

Uniformly Random Generation of Floorplans

Katsuhisa YAMANAKA^{†a)} and Shin-ichi NAKANO^{††}, *Members*

SUMMARY In this paper, we consider the problem of generating uniformly random mosaic floorplans. We propose a polynomial-time algorithm that generates such floorplans with f faces. Two modified algorithms are created to meet additional criteria.

key words: random generation, algorithm, floorplan, mosaic floorplan, classification tree

1. Introduction

It is useful to have a way to uniformly and randomly generate (or sample) objects in a specified class. For instance, uniformly random generators have been proposed for trees [2], triangulations [10], and bipartite permutation graphs [11]. In this paper, we design an algorithm for generating uniformly random mosaic floorplans with f faces.

A mosaic floorplan partitions a rectangle, called the boundary, into smaller rectangles, called faces. Each face includes its boundary. Some examples are shown in Fig. 1. Mosaic floorplans are one of basic models used for very-large-scale integration (VLSI) design [7], [8]. The number of mosaic floorplans with f faces is known [12], and there is a one-to-one correspondence between mosaic floorplans and Baxter permutations [1], [5]. Also, the number of mosaic floorplans with various properties is known [1]: more precisely, there is an exact formula for the number of mosaic floorplans with f faces, r maximal vertical line segments not on the boundary, n n -touch faces and w w -touch faces, where an n -touch face and a w -touch face share a line segment with the uppermost horizontal line segment and the leftmost vertical line segment, respectively.

Our idea for a random generation algorithm 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 each vertex are 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. If we can choose such paths uniformly and randomly, then we can generate uniformly random mosaic floorplans. We can make this uniformly random selection as follows. Assume that we are now at vertex v in T_f , and assume that v has children $c_1, c_2, \dots$. 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 for each i , the number, say $M(c_i)$, of leaves in the subtree rooted at c_i . Then, with probability $M(c_i)/M(v)$, we choose each child c_i .

In this paper, we first propose an algorithm that generates a mosaic floorplan with f faces in $O(f^2)$ time. With a similar idea but with modified classification trees, we present two algorithms that generate mosaic floorplans with additional specified properties.

An algorithm can be designed to generate a uniformly random mosaic floorplan with f faces in linear time; this can be done by generating uniformly random “watermelon structures” [4], which have a bijection into mosaic floorplans [6]. However, by slightly modifying our algorithm, we can also generate mosaic floorplans with various additional properties that are not possible with the linear-time method.

We modify our algorithm to create two additional algorithms that add additional criteria: one generates floorplans that include f_N n -touch faces, and the other generates floorplans that include f_N n -touch faces and f_W w -touch faces. Designating the number of faces that share a line segment with the boundary of a mosaic floorplan arises naturally from VLSI applications. A mosaic floorplan can be used as a model of a circuit, and each face corresponds to a module of the circuit. If a module shares a boundary with the circuit, then it can connect to outside devices. Hence, designating the number of faces sharing line segments with the boundary of a mosaic floorplan corresponds to designating the number of such modules.

The structure of the paper is as follows. Section 2 gives some definitions. Section 3 defines the classification tree. Section 4 presents our first random generation algorithm, and Sect. 5 presents the modified algorithms for floorplans with additional specified properties. Section 6 presents our conclusions. See [13] for a preliminary version of this paper.

2. Definitions

In this section we give some definitions.

A *mosaic floorplan* is a partitioning of a rectangle, called the *boundary*, into smaller rectangles, called *faces*.

Manuscript received March 27, 2015.

Manuscript revised August 2, 2015.

Manuscript publicized December 16, 2015.

[†]The author is with the Department of Electrical Engineering and Computer Science, Iwate University, Morioka-shi, 020-8551 Japan.

^{††}The author is with the Department of Computer Science, Gunma University, Kiryu-shi, 376-8515 Japan.

a) E-mail: yamanaka@cis.iwate-u.ac.jp

DOI: 10.1587/transinf.2015FCP0013

Some examples are shown in Fig. 1.

Two mosaic floorplans, M_1 and M_2 , are *isomorphic* if there exists a one-to-one correspondence between their faces, a one-to-one correspondence between their maximal vertical line segments, and a one-to-one correspondence between their maximal horizontal line segments, such that the set of faces on either side of each vertical line segment and on either side of each horizontal line segment are preserved. For example, the three mosaic floorplans shown in Fig. 1 are isomorphic. Intuitively, mosaic floorplans are isomorphic if and only if they can be converted into each other by sliding some maximal vertical and horizontal line segments, and if doing so preserves the set of faces located on either side of each vertical line segment and on either side of each horizontal line segment.

We assume that no vertices of degree four appear in any mosaic floorplan. A vertex of degree three is *w-missing* (west missing) if it has line segments to the top, bottom, and right; *e-missing* (east missing), *n-missing* (north missing), and *s-missing* (south missing) are defined similarly.

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 to the left of any *n-missing* vertex on any horizontal line segment, and any *e-missing* vertex appears above 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 $f > 1$ faces. The face of C that coincides with the upper-left corner of the boundary is called *the first face* of C . In Figs. 2–4, the first faces are shaded. Let v be the lower-right corner vertex of the first face F of C . If v is *e-missing*, as shown in Fig. 2(a), then by continually shrinking the first face F into the uppermost horizontal line of C while preserving the width of F and enlarging the faces below F , as shown in Fig. 3, we can obtain a canonical floorplan with one fewer face. Thus, if v is *e-missing*, we say that the first face F is *upward removable*. If v is *s-missing*, as shown in Fig. 2(b), then by continually shrinking the first face F into the leftmost verti-

cal line of C , while preserving the height of F and enlarging the faces located to the right of F , we can obtain a canonical floorplan with one fewer face. Thus, if v is *s-missing*, we say that F is *leftward removable*. So, 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 create a sequence $C, P(C), P(P(C)), \dots$ of canonical floorplans that eventually converge to a canonical floorplan with exactly one face. An example of this is shown in Fig. 4. We call this the *removing sequence* of C . Note that each canonical floorplan has a unique removing sequence.

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

Let f_U and f_L , respectively, 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 , respec-

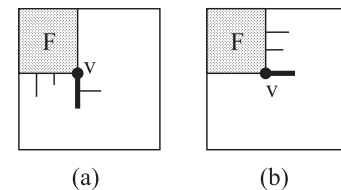


Fig. 2 (a) An upward-removable face, and (b) a leftward-removable face.

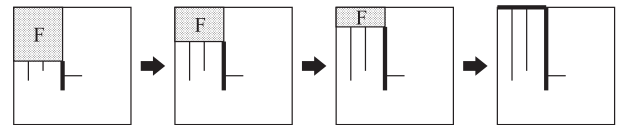


Fig. 3 Removing the first face.

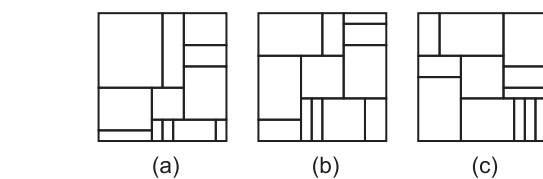


Fig. 1 Examples of mosaic floorplans.

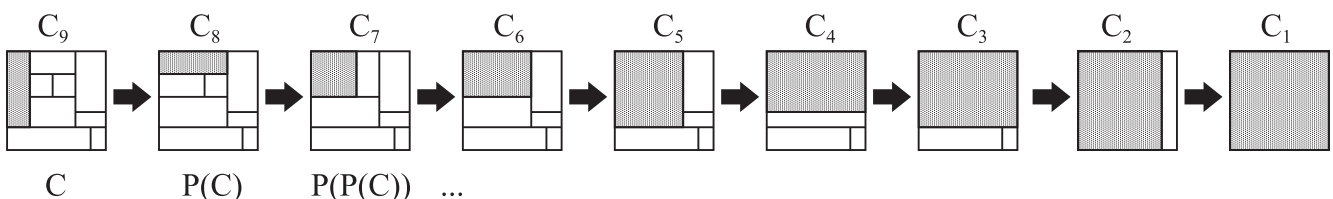
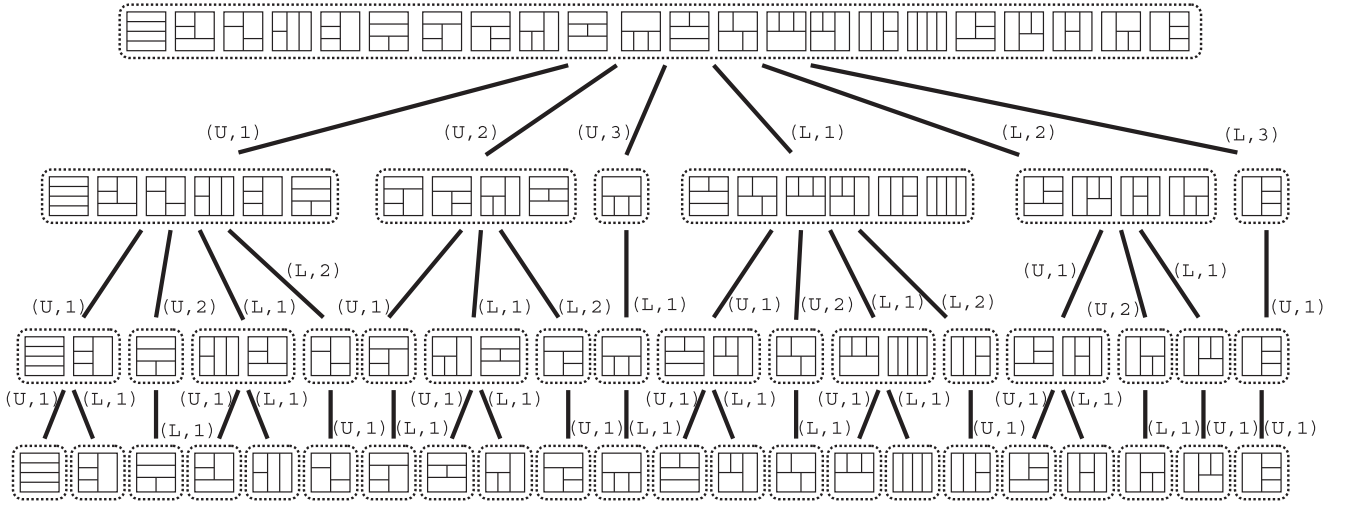


Fig. 4 The removing sequence.

Fig. 5 The classification tree T_4

tively, be the number of maximal vertical line segments and the number of maximal horizontal line segments, excluding the contour of the boundary of C . Then $f_U = e_h$ and $f_L = e_v$ hold, and so does $e_v + e_h = f - 1$.

3. Classification Tree

In this section, we define a tree T_f , called the classification tree, that is related to the canonical floorplans with f faces. In the next section, we will present our main algorithm, which is based on this tree.

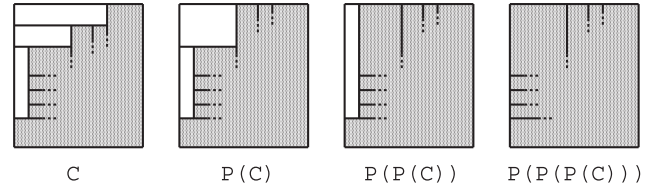
Each leaf in the classification tree corresponds to a distinct canonical floorplan, and each vertex v with depth d in the classification tree corresponds to the set of canonical floorplans that correspond to the leaves in the subtree rooted at v and sharing the first d labels. The classification tree T_4 is shown in Fig. 5. The root of the classification tree corresponds to the set of all canonical floorplans that have f faces and share the first 0 label.

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

A face F of a floorplan C is said to be n -touch if F shares a line segment with the uppermost horizontal line segment of C . Similarly, a face F of a floorplan C is said to be w -touch if F shares a line segment with the leftmost vertical line of C .

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

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

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

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

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 let v at depth three correspond to the set of canonical floorplans sharing the first three labels $((U, 3), (U, 2), (L, 4))$. Then, each canonical floorplan shares the same graph structure around the upper-left corner, as shown in Fig. 6, and removing the first three faces, as in the removing sequence, results in a distinct canonical floorplan with 17 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 fewer n -touch faces, then the first three labels will never be $((U, 3), (U, 2), (L, 4))$. Conversely, for each canonical floorplan with 17 faces that includes at least three n -touch faces and at least four w -touch faces, if three faces are added in the reverse of the removing sequence, then this results in a distinct canonical floorplan with 20 faces sharing the first three 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 faces sharing the first k labels L_k . Let $n(f, L_k)$ be the minimum number of n -touch faces in the canonical floorplans derived by removing the first k faces from a canonical floorplan in $S(f, L_k)$. We define $w(f, L_k)$

similarly. Note that if a floorplan has x faces, then the maximum number of maximal vertical line segments not on the boundary is $x - 1$. We have the following equation.

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

4. Algorithm

In this section, we present the first of our random generation algorithms for mosaic floorplans. We compute a path from the root to a leaf in the classification tree, without constructing the entire tree. We repeatedly choose the next vertex of the path from among the children of the current vertex, so that each leaf has an equal chance of being reached. In this way, we can generate uniformly random mosaic floorplans.

Our algorithm is shown as Algorithm 1. The next vertex of the path is randomly chosen from among the children of the current vertex. $S(f, \varepsilon)$, which corresponds to the root of the classification tree, is the argument for the algorithm; the next random vertex is found recursively.

Algorithm 1: Find-Child($S(f, L_k)$)

```

1 begin
2    $S(f, L_k)$  is the set of canonical floorplans with  $f$  faces
   sharing the first  $k$  labels  $L_k$ 
3   if  $k = f - 1$  then
4     return  $S(f, L_k)$  /*  $S(f, L_k)$  has exactly one
       canonical floorplan. */
5   else
6     Let  $S(f, L_{k+1}^1), S(f, L_{k+1}^2), \dots, S(f, L_{k+1}^d)$  be a partition
       of  $S(f, L_k)$ , where  $L_k$  is the common prefix of
        $L_{k+1}^1, L_{k+1}^2, \dots, L_{k+1}^d$ .
7     Uniformly and randomly generate an integer  $x$  in
        $[1, |S(f, L_k)|]$ .
8     Choose the minimum  $j$  such that  $x \leq \sum_{i=1}^j |S(f, L_{k+1}^i)|$ 
9     Find-Child( $S(f, L_{k+1}^j)$ )
```

Assume that we are now at a vertex v in the classification tree, and assume that v corresponds to the canonical floorplans $S(f, L_k)$ that share the first k labels L_k . Assume also that after removing k faces from the 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 , $i = 1, 2, \dots, d$, of v in the classification tree corresponds to the set of canonical floorplans $S(f, L_{k+1}^i)$ for some L_{k+1}^i that shares the first $k+1$ labels L_{k+1}^i . Since L_k is the common prefix of $L_{k+1}^1, L_{k+1}^2, \dots, L_{k+1}^d$, then $S(f, L_{k+1}^1), S(f, L_{k+1}^2), \dots, S(f, L_{k+1}^d)$ is a partition of $S(f, L_k)$. Algorithm 1 computes a uniformly random value, say x , in $[1, |S(f, L_k)|]$, and it chooses the minimum j with $x \leq \sum_{i=1}^j |S(f, L_{k+1}^i)|$. It then recursively computes $S(f, L_{k+1}^j)$. To randomly compute a path, we need to compute $|S(f, L_k)|$ and $|S(f, L_{k+1}^1)|, |S(f, L_{k+1}^2)|, \dots, |S(f, L_{k+1}^d)|$.

For the root, we can use Lemma 1 to compute $|S(f, L_0)| = \sum_{r=0}^{f-1} |C(f, r)|$. Now, $n(f, L_0) = w(f, L_0) = 1$ holds. If we assume that we know $|S(f, L_k)|$, $n(f, L_k)$, and $w(f, L_k)$, then we compute $|S(f, L_{k+1}^1)|, |S(f, L_{k+1}^2)|, \dots, |S(f, L_{k+1}^d)|$ as shown below. Note that we only need to know $n(f, L_{k+1}^i)$ and $w(f, L_{k+1}^i)$, since if we have them, then we can use Eq. (1) to compute $|S(f, L_{k+1}^i)|$. Assume that the $(k+1)$ -th label of L_{k+1}^i is (ℓ_1, ℓ_2) . We have the following two cases.

Case 1: $\ell_1 = U$

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

Case 2: $\ell_1 = L$

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

Thus, we can compute $n(f, L_{k+1}^i)$ and $w(f, L_{k+1}^i)$ 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 1: After preprocessing in polynomial time, our algorithm generates each uniformly random mosaic floorplan in $O(f^2)$ time.

Proof. For the preprocessing phase, we compute all possible values of $|S(f, L_k)|$. Specifically, we compute the right-hand side of Eq. (1) for all $f'(1 \leq f' \leq f)$, $k'(1 \leq k' \leq f')$, $n'(f, L_k)(1 \leq n'(f, L_k) \leq f - k)$, $w'(f, L_k)(1 \leq w'(f, L_k) \leq f - k)$, and $r'(1 \leq r' \leq f)$. This computation can be done in polynomial time. By using this table, we can obtain $|S(f, L_k)|$ in $O(1)$ time. To choose a child in Algorithm 1, since the number of the children is at most $2f$, we need to look up a value in the table at most $O(f)$ times. We repeatedly choose a child $f - 1$ times, so we need $O(f^2)$ time for the entire algorithm. Q.E.D.

5. Random Generation of Mosaic Floorplans with Particular Properties

We propose two algorithms for generating uniformly random mosaic floorplans with various particular properties.

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

Similar to what we did in Sect. 3, we can define a classification tree $T_f^{f_N}$ that is related to the canonical floorplans for this set.

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 that correspond to the leaves in the subtree rooted at v and that share 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 total faces, including exactly f_N n-touch faces and with 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 by removing the first k faces from a canonical floorplan in $S(f, f_N, L_k)$. 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 $w(f, f_N, L_k)$ in a manner similar to what we did in Sect. 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 using the following equation.

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

Similar to what was done in the algorithm in Sect. 4, we can compute a uniformly random path in the classification tree $T_f^{f_N}$.

We have the following theorem.

Theorem 2: After a preprocessing phase that can be completed in polynomial time, our algorithm generates a uniformly random mosaic floorplan with f total faces, including exactly f_N n-touch faces, in $O(f^2)$ time.

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

Similar to what we did in Sect. 3, we can define a classification tree $T_f^{f_N, f_W}$ that is related to this set of canonical floorplans.

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 that correspond to the leaves in the subtree rooted at v and that share the first d labels.

Let $S(f, f_N, f_W, L_k)$ be the set of canonical floorplans with f total faces, including exactly f_N n-touch faces and exactly f_W w-touch faces, and with the first k labels L_k .

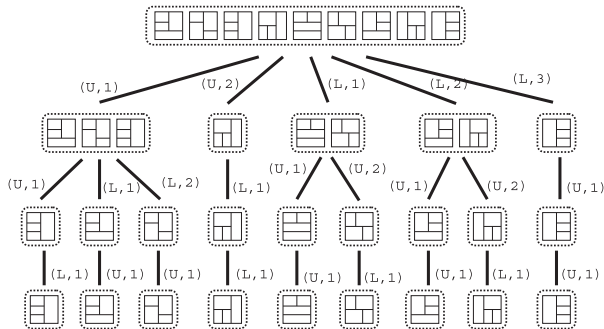


Fig. 7 The classification tree T_4^2 .

Let $n(f, f_N, f_W, L_k)$ be the number of n-touch faces in a canonical floorplan derived by removing the first k faces from a canonical floorplan in $S(f, f_N, f_W, L_k)$. We define $w(f, f_N, f_W, L_k)$ similarly. We can compute $n(f, f_N, f_W, L_k)$ and $w(f, f_N, f_W, L_k)$ in a manner similar to what we did in Sect. 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 following equation.

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

By using a method similar to the algorithm in Sect. 4, we can compute a uniformly random path in the classification tree $T_f^{f_N, f_W}$.

We have the following theorem.

Theorem 3: After a preprocessing phase that can be completed in polynomial time, our algorithm generates a mosaic floorplan with f total faces, including exactly f_N n-touch faces and exactly f_W w-touch faces, in $O(f^2)$ time.

6. Conclusions

We have designed an algorithm that generates a uniformly random mosaic floorplan in polynomial time. We propose two additional algorithms that generate a uniformly random mosaic floorplans with additional specified properties.

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

Our random generation algorithms are related to enumeration algorithms. For example, if we traverse the classification tree in a depth-first manner instead of choosing uniformly random children, we obtain an enumeration algorithm for mosaic floorplans. Based on a similar idea, we have designed enumeration algorithms for floorplans [9]. Such enumeration algorithms based on tree traversing, called reverse search [3], can be modified to obtain random generation algorithms, if the number of descendants of each vertex of the tree can be calculated.

References

- [1] E. Ackerman, G. Barequet, and R.Y. Pinter, "A bijection between permutations and floorplans, and its applications," *Discrete Applied Mathematics*, vol.154, pp.1674–1684, 2006.
- [2] L. Alonso and R. Schott, *Random Generation of Trees: Random Generators in Computer Science*, Kluwer Academic Publishers, 1995.
- [3] D. Avis and K. Fukuda, "Reverse search for enumeration," *Discrete Applied Mathematics*, vol.65, no.1-3, pp.21–46, 1996.

- [4] N. Bonichon and M. Mosbah, "Watermelon uniform random generation with applications," *Theoretical Computer Science*, vol.307, pp.241–256, 2003.
- [5] S. Dulucq and O. Guibert, *Baxter permutations*, *Discrete Mathematics*, vol.180, pp.143–156, 1998.
- [6] S. Felsner, É. Fusy, M. Noy, and D. Orden, *Bijections for baxter families and related objects*, *J. Combinatorial Theory, Series A*, vol.118, no.3, pp.993–1020, 2011.
- [7] B.D. He, "Optimal binary representation of mosaic floorplans and baxter permutations," *Proc. 6th International Frontiers in Algorithms, and The 8th International Conference on Algorithmic Aspects in Information and Management (FAW-AAIM 2012)*, vol.7285 of *Lecture Notes in Computer Science*, pp.1–12, 2012.
- [8] 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," *Proc. IEEE/ACM International Conference on Computer-Aided Design*, pp.8–12, 2000.
- [9] S. Nakano, "Enumerating floorplans with n rooms," *Proc. 12th International Symposium on Algorithms and Computation, (ISAAC 2001)*, volume 2223 of *Lecture Notes in Computer Science*, pp.107–115, 2001.
- [10] D. Poulalhon and G. Schaeffer, "Optimal coding and sampling of triangulations," *Proc. 30th International Colloquium on Automata, Languages and Programming (ICALP 2003)*, volume 2719 of *Lecture Notes in Computer Science*, pp.1080–1094, 2003.
- [11] T. Saitoh, Y. Otachi, K. Yamanaka, and R. Uehara, "Random generation and enumeration of bipartite permutation graphs," *J. Discrete Algorithms*, vol.10, pp.84–97, 2012.
- [12] Z.C. Shen and C.C.N. Chu, "Bounds on the number of slicing, mosaic, and general floorplans," *IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems*, vol.22, no.10, pp.1354–1361, 2003.
- [13] K. Yamanaka and S. Nakano, "Uniformly random generation of floorplans (extended abstract)," *Proc. 9th Hungarian-Japanese Symposium on Discrete Mathematics and Its Applications*, pp.188–195, 2015.



Katsuhisa Yamanaka is an assistant professor of Department of Electrical Engineering and Computer Science, Faculty of Engineering, Iwate University. He received B.E., M.E. and Dr. Eng. degrees from Gunma University in 2003, 2005 and 2007, respectively. His research interests include combinatorial algorithms and graph algorithms.



Shin-ichi Nakano received his B.E. and M.E. degrees from Tohoku University, Sendai, Japan, in 1985 and 1987, respectively. In 1987 he joined Seiko Epson Corp., and in 1990 he joined Tohoku University. In 1992, he received his Dr. Eng. degree from Tohoku University. Since 1999 he has been a faculty member of Department of Computer Science, Faculty of Engineering, Gunma University. His research interests are graph algorithms and graph theory. He is a member of IPSJ, JSIAM, ACM, and IEEE