} ((k+1)^3 - k^3) \right) + 1 \leq (K(x) + 1)^3,
\]

i.e., \( \sum_{k \geq 1} ((k+1)^3 - k^3)(1 - a_k(x)) \) is \( O(K(x)^3) \), hence is \( O((1-x)^{-3} \ln(1-x)^3) \). Thus, \( \Delta P(x) = O((1-x)^{-3} \ln(1-x)^3) \) as \( x \to 1^- \).

\[\square\]

5.4. **Analysis of the size of a plane partition under Boltzmann model.**

Given \( 0 < x < 1 \), denote by \( N_x \) the random variable giving the size of the output of \( \Gamma P(x) \); Figure 4 shows plots of \( N_x \) for several values of \( x \). As the Boltzmann probability of an object of size \( n \) is \( x^n/P(x) \), the expectation and variance of \( N_x \) satisfy (see [6] for details):

\[
\mathbb{E}(N_x) = \sum_{n \geq 1} n P_n \frac{x^n}{P(x)} = x \frac{P'(x)}{P(x)}, \quad \mathbb{V}(N_x) = \sum_{n \geq 1} n^2 P_n \frac{x^n}{P(x)} - \mathbb{E}(N_x)^2 = x \frac{d\mathbb{E}(N_x)}{dx}.
\]

**Lemma 5.4.** The expectation and variance of the size of a plane partition drawn under Boltzmann model satisfy

\[
\mathbb{E}(N_x) = \frac{2 \zeta(3)}{(1-x)^3} + \mathcal{O}_{x \to 1^-} \left( \frac{1}{(1-x)^2} \right), \quad \mathbb{V}(N_x) = \frac{6 \zeta(3)}{(1-x)^3} + \mathcal{O}_{x \to 1^-} \left( \frac{1}{(1-x)^3} \right).
\]

**Proof.** We use once again Mellin transform to derive the asymptotic estimates. Observe that \( P'(x)/P(x) \) is the logarithmic derivative of \( P(x) \), so that the infinite-product form of \( P(x) \) yields

\[
\mathbb{E}(N_x) = x \sum_{r \geq 1} \frac{r x^{r-1}}{1-x^r} = \sum_{r \geq 1} r^2 \frac{x^r}{1-x^r}.
\]

Define \( L(t) := \mathbb{E}(N_{e^{-t}}) \). Then

\[
L(t) = \sum_{r \geq 1} r^2 \frac{e^{-rt}}{1-e^{-rt}} = \sum_{r \geq 1} r^2 f(rt),
\]

where \( f(t) := e^{-t}/(1-e^{-t}) = \sum_{n \geq 1} e^{-nt} \). Hence

\[
L'(s) = \sum_{r \geq 1} (r^2 e^{-rs}) f^*(s) = \zeta(s-2) f^*(s) = \zeta(s-2) \sum_{n \geq 1} n^{-s} \Gamma(s) = \zeta(s-2) \zeta(s) \Gamma(s).
\]
The function $L^*(s)$ is defined on the fundamental domain $\text{Re}(s) > 3$. The rightmost pole such that $\text{Re}(s) \leq 3$ is at $s = 3$, where $L^*(s) \sim 2\zeta(3)/(s - 3)$. As there are no other poles for $\text{Re}(s) \geq 2$, the transfer rules yield

$$L(t) = \frac{2\zeta(3)}{t^4} + O(t^{-2}) \text{ as } t \to 0^+.$$  

Hence the change of variable $x = -\ln(t)$ gives

$$\mathbb{E}(N_x) = \frac{2\zeta(3)}{(1-x)^3} + O\left(\frac{1}{(1-x)^2}\right) \text{ as } x \to 1^-.$$  

The variance is treated similarly,

$$\mathbb{V}(N_x) = x \frac{d\mathbb{E}(N_x)}{dx} = \sum_{r \geq 1} r^3 \frac{x^r}{(1-x^r)^2}.$$  

Hence the function $L(t) := \mathbb{V}(N_{x^{-1}})$ satisfies $L(t) = \sum_{r \geq 1} r^3 g(rt)$, where $g(t) = e^{-t}/(1 - e^{-t})^2 = \sum_{n \geq 1} ne^{-nt}$. Thus, $L^*(s) = \zeta(s-3)\zeta(s-1)\Gamma(s)$. The location of the poles of $L^*(s)$ and the transfer rules yield

$$L(t) = \frac{6\zeta(3)}{t^4} + O(t^{-2}) \text{ as } t \to 0^+,$$

giving

$$\mathbb{V}(N_x) = \frac{6\zeta(3)}{(1-x)^4} + O\left(\frac{1}{(1-x)^3}\right) \text{ as } x \to 1^-.$$  

□

5.5. Complexity of the targetted samplers. The specificity of Boltzmann samplers is that the parameter of the sampler is not the size of the output but a parameter $x$, whose value influences the distribution of the size of the output. Given a “target size” $n$, the targetted heuristic consists in choosing the Boltzmann parameter $x$ so as to optimize the chances that the output of $\Gamma P(x)$ has size $n$, or close to $n$. A natural choice is to take $x$ solution of $\mathbb{E}(N_x) = n$. As $\mathbb{E}(N_x)$ might be difficult to compute, it proves more convenient to choose an approximate-solution that is easier to compute. This is where the asymptotic estimate of $\mathbb{E}(N_x)$ plays its part.

Lemma 5.5. For $n \geq 1$, let $\xi_n$ be the solution of $2\zeta(3)/(1-x)^3 = n$, i.e.,

$$\xi_n = 1 - (2\zeta(3)/n)^{1/3}.$$  

Define $\pi_n$ as the probability that the output of $\Gamma P(\xi_n)$ has size $n$. For any $\epsilon > 0$, define $\pi_{n,\epsilon}$ as the probability that the size of the output of $\Gamma P(\xi_n)$ is in the range $[n(1-\epsilon), n(1+\epsilon)]$. Then

$$\pi_{n,\epsilon} \to_{n \to \infty} 1, \quad (\pi_n \sim \frac{\epsilon}{n^{2/3}}, \text{ with } c \approx 0.1023).$$  

Proof. As $\xi_n$ is solution of $2\zeta(3)/(1-x)^3 = n$, Lemma 5.3 ensures that $\mathbb{E}(N_{\xi_n}) = n + \mathcal{O}(n^{2/3})$ as $n \to \infty$, i.e., there exists $C > 0$ such that $|\mathbb{E}(N_{\xi_n}) - n| \leq C n^{2/3}$. Hence Chebyshev’s inequality gives, for any $\epsilon > 0$,

$$1 - \pi_{n,\epsilon} = P(|N_{\xi_n} - n| > \epsilon n) \leq P \left( |N_{\xi_n} - \mathbb{E}(N_{\xi_n})| > \epsilon n - C n^{2/3} \right) \leq \frac{\mathbb{V}(N_{\xi_n})}{(\epsilon n - C n^{2/3})^2}.$$  

Given the fact that $\mathbb{V}(N_{\xi_n}) = \mathcal{O}((1 - \xi_n)^{-4}) = \mathcal{O}(n^{4/3})$ as $n \to \infty$, we have $1 - \pi_{n,\epsilon} \to 0$.

The asymptotic estimate of $\pi_n$ is more difficult to establish. From the expression $\pi_n = P_n c_n^n/\rho(\xi_n)$, the asymptotic of $\pi_n$ can easily be derived if we know the
asymptotic of $P_n$ as $n \to \infty$ and of $P(x)$ as $x \to 1^-$. These have first been found by Wright [23]. We sketch here a simpler proof, detailed in [8, ch.VIII.6.], based on the Mellin transform and the saddle-point method. It is necessary for us to recall the main arguments of the proof, as we will need the same ingredients to analyse the maximal hook length under fixed-size uniform distribution in Proposition 5.7. Let us first deal with the asymptotic of $P(x)$, already derived by Wright [23]. The exp-log transform yields

$$P(x) = \exp \left( \sum_{r \geq 1} r \ln \left( \frac{1}{1 - x^r} \right) \right) = \exp \left( \sum_{k \geq 1} \sum_{r \geq 1} \frac{r x^{rk}}{k} \right) = \exp \left( \sum_{k \geq 1} \frac{x^k}{k(1 - x^k)^2} \right).$$

Define $L(t) = \ln(P(e^{-t}))$, so that $L(t) = \sum f(tk)/k$, with $f(t) = e^{-t}/(1 - e^{-t})^2 = \sum_{n \geq 1} ne^{-nt}$. Thus, as already seen in the proof of Lemma 5.2, the Mellin transform of $L(t)$ is $L^*(s) = \zeta(s + 1)\zeta(s - 1)\Gamma(s)$, from which one derives the asymptotic

$$P(z) \sim C_1(1 - z)^{1/12} \exp \left( \zeta(3) \frac{z}{(1 - z)^2} \right) \text{ as } z \to 1^-,$$

with $C_1$ an explicit constant, $C_1 \approx 0.9368$. A crucial observation here is that the asymptotic expansion holds not only for $z \in \mathbb{R}_{\leq 1}$, but also for $z$ complex in a cone centered at 1 and of some fixed angle. Indeed, the Mellin transform inverse formula (10) is also valid for $t$ taking a complex value, provided the integral is convergent; the convergence is easily checked to occur as soon as $\text{Arg}(1 - z) \in (-\pi/3, \pi/3)$, given the fast decay of $\Gamma(s)$ for $s \to i\infty$. Hence, the estimate is valid for $z$ approaching 1 in a sector of the form $\text{Arg}(1 - z) \in (-\pi/3, \pi/3)$.

Then, the asymptotic of $P_n$ is derived by applying the saddle-point method to $P(z)$, seen as a complex analytic function defined for $|z| < 1$. The derivation has first been done by Wright [23] and the method was later generalized by Meinardus, see [1, ch.6] for a survey. The saddle-point method as a general way to get asymptotic estimates is presented in [8, ch.8].

Cauchy’s formula ensures that, for any $0 < x < 1$,

$$P_n = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(xe^{i\theta}) x^{-n} e^{-ins\theta} d\theta.$$

The saddle-point method consists in choosing the integration-circle $\{ xe^{i\theta}, -\pi \leq \theta < \pi \}$ with $x$ minimizing (or close to minimizing) the quantity $P(x)x^{-n}$. A simple calculation ensures that $x$ is precisely the solution of $\mathbb{E}(N_x) = n$; we take the simpler approximate-solution $\xi_n = 1 - (2\zeta(3)/n)^{1/3}$ as radius of the integration-circle. Using the asymptotic estimate $P(z)$ as $z \to 1$ with $\text{Arg}(1 - z) \in (-\pi/3, \pi/3)$, it can be shown that the integral has its dominant part on a small arc $[-\theta_n, \theta_n]$, with $\theta_n$ of order $n^{-5/18}$. In addition, as usual when applying the saddle-point method, the integrand is equivalent (after rescaling) to a gaussian function on the dominating arc; and the integral on the complement arc $[-\pi, -\theta_n] \cup [\theta_n, \pi]$ is exponentially negligible. All calculations done, one obtains

$$P_n \sim \frac{C_2}{n^{2\zeta(3)/36}} \exp \left( \frac{3}{2} (2\zeta(3))^{1/3} n^{2/3} \right),$$

with $C_2$ an explicit constant, $C_2 \approx 0.2315$. $\square$

The probabilities $\pi_{n,\epsilon}$ and $\pi_n$ give the probabilities of success of $\Gamma P(\xi_n)$, targeted respectively as an approximate-size sampler on $[n(1 - \epsilon), n(1 + \epsilon)]$ and as an exact-size sampler for size $n$. The following lemma establishes the precise connection between the expected complexity of a random sampling algorithm and the expected complexity of the same sampler targeted to a certain subset.
Lemma 5.6 (targetting-complexity). Let \( \Gamma C \) be a random sampler on a set \( C \), and let \( \Lambda C \) be the expected complexity of a call to \( \Gamma C \). Given \( A \) a subset of \( C \), let \( \pi_A \) be the probability that \( \Gamma C \) outputs an object in \( A \). Then the rejection algorithm

\[
\text{repeat (}\gamma \leftarrow \Gamma C\text{) until } \gamma \in A \text{ return } \gamma
\]

is a random sampler on \( A \) whose expected complexity \( \Lambda A \) satisfies

\[
\Lambda A = \frac{\Lambda C}{\pi_A}.
\]

Proof. The quantity \( \Lambda A \) satisfies the recursive equation

\[
\Lambda A = \Lambda C + (1 - \pi_A)\Lambda A.
\]

Indeed, a first trial, with expected complexity \( \Lambda C \), is always needed. In case of rejection, with probability \( 1 - \pi_A \), the sampler restarts in the same way as when it is launched. Hence, \( \Lambda A = \Lambda C/\pi_A \). \( \square \)

Proposition 5.7. For any \( \epsilon > 0 \), the expected complexity of the approximate-size sampler for plane partitions is

\[
\mathcal{O}(n \ln(n)^3) \quad \text{as } n \to \infty,
\]

the constant in the big \( \mathcal{O} \) being independent of \( \epsilon \).

The expected complexity of the exact-size sampler for plane partitions is

\[
\mathcal{O}(n^{4/3}) \quad \text{as } n \to \infty.
\]

Proof. By definition of the approximate-size sampler, the sampler \( \Gamma M(\xi_n) \) is launched until the size is in \([n(1 - \epsilon), n(1 + \epsilon)]\), in which case Pak’s algorithm is applied to the multiset. Hence, the expected complexity of the approximate-size sampler is bounded by the expected complexity of the algorithm “repeat \( \gamma \leftarrow \Gamma P(x) \) until \( |\gamma| \in [n(1 - \epsilon), n(1 + \epsilon)]\).” According to Lemma 5.6, the expected complexity of this last sampler is \( \Lambda P(\xi_n)/\pi_{n,\epsilon} \), which is equivalent to \( \Lambda P(\xi_n) = \mathcal{O}(n \ln(n)^3) \), as \( \pi_{n,\epsilon} \to 1 \).

The analysis of the exact-size sampler consists of two steps. Lemma 5.6 ensures that the overall expected complexity of drawing the multisets until the size is \( n \) is \( \Lambda M(\xi_n)/\pi_n \sim n^{4/3} \), the asymptotic being obtained from the estimates of \( \Lambda M(x) \) as \( x \to 1^- \) (obtained in Lemma 5.2) and of \( \pi_n \) as \( n \to \infty \) (obtained in Lemma 5.5). Then, we have to analyse the expected complexity of Pak’s algorithm under the uniform distribution at a fixed size \( n \), denoted by \( E_n[\text{PakAlgo}] \). Lemma 5.1 ensures that \( E_n[\text{PakAlgo}] \) is bounded by the expectation of the cube of the maximal hook length. Hence, with the notations of the proof of Lemma 5.5,

\[
E_n[\text{PakAlgo}] \leq \frac{[z^n](\sum_{k \geq 1} k^3 M_k(z))}{[z^n]M(z)}.
\]

We define, as in Lemma 5.3, \( R(z) = \sum_{k \geq 1} k^3 M_k(z)/M(z) \), so that we have to analyse \([z^n]R(z)M(z)/[z^n]M(z)\). We have already explained in the proof of Lemma 5.3 that the asymptotic of \( P_n = [z^n]P(z) = [z^n]M(z) \) is obtained by analysing the integral

\[
\frac{1}{2\pi} \int_{-\pi}^{\pi} P(\xi_n e^{i\theta}) \xi_n^{-\epsilon} e^{-i\theta} d\theta,
\]

The main arguments of the analysis which we need and recall here are the following. The dominating part of the integral is on a small arc \([-\theta_n, \theta_n]\), where the integrand is equivalent to a positive (gaussian up to rescaling) function; and the integrand is exponentially negligible on the remaining part of the circle. The same integration technique is then successfully applied to get an asymptotic bound for \([z^n]R(z)M(z)\), based on the fact that \( R(z) \) is of moderate growth on the circle of radius \( \xi_n \); precisely
the asymptotic of \( R(x) \) as \( x \to 1^- \) (obtained in Lemma 5.5) is easily extended to an estimate \( R(z) = \mathcal{O}((1-z)^{-3} \ln((1-z)^{-1})^3 \) as \( |z| \to 1 \) (this is due to \( | - \ln(a_k(z)) | \leq | - \ln(a_k(x)) | \) for \( |z| = x \), with the notations of Lemma 5.3). Hence, we have the bound \( R(\xi_n e^{i\theta}) = \mathcal{O}(n \ln(n)^3) \) on the whole circle. The chain of calculations of the saddle point method is then readily carried out (see [8, ch. 8] for details). The main properties that make the method work are the moderate growth of \( R(x) \), and the technical property that the angle delimited by the small arc \([\xi_n e^{-i\theta_n}, \xi_n e^{i\theta_n}]\) seen from 1, converges to 0, i.e., \( \theta_n = o(1-\xi_n) \). We obtain

\[
[z^n]R(z)M(z) \sim R(\xi_n) \int_{-\pi}^{\pi} M(\xi e^{i\theta}) e^{-in\theta} d\theta = R(\xi_n) P_n = \mathcal{O}(n \ln(n)^3) P_n.
\]

Hence \([z^n]R(z)M(z)/[z^n]M(z) = [z^n]R(z)M(z)/P_n = \mathcal{O}(n \ln(n)^3)\), which ensures that

\[
E_n[\text{PakAlgo}] = \mathcal{O}(n \log(n)^3).
\]

Hence, in the fixed-size random sampler for plane partitions, the complexity of Pak’s algorithm is dominated by the complexity \( \mathcal{O}(n^{3/2}) \) to get a random multiset of size \( n \).

\[\square\]

5.6. Complexity of the samplers for \((a \times b)\)-boxed plane partitions. As described in Section 4, the Boltzmann sampler \( \Gamma P_{a,b}(x) \) for \((a \times b)\)-boxed plane partitions, constrained to have length at most \( a \) and width at most \( b \), consists of \( ab \) calls to geometric laws. We have seen in Section 5.3 that any geometric law can be drawn at constant cost \( C \) (in the complexity model of operations on real numbers).

As a consequence, the complexity of \( \Gamma P_{a,b}(x) \) is \( C \cdot a \cdot b \), i.e., the complexity is a constant!

The generating function of \((a \times b)\)-boxed plane partitions \( P_{a,b} \) is

\[
P_{a,b}(z) = \prod_{0 \leq i < a} \frac{x^{i+j+1}}{1-x^i x^{i+j+1}} z_{x=1} \sim \frac{c}{(1-x)^{ab}}, \quad \text{with } c = \prod_{0 \leq i < a} \frac{1}{i+j+1}.
\]

It is readily checked that the expected size \( E(N_x) \) of the output of \( \Gamma P_{a,b}(x) \) satisfies \( E(N_x) \sim ab/(1-x) \) as \( x \to 1^- \). As a consequence, an approximate solution of the targetting equation \( E(N_x) = n \) is \( \xi_n^{ab} = 1 - ab/n \).

**Proposition 5.8.** Given \( n \geq 1 \), let \( \xi_n^{a,b} = 1 - \frac{ab}{n} \). Then the approximate-size sampler for \((a \times b)\)-boxed plane partitions, with a relative tolerance \( \epsilon > 0 \) for the size, has expected complexity bounded by the constant

\[
ab/c_{\epsilon}, \quad \text{with } c_{\epsilon} \text{ a constant of order } \epsilon \text{ for small } \epsilon.
\]

The expected complexity of the exact-size sampler for \((a \times b)\)-boxed plane partitions is linear, the constant of linearity being proportional to \( pq \).

**Proof.** Define \( \pi_n,e \) and \( \pi_n \) as the respective probabilities that the output of \( \Gamma P(\xi_n) \) has size \([n(1-\epsilon), n(1+\epsilon)]\) (resp. has size \( n \)). It is shown in [6] that \( \pi_n,e \sim n \to \epsilon \) \( \epsilon \) for some constant \( c_{\epsilon} \) of order \( \epsilon \) when \( \epsilon \) gets small; and that \( \pi_n \sim c/n \) for some constant \( c > 0 \). As \( \Gamma P_{a,b}(x) \) is of constant complexity \( C \cdot a \cdot b \), Lemma 5.6 implies that the expected complexity of the approximate-size sampler is \( \mathcal{A} P_{a,b}(\xi_n)/\pi_n,e \sim Cpq/c_{\epsilon} \); and the expected complexity of the exact-size sampler is \( \mathcal{A} P_{a,b}(\xi_n)/\pi_n \sim Cabb/c_{\epsilon} \). \[\square\]

Aknowledgement. The authors would like to thank Philippe Flajolet for his help to analyze the algorithm. The article has also greatly benefited from a discussion with Christian Krattenthaler. We finally thank Guénaël Renault for the pretty nice 3D drawings of plane partitions.
REFERENCES


O. B. and C. P.: LIP6 Université Paris 6, É. F.: Algorithms Project, INRIA Rocquencourt

E-mail address: bodini@calfor.lip6.fr, eric.fusy@inria.fr, carine.pivoteau@calfor.lip6.fr