Vol.2013-AL-145 No.28
2013/11/7

Uniformly Random Generation of Floorplans

Katsuhisa Yamanaka[1,a)] Shin-ichi Nakano[2,b)]

Abstract: In this paper we deal with problems on generating mosaic floorplans uniformly at random. We propose an algorithm that generates mosaic floorplans with f faces uniformly at random in polynomial time for each. To the best of our knowledge, this is the first such polynomial-time algorithm. By modifying the algorithm, we give two more algorithms to generate mosaic floorplans with some specified properties uniformly at random.

1. Introduction

It is useful to have means to generate (or sample) objects in a specified class uniformly at random, to have good test data, or good choice, especially in probabilistic algorithms. Several algorithms to generate graphs in a class uniformly at random are known, for trees [2], triangulations [6] and bipartite permutation graphs [7], etc. In this paper we design an algorithm to generate mosaic floorplans with f faces uniformly at random.

A mosaic floorplan is a structure of a partition of a rectangle into rectangles. See some examples in Fig. 1. Mosaic floorplans are one of basic models for VLSI design [4], [5]. The number of mosaic floorplans with f faces is known [8]. A one-to-one correspondence between mosaic floorplans and Baxter permutations is known [1], [3]. Also the number of mosaic floorplans with some properties is known [1].

Our idea for a random generation is as follows. First we define a tree T_f, called the classification tree, in which (1) each leaf in the tree corresponds to a distinct mosaic floorplan, and (2) each vertex v in the tree corresponds to the set of mosaic floorplans corresponding to the leaves in the subtree rooted at v. See Fig. 5. The mosaic floorplans corresponding to a vertex is partitioned into subsets, each of which corresponds to a child of the vertex. Thus each path from the root to a leaf corresponds to a distinct mosaic floorplan with f faces. So if we can choose such paths uniformly at random then we can generate mosaic floorplans uniformly at random. Such paths can be chosen uniformly at random as follows. Assume that we are now at vertex v in T_f and v has children $c_1, c_2, \ldots$. Which child should we choose as the next vertex of the path? We first compute the number, say $M(v)$, of leaves in the subtree rooted at v, and the number, say $M(c_i)$, of leaves in the subtree rooted at c_i for each i. Then with the probability $M(c_i)/M(v)$ we choose each child c_i.

1 Department of Electrical Engineering and Computer Science, Iwate University, Ueda 4-3-5, Morioka, Iwate 020-8551, Japan. Information Processing Society of Japan
2 Department of Computer Science, Gunma University, Tenjin-cho 1-5-1, Kiryu, Gunma 376-8515, Japan.
a) yamanaka@cis.iwate-u.ac.jp
b) nakano@cs.gunma-u.ac.jp

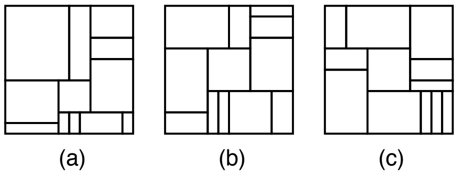

Fig. 1 Examples of mosaic floorplans.

With similar idea, but modified classification trees, we give two more algorithms to generate mosaic floorplans with some specified properties uniformly at random.

The structure of the paper is as follows. Section 2 gives some definitions. Section 3 defines the classification tree. Section 4 gives our first random generation algorithm. Section 5 gives two more random generation algorithms for mosaic floorplans with some specified properties. Section 6 is a conclusion.

2. Definitions

In this section we give some definitions.

A *mosaic floorplan* is a structure of a partition of a rectangle into rectangles, called *faces*. See some examples in Fig. 1. The unbounded face is *the outer face*, and other faces are *inner faces*.

Two mosaic floorplans M_1 and M_2 are *isomorphic* if there exist (1) a one-to-one correspondence between maximal vertical line segments and (2) a one-to-one correspondence between maximal horizontal line segments such that the set of faces located to the left and right of each maximal vertical line segment and the set of faces located to the top and bottom of each maximal horizontal line segment are preserved, respectively. For instance the three mosaic floorplans in Fig. 1 are isomorphic. Intuitively mosaic floorplans are isomorphic if and only if they can be converted to each other by sliding some maximal horizontal and vertical line segments, preserving the sets of faces located to the top, bottom, left and right of each maximal line segment.

We assume no degree four vertex appears in any mosaic floorplan. A vertex with degree three is *w-missing* (west missing) if it has line segments to the top, bottom and right. Similarly we define *e-missing* (east missing), *n-missing* (north missing), *s-missing* (south missing).

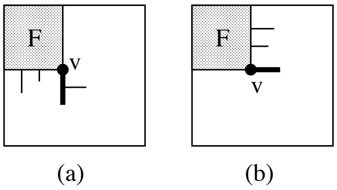

Fig. 2 (a) An upward removable face and (b) a leftward removable face.

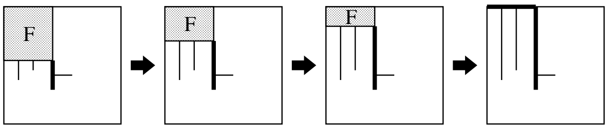

Fig. 3 Removing the first face.

Now we define a floorplan for each set of isomorphic mosaic floorplans as follows. A mosaic floorplan is a *canonical floorplan* if any s-missing vertex appears on the left of any n-missing vertex on any horizontal line segment, and e-missing vertex appears on the top of any w-missing vertex on any vertical line segment. For instance, the mosaic floorplan in Fig. 1(c) is a canonical floorplan.

Let C be a canonical floorplan with $f > 1$ inner faces. The inner face of C having the upper-left corner of the outer face is called *the first* face of C. The first faces are shaded in Figs. 2–4. Let v be the lower-right corner vertex of the first face F of C. If v is e-missing (See Fig. 2(a)), then by continually shrinking the first face F into the uppermost horizontal line of C with preserving the width of F and enlarging the faces below F (See Fig. 3), we can obtain a canonical floorplan with one less faces. So if v is e-missing we say the first face F is *upward removable*. Otherwise, v is s-missing (See Fig. 2(b)), then by continually shrinking the first face F into the leftmost vertical line of C, with preserving the height of F, and enlarging the faces located to the right of F, we can obtain a canonical floorplan with one less faces. So if v is s-missing we say F is *leftward removable*. Thus if $f > 1$ then F is either upward removable or leftward removable. In either case let $P(C)$ be the floorplan derived from C by removing the first face of C as above. Note that $P(C)$ is also a canonical floorplan.

Given a canonical floorplan C, by repeatedly removing the first face of the derived canonical floorplan, we have a sequence $C, P(C), P(P(C)), \ldots$ of canonical floorplans which eventually ends with the canonical floorplan with exactly one inner face. See an example in Fig. 4. We call the sequence the *removing sequence* of C. Note that each canonical floorplan has a unique removing sequence.

Let $RS = (C_f = C, C_{f-1}, \ldots, C_1)$ be the removing sequence of a canonical floorplan C. We define a label $L(C_i)$ for each C_i with $i > 1$ in RS so that $L(C_i)$ explains how the first face of C_i is removed to have C_{i-1}. Let F_i be the first face of C_i. If F_i is upward removable, and the number of faces located to the south of F_i is s, then we define $L(C_i) = (\mathtt{U}, s)$. Otherwise, F_i is leftward removable, and if the number of faces located to the east of F_i is e, then we define $L(C_i) = (\mathtt{L}, e)$. We call $L(C_i)$ the *removing label* of C_i. The *first k labels* of C is the sequence of k removing labels $(L(C_f), L(C_{f-1}), \ldots, L(C_{f-k+1}))$. For example, the first 5 labels of the leftmost canonical floorplan C in Fig. 4 are $((\mathtt{L}, 3), (\mathtt{U}, 2), (\mathtt{L}, 1), (\mathtt{U}, 1), (\mathtt{L}, 2))$. Each canonical floorplan has a unique first $f-1$ labels.

Let f_U and f_L be the number of upward removable faces and the number of leftward removable faces in the removing sequence of C. Let e_v and e_h be the number of maximal vertical line segments and the number of maximal horizontal line segments excluding the contour of the outer face of C. Then $f_U = e_h$ and $f_L = e_v$ hold. Also $e_v + e_h = f - 1$ holds.

3. Classification Tree

In this section we define a tree T_f, called the classification tree, related to the canonical floorplans with f inner faces. In the next section we design our main algorithm based on the tree.

Each leaf in the classification tree corresponds to a distinct canonical floorplan, and each vertex v with depth d in the classification tree corresponds to the set of canonical floorplans (1) corresponding to the leaves in the subtree rooted at v, and (2) sharing the first d labels. Fig. 5 shows the classification tree T_4. We regard the root of the classification tree corresponds to the set of all canonical floorplans with f inner faces, and sharing the first 0 label.

Now we explain how to compute the number of leaves in the subtree rooted at a given vertex.

An inner face F of a floorplan C is *n-touch* if F shares a line segment with the uppermost horizontal line segment of C. Similarly, an inner face F of a floorplan C is *w-touch* if F shares a line segment with the leftmost vertical line of C.

Lemma 3.1 *([1])* Let $C(f, r)$ be the set of canonical floorplans with f inner faces and r maximal vertical line segments not on the outer face, and $C(f, n, w, r)$ the set of canonical floorplans with f inner faces, r maximal vertical line segments not on the outer face, n n-touch faces and w w-touch faces. Then the following equations hold.

$$|C(f, r)| = \frac{\binom{f+1}{r}\binom{f+1}{r+1}\binom{f+1}{r+2}}{\binom{f+1}{1}\binom{f+1}{2}}$$

$$|C(f, n, w, r)| = \binom{f+1}{r+1}\frac{wn}{f(f+1)}\left(\binom{f-n-1}{f-r-2}\binom{f-w-1}{r-1} - \binom{f-n-1}{f-r-1}\binom{f-w-1}{r}\right)$$

Given a vertex v in the classification tree, now we can calculate the number of corresponding canonical floorplans (sharing some first labels), as follows. We start with an example. Let $f = 20$ and v at depth 3 corresponds to the set of canonical floorplans sharing the first 3 labels $((\mathtt{U}, 3), (\mathtt{U}, 2), (\mathtt{L}, 4))$. Then each canonical floorplan shares the same graph structure around the upper-left corner, as shown in Fig. 6, and removing the first 3 inner faces, as in the removing sequence, results in a distinct canonical floorplan with 17 inner faces, including at least three n-touch faces and at least four w-touch faces. Note that if the resulting canonical floorplan has two or less n-touch faces, then the first 3 labels is never $((\mathtt{U}, 3), (\mathtt{U}, 2), (\mathtt{L}, 4))$. Conversely, for each canonical floorplan with 17 inner faces including at least three n-touch faces and at least four w-touch faces, adding three faces in the reverse way in the removing sequence results in a distinct canonical floorplan

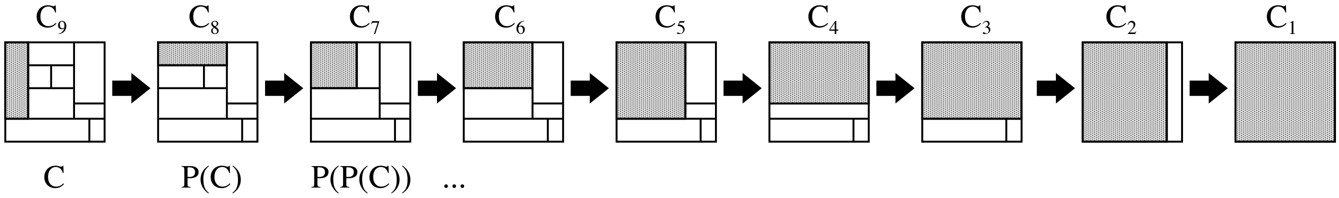

Fig. 4 The removing sequence.

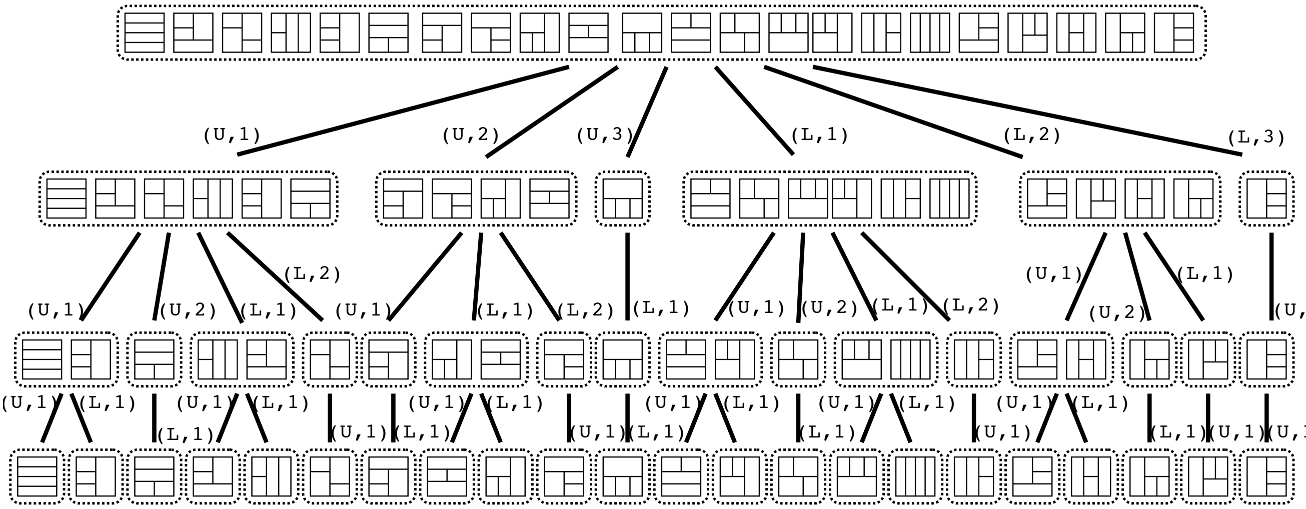

Fig. 5 The classification tree T_4

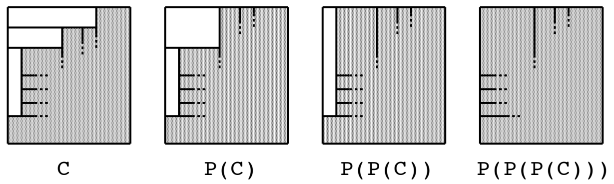

Fig. 6 A canonical floorplan C with the first 3 labels ((U, 3), (U, 2), (L, 4)) and the first 4 floorplans in its removing sequence.

with 20 inner faces sharing the first 3 labels ((U, 3), (U, 2), (L, 4)). We can generalize this example. Let $S(f, L_k)$ be the set of canonical floorplans with f inner faces sharing the first k labels L_k. Let $n(f, L_k)$ be the minimum number of n-touch faces in a canonical floorplans derived from a canonical floorplan in $S(f, L_k)$ by removing the first k inner faces. Similarly $w(f, L_k)$ is defined. Now we have the following equation. Note that if a floorplan has x faces then the maximum number of maximal vertical line segments not on the outer face is $x - 1$.

$$|S(f, L_k)| = \sum_{r=0}^{f-k-1} \sum_{n=n(f,L_k)}^{f-k} \sum_{w=w(f,L_k)}^{f-k} |C(f-k, n, w, r)| \qquad (1)$$

4. Algorithm

In this section we explain our first random generation algorithm for mosaic floorplans. We compute a path from the root to a leaf in the classification tree, without constructing the whole part of the tree. We repeatedly choose the next vertex of the path among the children of the current vertex so that each leaf has an equal chance to be reached. Thus we can generate mosaic floorplans uniformly at random.

Our algorithm is shown below. Algorithm 1 is the main routine. Algorithm 2 randomly chooses the next vertex of the path among the children of the current vertex in the classification tree.

Algorithm 1: Generate-Mosaic(f)

begin
  Find-Child($S(f, \varepsilon)$)

Algorithm 2: Find-Child($S(f, L_k)$)

begin
  $S(f, L_k)$ is the set of canonical floorplans with f inner faces sharing the first k labels L_k
  if $k = f - 1$ **then**
    return $S(f, L_k)$
    /* $S(f, L_k)$ has exactly one canonical floorplan. */
  else
    Let $S(f, L^1_{k+1}), S(f, L^2_{k+1}), \ldots, S(f, L^d_{k+1})$ be a partition of $S(f, L_k)$, where L_k is the common prefix of $L^1_{k+1}, L^2_{k+1}, \ldots, L^d_{k+1}$.
    Generate an integer x in $[1, |S(f, L_k)|]$ uniformly at random.
    Choose the minimum j such that $x \leq \sum_{i=1}^{j} |S(f, L^i_{k+1})|$
    Find-Child($S(f, L^j_{k+1})$)

Assume that we are now at a vertex v in the classification tree, and v corresponds to the canonical floorplans $S(f, L_k)$ sharing the first k labels L_k. Also assume that after removing k inner faces from canonical floorplans in $S(f, L_k)$ the minimum number of n-touch faces is $n(f, L_k)$ and the minimum number of w-touch faces is $w(f, L_k)$. Each child c_i, $i = 1, 2, \ldots, d$, of v in the classification tree corresponds to the set of canonical floorplans $S(f, L^i_{k+1})$ for some L^i_{k+1} sharing the first $k+1$ labels L^i_{k+1}. Since L_k is the common prefix of $L^1_{k+1}, L^2_{k+1}, \ldots, L^d_{k+1}$ then $S(f, L^1_{k+1}), S(f, L^2_{k+1}), \ldots, S(f, L^d_{k+1})$ is a partition of $S(f, L_k)$. Algorithm 2 computes a random value, say x, in $[1, |S(f, L_k)|]$ uniformly at random, chooses the minimum j with $x \leq \sum_{i=1}^{j} |S(f, L^i_{k+1})|$, then recurse with $S(f, L^j_{k+1})$. To compute a path randomly we need to compute $|S(f, L_k)|$ and $|S(f, L^1_{k+1})|, |S(f, L^2_{k+1})|, \ldots, |S(f, L^d_{k+1})|$.

For the root we can compute $|S(f, L_0)| = \sum_{r=0}^{f-1} |C(f, r)|$ by Lemma 3.1. Now $n(f, L_0) = w(f, L_0) = 1$ holds. Assume that we know $|S(f, L_k)|$, $n(f, L_k)$ and $w(f, L_k)$, then we now explain how to compute $|S(f, L^1_{k+1})|, |S(f, L^2_{k+1})|, \ldots, |S(f, L^d_{k+1})|$. All we need to know is $n(f, L^i_{k+1})$ and $w(f, L^i_{k+1})$, since then we can compute $|S(f, L^i_{k+1})|$ by Equation (1). Assume that the $(k+1)$-th label of L^i_{k+1} is (ℓ_1, ℓ_2). We have the following two cases.

Case 1: $\ell_1 = \mathtt{U}$

Let F_{k+1} be the first face of the canonical floorplan C_{f-k} derived from some canonical floorplan in $S(f, L^i_{k+1})$ by removing k inner faces. Since $\ell_1 = \mathtt{U}$, F_{k+1} is upward removable. Hence, the minimum number $n(f, L^i_{k+1})$ of n-touch faces of a canonical floorplan in $S(f, L^i_{k+1})$ is $n(f, L_k) + \ell_2 - 1$. So we have $n(f, L^i_{k+1}) = n(f, L_k) + \ell_2 - 1$. Also if $w(f, L_k) > 1$ then $w(f, L^i_{k+1}) = w(f, L_k) - 1$ holds, otherwise $w(f, L_k) = 1$ then $w(f, L^i_{k+1}) = 1$ holds.

Case 2: $\ell_1 = \mathtt{L}$

Similarly, we have $w(f, L^i_{k+1}) = w(f, L_k) + \ell_2 - 1$. Also if $n(f, L_k) > 1$ then $n(f, L^i_{k+1}) = n(f, L_k) - 1$ holds, otherwise $n(f, L_k) = 1$, then $n(f, L^i_{k+1}) = 1$ holds.

Thus we can compute $n(f, L^i_{k+1})$ and $w(f, L^i_{k+1})$ in constant time. Also note that $\ell_2 < f$ holds. Thus the maximum number of children is at most $2f - 2$.

We have the following theorem.

Theorem 4.1 Our algorithm generates mosaic floorplans uniformly at random in polynomial time for each.

Proof. We can compute $|C(f, n, w, r)|$ in O(1) time by Lemma 3.1 assuming we have a table of $\binom{a}{b}$. Thus we can compute $|S(f, L_0)|$ in $O(f^3)$ time by Equation (1). To choose a child in Algorithm 2, since the number of the children is at most $2f$, we need to compute Equation (1) at most $O(f)$ times and $O(f^4)$ time in total. We repeatedly choose a child $f-1$ times, so we need $O(f^5)$ time for whole algorithm. □ □

5. Random Generation of Mosaic Floorplans with Some Properties

We propose two more algorithms that generates mosaic floorplans with some properties uniformly at random.

The first algorithm generates mosaic floorplans with f inner faces including exactly f_N n-touch faces.

Similar to Section 3, we can define a classification tree $T_f^{f_N}$ related to the canonical floorplans with f inner faces including exactly f_N n-touch faces.

Each leaf in the classification tree corresponds to a distinct canonical floorplan with f inner faces including exactly f_N n-touch faces, and each vertex v with depth d in the classification tree corresponds to the set of canonical floorplans (1) correspond-

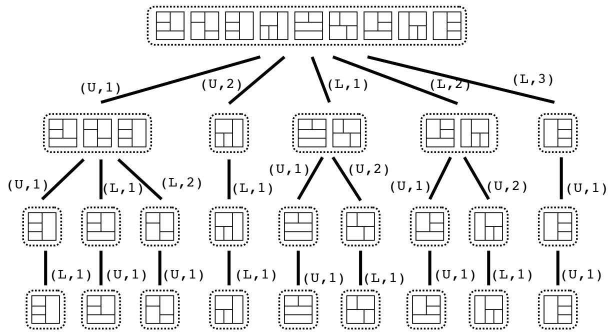

Fig. 7 The classification tree T_4^2.

ing to the leaves in the subtree rooted at v, and (2) sharing the first d labels. For example, T_4^2 is shown in Fig.7.

Let $S(f, f_N, L_k)$ be the set of canonical floorplans with f inner faces including exactly f_N n-touch faces and having the first k labels L_k. Let $n(f, f_N, L_k)$ be the number of n-touch faces in a canonical floorplan derived from a canonical floorplan in $S(f, f_N, L_k)$ by removing the first k inner faces. Note that every such floorplan has exactly $n(f, f_N, L_k)$ n-touch faces. Let $w(f, f_N, L_k)$ be the number of w-touch faces in such floorplans. We can compute $n(f, f_N, L_k)$ and $w(f, f_N, L_k)$ with a similar manner in Section 4 but with a different initialization $n(f, f_N, L_k) = f_N$. Now $|S(f, f_N, L_k)|$ can be calculated by the equation below.

$$|S(f, f_N, L_k)| = \sum_{r=0}^{f-k-1} \sum_{w=w(f, f_N, L_k)}^{f-k} |C(f-k, n(f, f_N, L_k), w, r)| \quad (2)$$

Similar to the algorithm in Section 4, we can compute a path in the classification tree $T_f^{f_N}$ uniformly at random.

We have the following theorem.

Theorem 5.1 Our algorithm generates mosaic floorplans with f inner faces including exactly f_N n-touch faces uniformly at random in polynomial time for each.

The second algorithm generates mosaic floorplans with f inner faces including exactly f_N n-touch faces and exactly f_W w-touch faces.

Similar to Section 3, we can define a classification tree $T_f^{f_N, f_W}$ related to the canonical floorplans with f inner faces including exactly f_N n-touch faces and exactly f_W w-touch faces.

Each leaf in the classification tree corresponds to a distinct canonical floorplan with f inner faces including exactly f_N n-touch faces and exactly f_W w-touch faces, and each vertex v with depth d in the classification tree corresponds to the set of canonical floorplans (1) corresponding to the leaves in the subtree rooted at v, and (2) sharing the first d labels.

Let $S(f, f_N, f_W, L_k)$ be the set of canonical floorplans with f inner faces including exactly f_N n-touch faces and exactly f_W w-touch faces, and having the first k labels L_k. Let $n(f, f_N, f_W, L_k)$ be the number of n-touch faces in a canonical floorplan derived from a canonical floorplan in $S(f, f_N, f_W, L_k)$ by removing the first k inner faces. Similarly, $w(f, f_N, f_W, L_k)$ is defined. We can compute $n(f, f_N, f_W, L_k)$ and $w(f, f_N, f_W, L_k)$ with a similar manner in Section 4 but with different initializations $n(f, f_N, f_W, L_k) = f_N$ and $w(f, f_N, f_W, L_k) = f_W$.

Now we can compute $|S(f, f_N, f_W, L_k)|$ by the follwing equation.

$$
\begin{aligned}
&|S(f, f_N, f_W, L_k)| \\
&\quad = \sum_{r=0}^{f-k} |C(f-k, n(f, f_N, f_W, L_k), w(f, f_N, f_W, L_k), r)| \qquad (3)
\end{aligned}
$$

Similar to the algorithm in Section 4, we can compute a path in the classification tree $T_f^{f_N, f_W}$ uniformly at random.

We have the following theorem.

Theorem 5.2 Our algorithm generates mosaic floorplans with f inner faces including exactly f_N n-touch faces and exactly f_W w-touch faces uniformly at random in polynomial time for each.

6. Conclusions

We have designed an algorithm that generates mosaic floorplans uniformly at random in polynomial time for each. To the best of our knowledge, this is the first polynomial-time algorithm. Also we proposed two more algorithms to generate a mosaic floorplan with some specified properties uniformly at random.

A rectangular drawing is a drawn graph in which every face is a rectangle. The three drawings in Fig. 1(b) and (c) are isomorphic as mosaic floorplans but distinct as rectangular drawings. Can we generate rectangular drawings uniformly at random?

References

[1] E. Ackerman, G. Barequet, and R.Y. Pinter. A bijection between permutations and floorplans, and its applications. *Discrete Applied Mathematics*, 154:1674–1684, 2006.

[2] L. Alonso and R. Schott. *Random Generation of Trees:Random Generators in Computer Science*. Kluwer Academic Publishers, 1995.

[3] S. Dulucq and O. Guibert. Baxter permutations. *Discrete Mathematics*, 180:143–156, 1998.

[4] B.D. He. Optimal binary representation of mosaic floorplans and baxter permutations. In *Proc. The 6th International Frontiers in Algorithmics, and The 8th International Conference on Algorithmic Aspects in Information and Management (FAW-AAIM 2012)*, volume 7285 of *Lecture Notes in Computer Science*, pages 1–12, 2012.

[5] X. Hong, G. Huang, Y. Cai, J. Gu, S. Dong, C.-K. Cheng, and J. Gu. Corner block list: An effective and efficient topological representation of non-slicing floorplan. In *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pages 8–12, 2000.

[6] D. Poulalhon and G. Schaeffer. Optimal coding and sampling of triangulations. In *Proc. the 30th International Colloquium on Automata, Languages and Programming (ICALP 2003)*, volume 2719 of *Lecture Notes in Computer Science*, pages 1080–1094, 2003.

[7] T. Saitoh, Y. Otachi, K. Yamanaka, and R. Uehara. Random generation and enumeration of bipartite permutation graphs. *Journal of Discrete Algorithms*, 10:84–97, 2012.

[8] Z.C. Shen and C.C.N. Chu. Bounds on the number of slicing, mosaic, and general floorplans. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 22(10):1354–1361, 2003.

