数理解析研究所講究録
第1644巻 2009年 135-149

Polygons Folding to Plural Incongruent Orthogonal Boxes

Jun Mitani*†and Ryuhei Uehara‡

Abstract

We investigate the problem of finding orthogonal polygons that fold to plural incongruent orthogonal boxes. There are two known polygons that fold to produce two incongruent orthogonal boxes. In this paper, we show that there are infinite such polygons. We also show that there exists a tile that produces two incongruent orthogonal boxes.

1 Introduction

Polygons that can fold to a convex polyhedron have been investigated since Lubiw and O'Rourke posed the problem in 1996 [5]. Recently, Demaine and O'Rourke published a book about geometric folding algorithms that includes many results about such polygons [3, Chapter 25]. One of the many interesting problems in this area is that whether there exists a polygon that folds to plural incongruent orthogonal boxes. Biedl et al. answered "yes" by finding two polygons that fold to two incongruent orthogonal boxes [2] (see also [3, Figure 25.53]). However, are these two polygons exceptional? We show that the answer is "no."

In this paper, we first report that there are more than twenty five thousands such polygons of several sizes. These polygons are obtained by two randomized algorithms. The first algorithm repeatedly produces many nets of orthogonal boxes at random, and matches them in a huge hash table. If two or more nets of distinct boxes coincide, we have a desired polygon. The second algorithm first holds two or more distinct orthogonal boxes with the same surface area, and it randomly cuts them open into a same net simultaneously. If the process succeeds to the last step, we have a desired polygon. The correctness of the algorithms are based on some nontrivial properties of these nets of convex orthogonal boxes.

Some of those polygons can be extended to general size. Using this fact, we also show that there exist an infinite number of polygons that can fold to two orthogonal boxes. Moreover, we show that there exists a simple polygon that can fold to two orthogonal

*Department of Computer Science, University of Tsukuba, Tennodai 1-1-1, Tsukuba, Ibaraki 305-8573, Japan. mitani@cs.tsukuba.ac.jp
†PRESTO, JST.
‡School of Information Science, JAIST, Asahidai 1-1, Nomi, Ishikawa 923-1292, Japan. uehara@jaist.ac.jp

$$
\begin{aligned}
P(11) &= \{(1,1,5),(1,2,3)\} \\
P(15) &= \{(1,1,7),(1,3,3)\} \\
P(17) &= \{(1,1,8),(1,2,5)\} \\
P(19) &= \{(1,1,9),(1,3,4)\} \\
P(23) &= \{(1,1,11),(1,2,7),(1,3,5)\} \\
P(27) &= \{(1,1,13),(1,3,6),(3,3,3)\} \\
P(29) &= \{(1,1,14),(1,2,9),(1,4,5)\} \\
P(31) &= \{(1,1,15),(1,3,7),(2,3,5)\} \\
P(32) &= \{(1,2,10),(2,2,7),(2,4,4)\} \\
P(35) &= \{(1,1,17),(1,2,11),(1,3,8),(1,5,5)\} \\
P(44) &= \{(1,2,14),(1,4,8),(2,2,10),(2,4,6)\} \\
P(45) &= \{(1,1,22),(2,5,5),(3,3,6)\} \\
P(47) &= \{(1,1,23),(1,2,15),(1,3,11),(1,5,7),(3,4,5)\} \\
P(56) &= \{(1,2,18),(2,2,13),(2,3,10),(2,4,8),(4,4,5)\} \\
P(59) &= \{(1,1,29),(1,2,19),(1,3,14),(1,4,11),(1,5,9),(2,5,7)\} \\
P(68) &= \{(1,2,22),(2,2,16),(2,4,10),(2,6,7),(3,4,8)\} \\
P(75) &= \{(1,1,37),(1,3,18),(3,3,11),(3,4,9),(5,5,5)\}
\end{aligned}
$$

Table 1: Sets $P(S)$ for small S.

boxes, and that tiles the plane. This pattern may be used to produce two kinds of boxes of two different volumes on demand without loss of material.

We also consider some general cases of the problem. If we admit to congruent orthogonal boxes, we have a smallest net that can fold to an orthogonal box of size $1 \times 1 \times 2$ in three different ways. When we admit to general polyhedra, which may be concave, a puzzle called "Cubigami puzzle" gives us one of the smallest nets that can fold to seven (i.e., all possible) orthogonal polyhedra of size 14. We also show that there is a simple net that folds to at least k orthogonal distinct polyhedra for any positive integer k. That is, if we admit concave polyhedra, we have nets that fold to arbitrary many orthogonal polyhedra.

2 Preliminaries

In this paper, we concentrate on orthogonal polygons that consist of unit squares. For a positive integer S, we denote by $P(S)$ the set of three integers a, b, c with $0 < a \leq b \leq c$ and $ab + bc + ca = S$, i.e., $P(S) = \{(a,b,c) \mid ab + bc + ca = S\}$. Clearly, it is necessary to satisfy $|P(S)| \geq k$ to have a polygon of size $2S$ that can fold to k incongruent orthogonal boxes. For example, the two known polygons in [2] correspond to $P(11) = \{(1,1,5), (1,2,3)\}$ and $P(17) = \{(1,1,8), (1,2,5)\}$. Some examples are shown in Table 1.

Let B be an orthogonal box of size $a \times b \times c$. Then there are six faces that consist of two rectangles of size $a \times b$, $b \times c$, and $c \times a$, respectively. We regard each rectangle as a set of unit squares. That is, B consists of $2(ab + bc + ca)$ unit squares. Then, for B, we define a dual graph $G(B) = (V, E)$ of B as follows; V is the set of $2(ab + bc + ca)$ unit squares, and E contains an edge $\{u, v\}$ iff two unit squares u and v share an edge on B, or they are incident on B. It is easy to see that $G(B)$ is a 4-regular graph of $2(ab + bc + ca)$

vertices, and hence $|E| = 4(ab + bc + ca)$. Then we have the following observation:

Observation 1 *Let T be a spanning tree of $G(B)$ for some B. For every edge $\{u, v\}$ not in T, we cut the edge shared by two unit squares u and v on B. Then, we obtain a net P of B.*

That is, we can make a net P of B for any orthogonal box B. In the case, we say that the spanning tree T *produces* P. However, spanning trees themselves are not good to represent nets of a box. Suppose that a polygon P can fold to an orthogonal box B. In general, P contains a rectangle R of size $x \times y$ with $x > 1$ and $y > 1$. Then, no spanning tree T generates P since T forces unnecessary cuts of inside of R. The following lemma patches this problem.

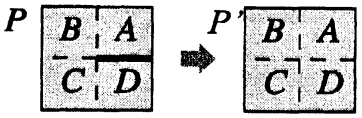

Figure 1: Gluing.

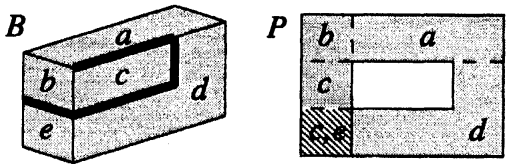

Figure 2: A half of a nonsimple polygon folding to a box.

Lemma 1 *Let P be a polygon that can fold to a box B. If P has a cut between two unit squares A and D in Figure 1, we glue them and obtain P'. Then P' also can fold to B.*

Proof. Since B is a convex orthogonal box, it follows. □

Repeating the gluing in Lemma 1, we obtain a polygon P that has no two consecutive identical edges, which means P contains no unnecessary cuts. From the viewpoint of programming, it is sufficient to represent each polygon P by a usual 0/1 matrix in a natural way, and ignore such cuts. Hence, hereafter, we assume that polygons have no two consecutive identical edges.

Let P be a polygon that can fold to a box B (or convex orthogonal polyhedron). We say that P *overlaps* if P contains two unit squares s_1 and s_2 such that s_1 and s_2 overlap when we lay out P. We also say that P *touches* if P contains two unit squares s_1 and s_2 such that though s_1 and s_2 do not share an edge on B, so are they when we lay out P. P is said to be *simple* if P neither overlaps nor touches. One may think that any polygon that can fold to a box is simple. However, it is not the case.

Lemma 2 *Let B be an orthogonal box and P a polygon that can fold to B. Then, P is not necessarily simple.*

Proof. For B of size $1 \times 2 \times 3$, we make a (half of) polygon P as in Figure 2. Then, clearly, P is a polygon that can fold to B, but P overlaps. If we cut between d and e, instead of b and e, P touches. □

To characterize simple polygon P that can fold to B, we introduce some notations. We imagine that a polygon P is represented by a usual 0/1 matrix in a natural way. For a polygon P, the *silhouette* P' of P is given by the 0/1 matrix after laying out of P. Then, a *boundary edge* of P' is an edge between 0-pixel and 1-pixel. Given any boundary edge e of P', it is easy to follow the whole boundary containing e. In general, a polygon P' has a single *external boundary* and any number of *internal boundaries* delimiting internal *holes*. In this paper, we assume 8-connectivity. That is, the fourth polygon in Figure 4 contains no hole, and the silhouette of P in Figure 2 contains one hole.

Theorem 3 *Let B be an orthogonal box and P a polygon that can fold to B. By Lemma 1, we assume that P contains no unnecessary cuts. We lay out P on the plane, and let P' be a silhouette of P. Then P is simple if and only if P' does not contain a hole.*

Proof. We first assume that P touches and show that P' has a hole. To derive a contradiction, we suppose P' does not have a hole. Let e_1 be an external boundary edge of P'. Then P has a boundary $e_1, e_2, \ldots, e_k$ such that they appear in this order. Since P touches, there are two edges e_i and e_j such that they are placed at the same place when we lay out P, and hence they do not appear on the external boundary of P'. It is easy to see that there are no four indices $i < i' < j < j'$ such that e_i and e_j are placed at the same place, $e_{i'}$ and $e_{j'}$ are placed at the same place, and these places are distinct. Hence, without loss of generality, we can assume that $i < j$ and they are minimal; that is, there is no edge $e_{i'}$ with $i < i' < j$ such that P touches at $e_{i'}$. Then we have $i + 1 < j$ by Lemma 1. Moreover, we can assume that there is no external boundary edge between e_i and e_j. (That is, external boundary edges are in $e_1, \ldots, e_{i-1}$ and $e_{j+1}, \ldots, e_k$.) Now we consider two edges e_{i+1} and e_{j-1} (Figure 3). Since e_i and e_j are minimal edges with

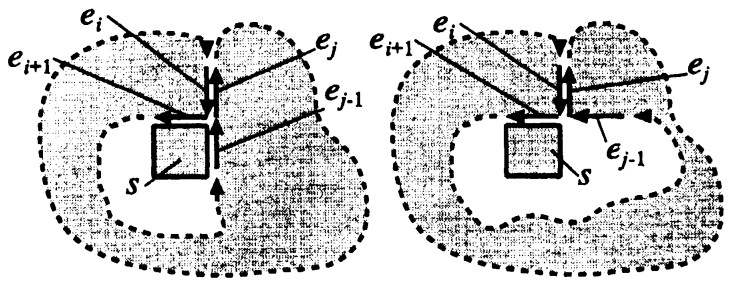

Figure 3: Touching edges e_i and e_j.

respect to touch, e_{i+1} and e_{j-1} are placed at the different places. Since they are edges of unit squares, we have three possible cases. In Figure 3, the first two cases are illustrated. The third case is symmetric to the first one, and hence omitted. Now we assume that P' has no hole. Hence, in any case, we have a unit square s incident to e_{i+1} but not incident to e_i as depicted in Figure 3. Let e be the edge of s placed on the same place of e_{i+1}. By the minimality of e_i and e_j, e is not between e_i and e_j. Hence $e < e_i$ or $e > e_j$. However, this contradicts that $e_1, e_2, \ldots, e_k$ be edges appearing on the boundary of P in this order; if $e < e_i$ or $e > e_j$, the boundary of P is twisted and hence P is disconnected. Thus there is no square s at this place, and hence P' has a hole.

When P overlaps, we remove duplicate regions that occur when we lay out P, and reduce P to P'' that does not overlap but touches. Then we can use the same argument again for P'' and have the theorem. □

3 Randomized Algorithms

In this section, we give two randomized algorithms and their experimental results.

3.1 Random unfolding based on spanning tree

The first algorithm randomly generates the nets of boxes of a same size. It maintains a huge hush table and matches the nets. Precisely, it is described in Algorithm 1.

Input : S with $|P(S)| > 1$;
Output: Polygons of size $2S$ that fold to plural boxes;
1 clear a hash table H;
2 **while** *true* **do**
3 | choose a type $t = (a, b, c)$ in $P(S)$ at random;
4 | generate a spanning tree T of $G(B)$ for an orthogonal box B of size $a \times b \times c$ at random;
5 | represent a polygon P corresponding to T by a 0/1 matrix;
6 | **if** (t', P) *is in* H *with* $t \neq t'$ **then** output P (and all associate types);
7 | **if** P *is not in* H **then** add (t, P) into H;
8 **end**

Algorithm 1: Outline of random generation based on spanning tree

We aim at finding polygons shared by two or more types. Hence, the algorithm ignores weak points mentioned in Preliminaries. More precisely, the algorithm has the following flaws; (1) it does not generate the polygons *uniformly* at random, and (2) some polygons may not be simple.

Fortunately, the flaws cause few errors through our experiments; in fact, among 2165 outputs, the algorithm produced 2139 simple polygons, which are solutions, and only 26 non-simple polygons, which are not solutions. We note that from the algorithmic point of view, it is easy to avoid the flaw (2) in linear time when the algorithm outputs each solution. By Theorem 3, we can avoid the flaw (2) by finding a hole of the silhouette of P, and that can be done in linear time (for example, Asano and Tanaka propose a simple linear time algorithm with constant working space [1]).

3.1.1 Experimental results

We first ran the algorithm on a laptop (IBM ThinkPad X40: 1 Processor with 1.5GB Memory). This generated approximately 3×10^6 polygons in 1 hour, and obtained around 100 solutions for $P(11)$. To experiment more efficiently, we used a supercomputer (SGI Altix 4700: 96 Processors with 2305GB Memory). We used an implement of the Mersenne

Table 2: Experimental results (1)

$2S(S)$	$\lvert P(S)\rvert$	~RG($\times 10^7$)	Sols	Errs
22(11)	2	6.7	541	15
30(15)	2	18.6	72	1
34(17)	2	28.4	708	0
38(19)	2	30.4	41	0
46(23)	3	191.0	660	8
54(27)	3	126.7	3	0
58(29)	3	89.3	37	0
62(31)	3	82.4	5	0
64(32)	3	204.8	56	2
70(35)	4	91.3	14	0
88(44)	4	217.0	2	0
90(45)	3	34.6	0	0
94(47)	5	51.3	0	0
112(56)	5	36.0	0	0
118(59)	6	35.5	0	0
Total	-	-	2139	26

Table 3: Experimental results (2)

$2S(S)$	Types	Sols	Errs
46(23)	(1,1,11), (1,3,5)	568	3
	(1,2,7), (1,3,5)	92	5
54(27)	(1,1,13), (3,3,3)	2	0
	(1,3,6), (3,3,3)	1	0
58(29)	(1,1,14), (1,4,5)	37	0
62(31)	(1,3,7), (2,3,5)	5	0
64(32)	(1,2,10), (2,2,7)	50	2
	(2,2,7), (2,4,4)	6	0
70(35)	(1,1,17), (1,5,5)	3	0
	(1,2,11), (1,3,8)	11	0
88(44)	(2,2,10), (1,4,8)	2	0

Twister algorithm[1] to generate random numbers. Our results are summarized in Tables 2 and 3. In Table 2, "$2S(S)$" denotes the (half) size of a polygon, "$|P(S)|$" denotes the number of distinct box types, "RG" denotes the number of random generations, "Sols" denotes the number of simple polygons that can fold to two incongruent orthogonal boxes, and "Errs" denotes the number of non simple polygons. For example, for $P(11)$, the algorithm generates around 6.7×10^7 nets of boxes of size $(1, 1, 5)$ *or* $(1, 2, 3)$, and we have 556 outputs. Among them, 15 polygons have a hole, and hence we have 541 distinct simple polygons that can fold to boxes of size $(1, 1, 5)$ *and* $(1, 2, 3)$. In total, we have 2139 distinct simple polygons that can fold to two incongruent orthogonal boxes. For each S with $|P(S)| > 2$, more details can be found in Table 3. All cases are checked in parallel on the machine, and the computations take from a few days to a few weeks (we stopped execution when each process requires too much memory). Some solutions are illustrated in Figure 4, and all solutions can be found at `http://www.jaist.ac.jp/~uehara/etc/origami/nets/index-e.html`.

3.2 Random simultaneous unfolding

The first algorithm generates each spanning tree of a net uniformly at random. This means that a "fat" net, which contains a large rectangle, appears with higher probability than a "thin" net, which consists of small rectangles. Thus we developed another algorithm which was based on a different idea. The second algorithm keeps plural boxes, say B_1 and B_2, of a same size with their correspondence of unit squares. First, the algorithm picks up two unit squares s_1^1 on B_1 and s_2^1 on B_2 uniformly at random. These squares s_1^1 and s_2^1 have the "direction", which is also determined uniformly at random. Now the

[1] `http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html`

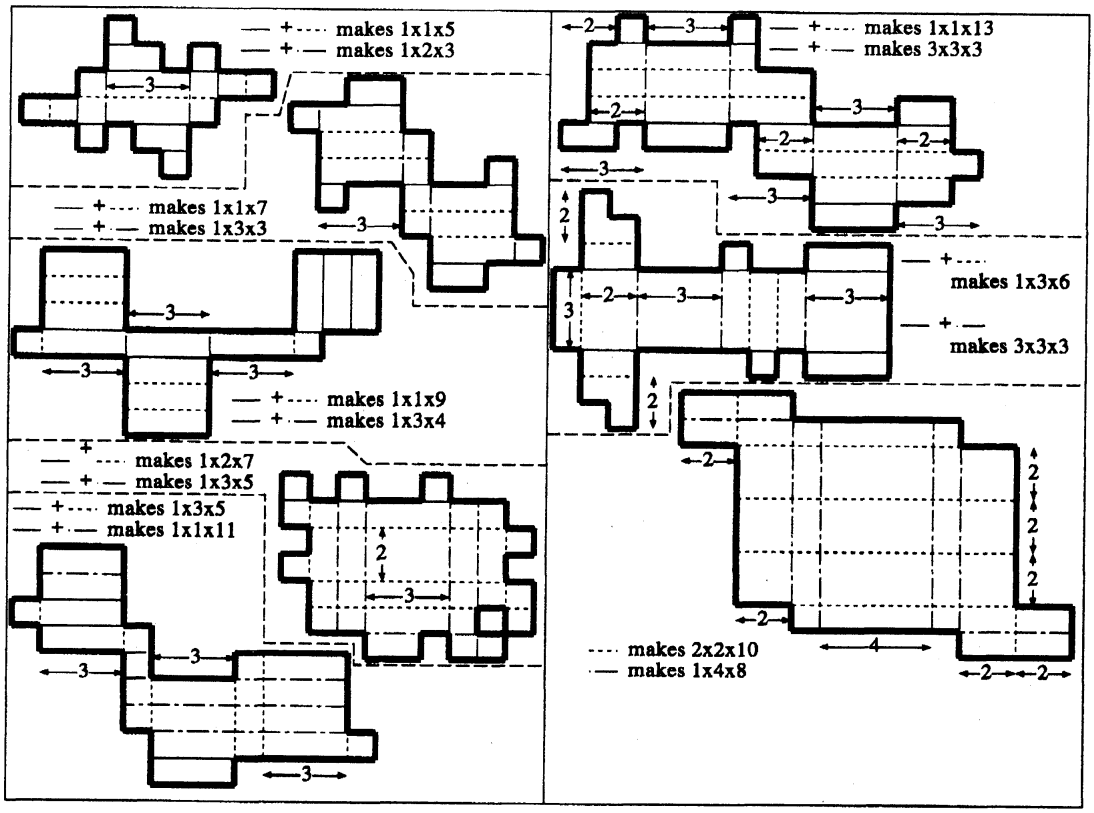

Figure 4: A part of solutions.

algorithm regards them as the same unit square on the resultant polygon P, which will be extended to a common net of B_1 and B_2. The algorithm checks each external boundary edge e of the current P. The neighbor square s of P at the edge e corresponds to two unit squares s' on B_1 and s'' on B_2, respectively. If both of them are not yet in P, the edge is *extensible*. The algorithm then picks up an extensible edge of P uniformly at random, and add the square s to P. Repeating this process, the algorithm meets one of two possible cases. First, P contains all squares. Then P is a common net of B_1 and B_2. Second, P has no extensible edge before the first case. This case is fail to find a solution. Then the algorithm halts, and we will start it again. Precisely, it is described in Algorithm 2 for finding a polygon that can fold to two incongruent boxes.

It is easy to extend the algorithm to handle with three or more boxes. This algorithm always outputs a correct answer.

3.2.1 Experimental results

The second algorithm was run on a desktop (Intel Xeon CPU 5110 1.60GHz Windows Vista). For each pair of two orthogonal boxes of the same size, the algorithm tried to find common nets 1×10^7 times. The results are summarized in Table 4. In Table 4, "$2S(S)$" denotes the (half) size of a polygon, "Types" denotes the distinct box types, "Time" denotes the computation time of 1×10^7 trials, "Sols" denotes the number of

Input : S with $|P(S)| > 1$;
Output: Polygon of size $2S$ that folds to two boxes;
choose desired types $t_1 = (a_1, b_1, c_1)$ and $t_2 = (a_2, b_2, c_2)$ in $P(S)$;
pick up two start unit squares s_1^1 on the box B_1 of type t_1 and s_2^1 on the box B_2 of type t_2 uniformly at random;
set polygon P to contain a unit square corresponding to both of s_1^1 on B_1 and s_2^1 on B_2;
repeat
  check each external boundary edge e of P if e is extensible on B_1 and B_2;
  if *no edge is extensible* **then** output "fail to find" and halt;
  pick up an extensible edge e of P uniformly at random;
  attach one unit square s to P at e;
until *all squares on* B_1 *and* B_2 *are attached to* P ;
output P.

Algorithm 2: Outline of random simultaneous unfolding

Table 4: Experimental results (3)

$2S(S)$	Types	Time(sec)	Sols	Errs
22(11)	(1,1,5),(1,2,3)	194	6495	29
30(15)	(1,1,7),(1,3,3)	290	1142	0
34(17)	(1,1,8),(1,2,5)	371	11291	2
38(19)	(1,1,9),(1,3,4)	402	2334	0
54(27)	(1,1,13),(3,3,3)	659	1735	1
54(27)	(1,1,13),(1,3,6)	648	1806	1
54(27)	(1,3,6),(3,3,3)	878	387	3
88(44)	(1,4,8),(2,2,10)	1740	218	0
88(44)	(2,2,10),(2,4,6)	1782	86	1
Total	-	6964	25494	37

simple polygons that can fold to two given orthogonal boxes, and "Errs" denotes the number of non simple polygons. For example, for $P(17)$, during 1×10^7 independent trials, the algorithm found 11291 simple polygons and 2 non-simple polygons that can fold to two boxes of sizes $1 \times 1 \times 8$ and $1 \times 2 \times 5$ in 371 seconds. In total, we have 25494 distinct simple polygons that can fold to two incongruent orthogonal boxes.

For $P(11)$, a graph of the number of trials and the number of solutions is depicted in Figure 5. According to the graph, the number of nets that can fold to boxes of sizes $1 \times 1 \times 5$ and $1 \times 2 \times 3$ seems to be around 7×10^3.

Notes: We note that some values of S are related; for example, the solutions for $P(11)$ give the solutions for $P(44)$ by dividing a unit square into four unit squares. For example, by the first algorithm, although we have 541 solutions for $P(11)$ after 6.7×10^7 random generations (it takes 3 days), we have only two solutions for $P(44)$ after 217.0×10^7 random generations (it takes 1 month). After these experiments, we still have no polygon that can fold to three (or more) incongruent orthogonal boxes.

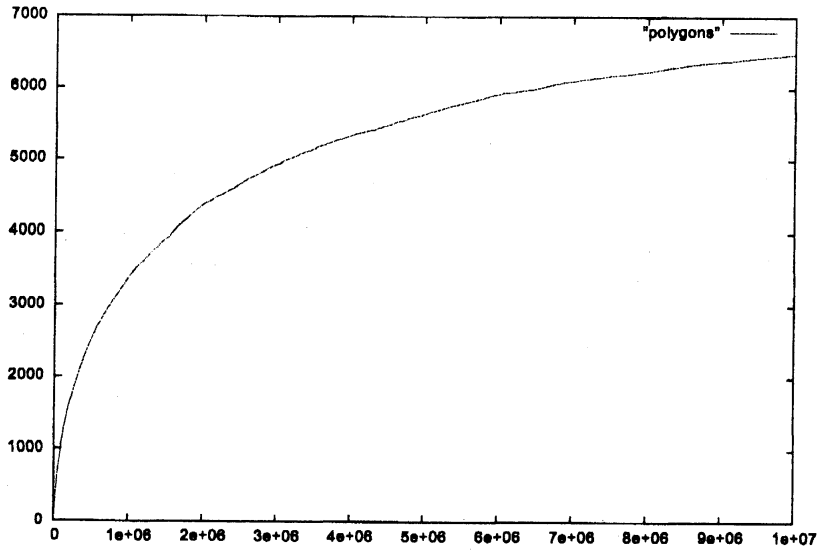

Figure 5: Ratio of the number of solutions to the number of trials.

4 Special patterns

In this section, we show some special patterns found in the solutions.

Tiling: The discovered polygonal patterns reminded us of *tiling*. Indeed, there exists a simple polygon that can fold to two incongruent orthogonal boxes and it forms a tiling. The polygon in Figure 6 can fold to two boxes of size $1 \times 1 \times 8$ and $1 \times 2 \times 5$, and it tiles the plane.

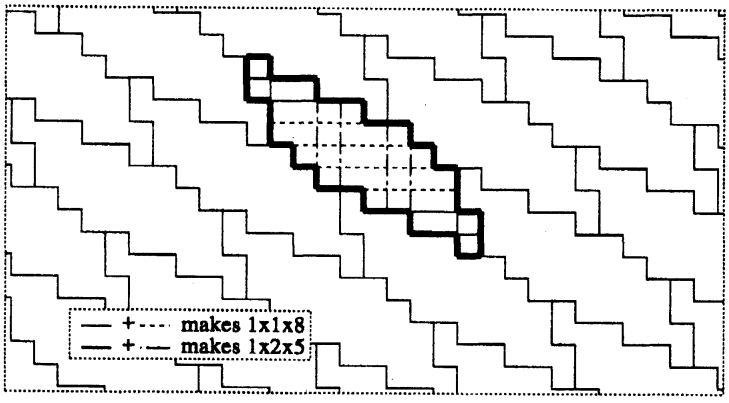

Figure 6: Polygon folding to two boxes of $1 \times 1 \times 8$ and $1 \times 2 \times 5$, and tiling the plane.

A polygon is called a *double packable solid* if it tiles the plane and a polyhedron from the polygon fills the space [4, Section 3.5.2]. It is easy to see that every orthogonal box fills the space. Therefore, the polygon in Figure 6 forms two double packable solids!

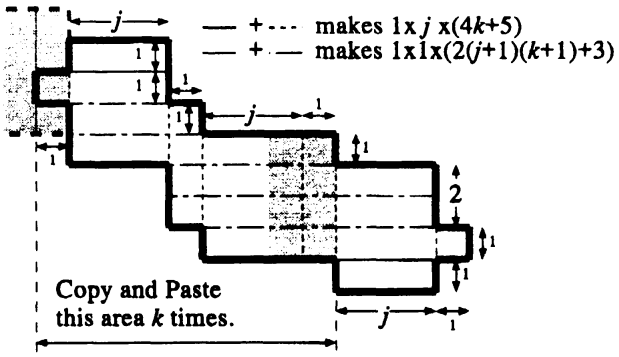

Figure 7: Polygon folding to two boxes of $1 \times 1 \times (2(j+1)(k+1)+3)$ and $1 \times j \times (4k+5)$.

Disjoint crease patterns: There exists a simple polygon that can fold to two incongruent orthogonal boxes and that foldings to two boxes are disjoint; the last polygon in Figure 4 satisfies the property.

Cross-free patterns: There exists a simple polygon that can fold to two incongruent orthogonal boxes and that foldings to two boxes are cross free. The second last polygon in Figure 4 satisfies the property. We note that the previously known results in [2] also satisfy the property.

We have not checked if there exists a simple polygon such that foldings are disjoint and cross free.

5 Infinite polygons

A natural question is whether or not there are infinite distinct[2] polygons that can fold to plural boxes? The answer is "yes." Some polygons obtained by the experiments can be generalized. From one of them, we have the following theorem.

Theorem 4 *For any positive integers j and k, there is a distinct polygon that can fold to two incongruent orthogonal boxes of sizes $1 \times 1 \times (2(j+1)(k+1)+3)$ and $1 \times j \times (4k+5)$.*

Proof. For any positive integers j and k, Figure 7 gives a polygon that satisfies the condition. The first parameter j just stretches each rectangle in Figure 7 in the same rate, which has no effect to construct two distinct boxes; two folding ways are similar to the polygon in Figure 6. For the second parameter k, we copy in the leftside polygon in Figure 7 and glue it to the leftmost square (with overlapping at gray areas) and repeat it k times. Then, the folding way of the box of size $1 \times 1 \times (2(j+1)(k+1)+3)$ is essentially the same for every k; just we roll up four unit squares vertically. The folding way of the box of size $1 \times j \times (4k+5)$ depends on k. We spiral up the polygon k times, and obtain vertically long rectangles. By these foldings, we have two distinct boxes of different sizes from a polygon. □

[2]Precisely, *distinct* means $gcd(a, b, c, a', b', c') = 1$ for two boxes of size $a \times b \times c$ and $a' \times b' \times c'$.

Corollary 5 *There exist an infinite of distinct polygons that can fold to two incongruent orthogonal boxes.*

6 Generalizations

Since $|P(S)| = 1$ for $S < 11$, there is no smaller net than the solutions for $P(11)$. If we admit to congruent orthogonal boxes, we have smaller one. See Figure 8; we can make three orthogonal congruent boxes in three different ways (bold lines are cut lines). Each

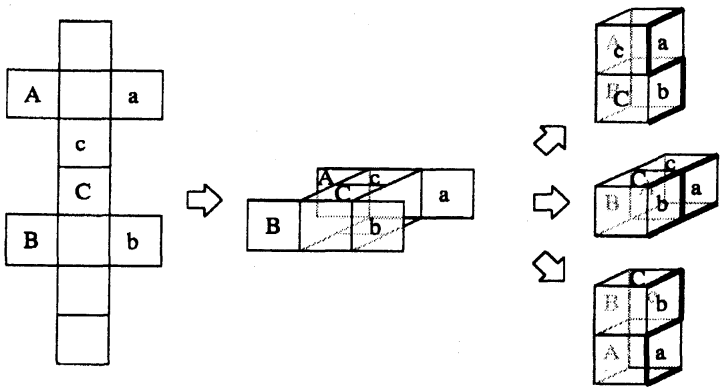

Figure 8: Three folding ways of a net

of all eleven distinct nets of a unit cube can fold to the unit cube in a unique way. Thus the net in Figure 8 is a smallest net that can fold to an orthogonal box in plural ways.

Next we consider the net that can fold to plural incongruent orthogonal polyhedra which may be concave. In this case, there is a net of size 18 which can fold to seven different (and all possible) orthogonal polyhedra. This is sold as the Cubigami puzzle, originated by Miller and Knuth (Figures 9 and 10). This puzzle won an Honorable Men-

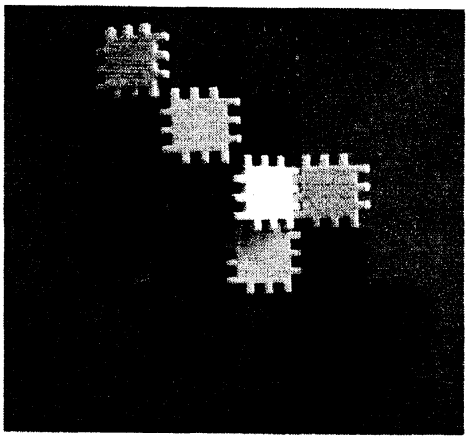

Figure 9: The Cubigami puzzle.

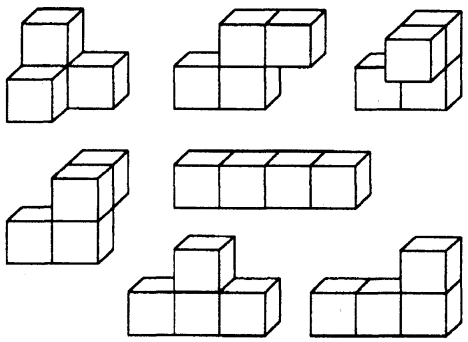

Figure 10: Seven possible polyhedra of Cubigami.

tion in the 2005 World Puzzle Contest (see `http://www.puzzlepalace.com/resources/t9/index.html` for further details). The Cubigami is not a simple net (we remind that Lemma 1 holds only for convex polyhedra), but there are 68 simple nets which can fold to seven orthogonal polyhedra according to the results by Knuth. One of them is depicted in Figure 11, which is found by our second algorithm after around 1×10^8 trials. These

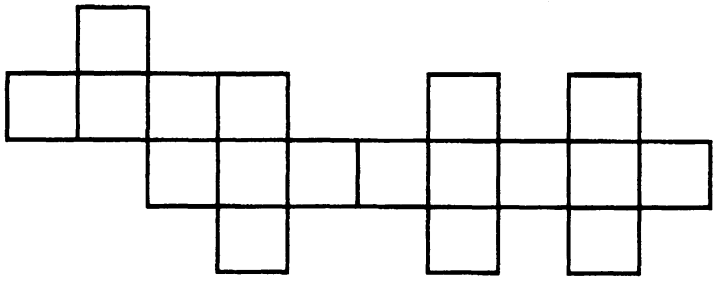

Figure 11: Net folding to seven orthogonal polyhedra.

patterns are not smallest in this context; extending the net in Figure 8, we can also obtain a smallest net that can fold to plural orthogonal polyhedra in three different ways; see Figure 12.

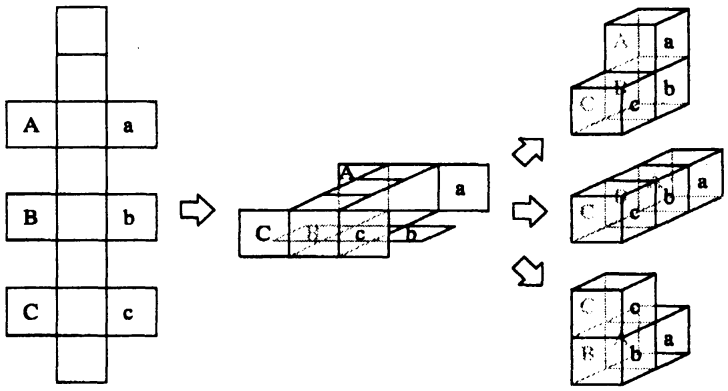

Figure 12: Three folding ways of a net

Now it is natural to ask if there exist nets that fold to arbitrary many orthogonal polyhedra. We give an affirmative answer.

Theorem 6 *For any positive integer k, there exists a simple net that folds to at least k orthogonal incongruent polyhedra.*

Proof. We use a polygon that fold to two boxes of size $1 \times 1 \times 8$ and $1 \times 2 \times 5$ in Figure 13(a). Two black squares in Figure 13(a) is called "lids" that are placed at opposite sides on both boxes. On the polygon, we replace two lids by two gadgets called "pipes" that consist of gray squares in Figure 13(b-e). Two pipes on a polygon make two "holes" on both boxes at opposite sides as Figure 13(c)(e). Using this gadget, we can connect k'

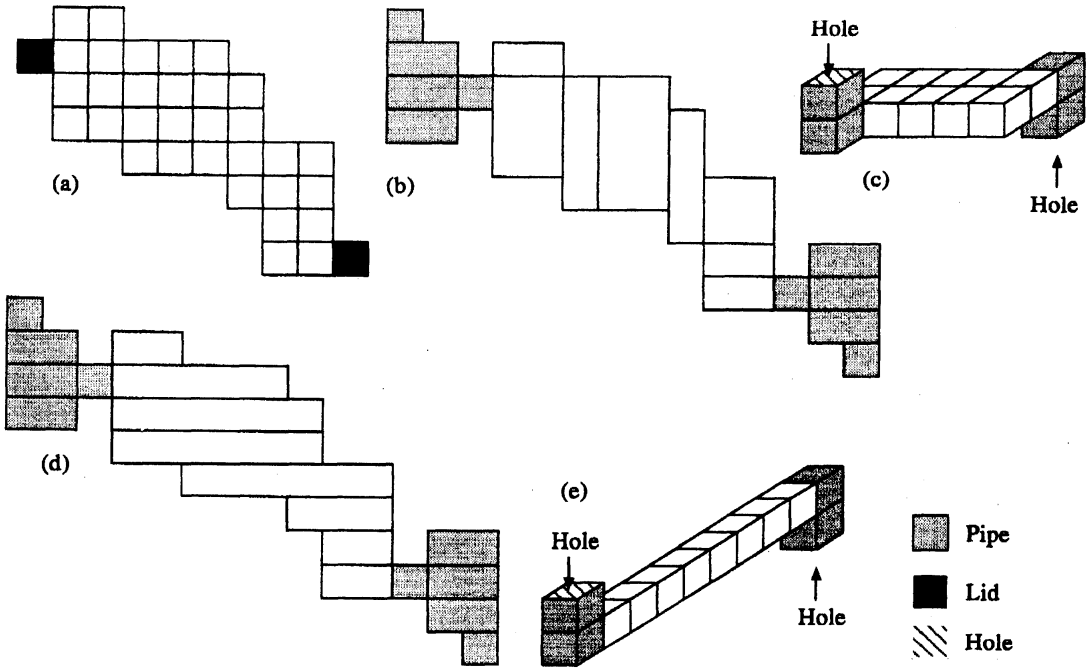

Figure 13: Gadget for a chain of polyhedra.

polygons as follows; we prepare k' gadgets and join them by their pipes. At two endpoint gadgets, we use the lids instead of pipes. See Figure 14 for $k' = 3$: It is easy to see that each gadget can fold to two boxes of size $1 \times 1 \times 8$ or $1 \times 2 \times 5$ independently. Among $2^{k'}$ polyhedra, $2^{\lceil k'/2 \rceil}$ polyhedra are symmetric (with respect to reverse). Hence we have $(2^{k'} - 2^{\lceil k'/2 \rceil})/2 + 2^{\lceil k'/2 \rceil} = 2^{k'-1} + 2^{\lceil k'/2 \rceil - 1}$ incongruent orthogonal polyhedra. (Hence Figure 14 gives us a net of $4 + 2 = 6$ incongruent orthogonal polyhedra.) Thus, letting k' large enough, we have the theorem. □

7 Concluding remarks

From the theoretical point of view, uniform random generation and/or enumeration of all simple nets for a given box of size $a \times b \times c$ are interesting problems. This is an extension work of one done by Knuth (see `http://www.puzzlepalace.com/resources/t9/index.html` for further details). However, those algorithms are not necessarily useful to find polygons that can fold to plural incongruent orthogonal boxes. Indeed we may have to search "similar" polygons heuristically to find such polygons. It is an open question if a polygon exists that can fold to three or more orthogonal boxes. The authors conjecture "yes;" through experience, there is a polygon that seems to be "close" to the answer. The polygon in Figure 15 can fold to two boxes of size $1 \times 1 \times 17$ and $1 \times 5 \times 5$ in the similar ways in Figure 7. Moreover, it also can fold to the box of size $1 \times 3 \times 8$ with only two overlapping squares (and hence with two holes); a and b overlap with a' and b', respectively (with a cut between a and a').

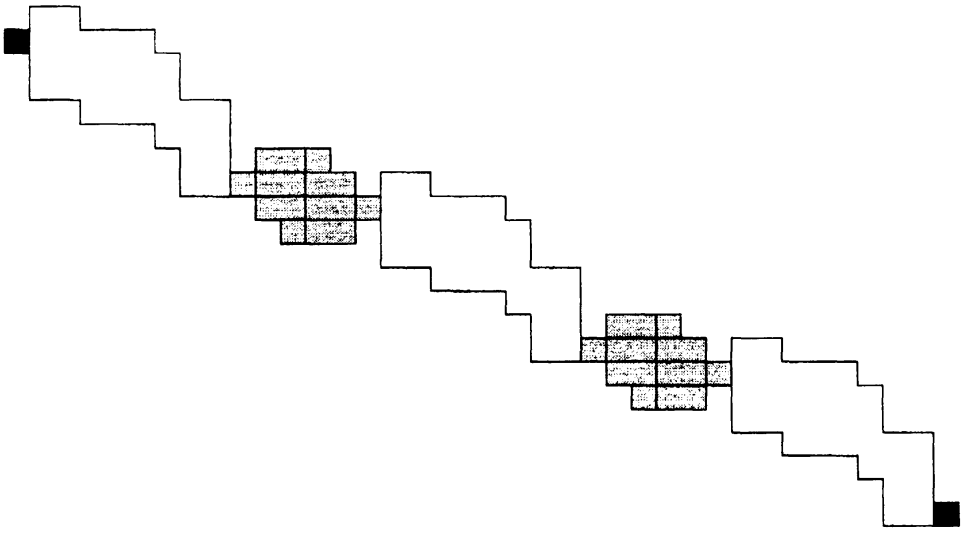

Figure 14: Polygon obtained by joining three gadgets.

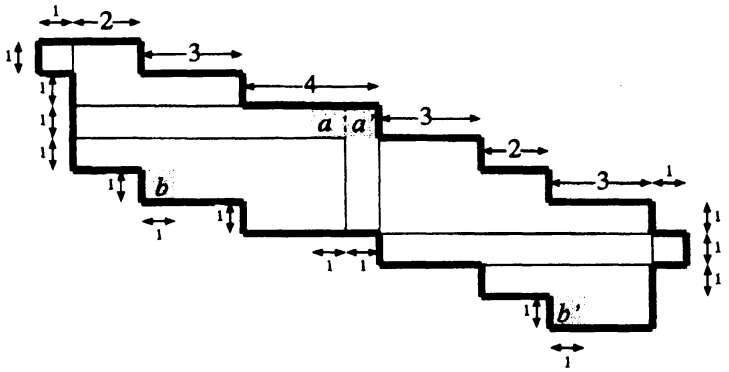

Figure 15: A polygon folding to two boxes of $1 \times 1 \times 17$, $1 \times 5 \times 5$, and "close" to $1 \times 3 \times 8$.

Acknowledgements

The authors thank the following people. Joseph O'Rourke, who gave an interesting talk at JAIST, which involved the first author in this work. Erik Demaine, who kindly sent the note [2] to the authors. Yoshio Okamoto, who gave the basic idea of joining the gadgets like Figure 14. Stefan Langerman, who pointed out that the parameter j is available in Figure 7. Tsuyoshi Ito, who found an error of the number of valid polygons on the Web cite for the case $S = 11$.

References

[1] T. Asano and H. Tanaka Constant-Working Space Algorithm for Connected Components Labeling IEICE Technical Report, COMP2008-1:1–8, 2008.

[2] T. Biedl, T. Chan, E. Demaine, M. Demaine, A. Lubiw, J. I. Munro, and J. Shallit. Notes from the University of Waterloo Algorithmic Problem Session. Sept. 8 1999.

[3] E. D. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra.* Cambridge University Press, 2007.

[4] M. Kano, M.-J. P. Ruiz, and J. Urrutia. Jin Akiyama: A Friend and His Mathematics. *Graphs and Combinatorics*, 23[Suppl]:1–39, 2007.

[5] A. Lubiw and J. O'Rourke. When Can a Polygon Fold to a Polytope? Technical Report Technical Report 048, Department of Computer Science, Smith College, 1996.