Efficient Enumeration of Flat-Foldable Single Vertex Crease Patterns

Koji Ouchi and Ryuhei Uehara

School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), {k-ouchi,uehara}@jaist.ac.jp.

Abstract. We investigate the following computational origami problem; the input is a positive integer n. We then draw n lines in a radial pattern. They are incident to the central point of a sheet of paper, and every angle between two consecutive lines is equal to $2\pi/n$. Each line is assigned one of "mountain," "valley," and "flat" (or consequently unfolded), and only flat-foldable patterns will be output. We consider two crease patterns are the same if they can be equal with rotations and reflections. We propose an efficient enumeration algorithm for flat-foldable single vertex crease patterns for given n. In computational origami, there are well-known theorems for flat-foldability; Kawasaki Theorem and Maekawa Theorem. However, they give us necessary conditions, and sufficient condition is not known. Therefore, we have to enumerate and check flat-foldability one by one using the other algorithm. In this paper, we develop the first algorithm for the above stated problem by combining these results in nontrivial way, and show its analysis of efficiency.

1 Introduction

Recent origami is a kind of art, and origamists around the world struggle with their problems; what the best way to fold origami models are. One of the problems is that a unit of angle that appears in the origami model. Some origamists restrict themselves to use only multiples of $22.5°$, $15°$ or some other specific angle which divides $360°$. A not-simple example is shown in Fig. 1, which is designed based on $15°$ unit angle by the first author. Once origamists fix the unit angle as $(360/n)°$ for suitable positive integer n, their designs are restricted to one between quite real shapes and abstract shapes, which is the next matter in art.

When we are given a positive integer n, we face a computational origami problem which is interesting from the viewpoints of mathematics and algorithms. We consider the simplest origami model; all crease lines are incident to the single vertex at the center of origami, and each angle between two creases is a multiple of $(360/n)°$. We are interested in only flat-foldable crease patterns.

When mountain or valley folding is assigned to every crease pattern, the flat-foldability can be computed in linear time [1]. However, its rigorous proof is not so simple, which is the main topic of Chapter 12 in [3]. Roughly speaking, the algorithm repeats to fold and glue locally smallest angle in each step in general.

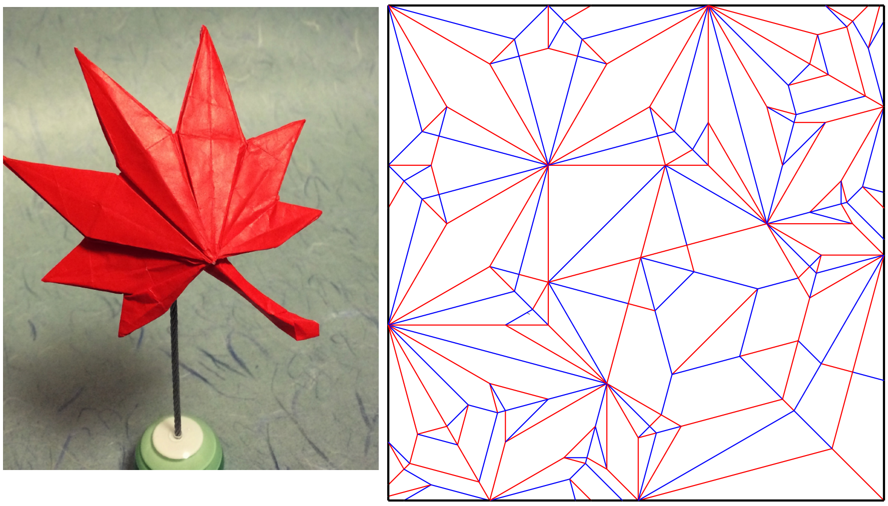

Fig. 1. "Maple leaf" designed and folded by the first author, and its crease pattern is based on 15° unit angle.

In other words, we have no mathematical characterization for this problem, and we have to check one by one.

When mountain or valley folding are not given to the crease pattern, the problem has different issue. Hull investigated this problem [5] from the viewpoint of counting. Precisely, he considered the number of flat-foldable assignments of mountain and valley to a given crease pattern of n lines which were incident to the single vertex. He gave tight lower bound and upper bound. These bounds are given in two extreme situations; one is given in the case that all n angles are different, and the other is given in the case that all n angles are equal to each other. From the viewpoint of origami design, we are interested in the case between these two extreme situations. To deal with reasonable situations between extreme ones, we slightly modify the input of the problem. The input of our problem is a positive integer n, and we restrict ourselves to the single vertex folding of unit angle $(360/n)^\circ$. In order to investigate our problem, we assign one of three labels — "mountain," "valley," and "flat" — to each of n creases. When a crease line is labeled by "flat," this crease line is not folded in the final folded state. In this way, we can deal with the single vertex crease patterns of unit angle equal to $(360/n)^\circ$, which is more realistic situation from the viewpoint of origami design.

Our aim is to enumerate all distinct flat-foldable assignments of the three labels to n creases. In other words, our algorithm eventually enumerates all flat-foldable crease patterns with labels of "mountain" and "valley" of unit angle $(360/n)^\circ$. We consider the sheet of paper is a disk, the vertex is at the center of the disk, and two crease patterns are the same if they can be equal with rotations and reflections (i.e., including turning over and exchanging all mountains and valleys). Our algorithm enumerates all distinct crease patterns under this assumption.

For flat-foldability of a given crease pattern, there are two well-known theorems in the area of computational origami, which are called "Kawasaki Theorem" and "Maekawa Theorem" (see [3, Chapter 12] for further details):

Theorem 1 (Kawasaki Theorem). *Let θ_i be an angle between the ith crease line and the $i+1$st crease line. A single-vertex crease pattern defined by angles $\theta_1 + \theta_2 + \cdots + \theta_{n'} = 360°$ is flat-foldable if and only if n' is even and the sum of the odd angles θ_{2i+1} is equal to the sum of the even angles θ_{2i}, or equivalently, either sum is equal to $180°$: $\theta_1 + \theta_3 + \cdots + \theta_{n'-1} = \theta_2 + \theta_4 + \cdots + \theta_{n'} = 180°$.*

We note that Kawasaki Theorem gives a necessary and sufficient condition for flat-foldability, but mountain-valley assignments are not given. That is, we have to compute foldable assignments for foldable crease pattern satisfying Kawasaki Theorem. In order to compute a flat-foldable assignment, we can use Maekawa Theorem:

Theorem 2 (Maekawa Theorem). *In a flat-foldable single-vertex mountain-valley pattern defined by angles $\theta_1 + \theta_2 + \cdots + \theta_{n'} = 360°$, the number of mountains and the number of valleys differ by ± 2.*

We again note that Maekawa Theorem is necessary condition, but not sufficient condition.

In the last decades, enumeration algorithms have been well investigated, and many efficient enumeration algorithms have been given. Using the techniques with above properties of origami, we show an enumeration algorithm for flat-foldable crease patterns for given n, where n is the maximum number of crease lines of unit angle $(360/n)°$. As far as the authors know, this is the first algorithm for this realistic computational origami problem. As a result, we succeeded to enumerate flat-foldable crease patterns up to $n = 32$ in a reasonable time.

2 Preliminaries and Outline of Algorithm

Based on the Kawasaki Theorem and Maekawa Theorem, for given n, we can design the outline of our enumeration algorithm as follows:

(1) Assign "crease" or "flat" to each of n crease lines incident to the single-vertex so that Kawasaki Theorem is satisfied for this assignment.
(2) For each "crease", assign "mountain" or "valley" so that Maekawa Theorem is satisfied.
(3) Output this pattern if this crease pattern is flat-foldable.

Essentially, the outline consists of two different kinds of enumeration problems in phases 1 and 2, and flat-foldability checking in phase 3.

A simple example is given in Fig. 2. For $n = 8$, we first generate all possible crease lines in phase 1 which is described in binary string (in the figure, we only show one, but there are exponentially many). Here we assign "0" means "crease"

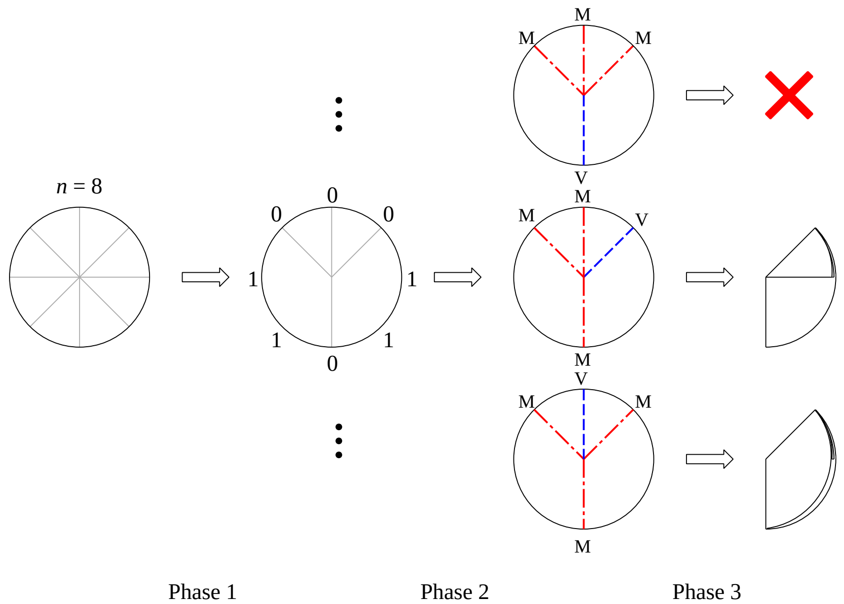

Fig. 2. Simple example for $n = 8$.

and "1" means "flat." Therefore, for a string 00011011, we have four crease lines in the shape in Fig. 2. Then, in phase 2, we assign mountain (M) or valley (V) to each of crease line. In phase 3, we check if each crease pattern with M/V assignment flat-foldable or not, and output it if it is flat-foldable.

We have different issue in each phase, especially, in phases 1 and 2, we have to consider two different problems about symmetry (to reduce redundant output), and enumeration.

3 Description of Algorithm

Now we describe more details in each phase.

3.1 Phase 1: Assignment of "crease"/"flat"

In phase 1, we are given n crease lines, and we have to assign "crease" or "flat" to them so that Kawasaki Theorem is satisfied for this assignment. Since the crease pattern cannot be flat-folded for odd number n, without loss of generality, we assume that n is even hereafter.

We first describe "crease" by 0 and "flat" by 1, and consider binary string. Then it is easy to see that, before checking Kawasaki Theorem, we have to generate all binary strings over $\Sigma = \{0, 1\}$ efficiently under the equivalent of rotations and reflection. To consider this problem, we define *bracelet* as follows;

a *bracelet* is the lexicographically smallest element in an equivalence class of strings under string rotation and reversal. It is easy to observe that our problem is now enumeration of binary bracelet of length n. For bracelets, we have an optimal enumeration algorithm [6]:

Theorem 3 (Sawada2001). *Bracelets of length n can be enumerated in constant amortized time.*

That is, the algorithm in [6] runs in time proportional to the number of bracelets of length n.

On the other hand, it is easy to check whether Kawasaki Theorem holds or not for the crease pattern given by a bracelet. Thus we have the following theorem:

Theorem 4. *For a given even number n, phase 1 can be done in $O(nB(n))$ time, where $B(n)$ is the number of bracelets of length n.*

We note that the values of the function $B(n)$ are listed in the OEIS (The Online Encyclopedia of Integer Sequences; `http://oeis.org/`) as A000029, and it is given as

$$B(n) = \sum_{d \text{ divides } n} \frac{2^{n/d}\phi(d)}{2n} + 2^{n/2-1} + 2^{n/2-2} \tag{1}$$

for even number n, where ϕ is Euler's totient function.

3.2 Phase 1: Satisfying Kawasaki Theorem

After assigning "crease" or "flat" to each crease, we have to check whether these crease lines satisfy Kawasaki Theorem or not. Kawasaki Theorem states that the alternating sum of angles is equal to 0. This notion corresponds to a kind of necklace in a nontrivial way as follows. We first observe that each angle θ_i is $k \times \frac{360}{n}^\circ$ for given even n. That is, θ_i consists of k unit angles. Now we regard θ_i as the integer k, and we consider $\theta_1, \theta_3, \ldots$ as "white," and $\theta_2, \theta_4, \ldots$ as "black." Then, each sequence of angles corresponds to a necklace with n beads such that the number of white beads is equal to the number of black beads. That is, each sequence of n' creases satisfying Kawasaki Theorem corresponds to a necklace with n beads such that (1) the necklace consists of $n/2$ white beads and $n/2$ black beads, and (2) the number of runs of white beads (and hence black beads) is n'. This notion is investigated as "balanced twills on n harnesses" in [4], and labeled as A006840 in OEIS. For $k = n/2$, the number is given as follows:

Theorem 5 (Hoskins and Street 1982). *The number of distinct balanced twills on $n = 2k$ harnesses is*

$$B'(2k) = \frac{1}{8k}\left\{ \sum_{\substack{d \text{ divides } n \\ d=2e}} \phi\left(\frac{k}{e}\right)\binom{2e}{e} + \sum_{d \text{ divides } k} \phi\left(\frac{2k}{d}\right)2^d + 2k\binom{2\lfloor k/2\rfloor}{\lfloor k/2\rfloor} + k2^k \right\}. \tag{2}$$

We note that Equation 2 just gives us the numbers for each n, however, they will not give us concrete set of creases. Therefore, we have to enumerate them by ourselves.

3.3 Phase 2: Assignment of "mountain"/"valley"

In this phase, we inherit a binary string s of length n from the phase 1, which describes "crease" (=0) or "flat" (=1). We note that s is the lexicographically smallest element under rotation and reversal. Then we translate this binary string to a set of other strings that represent the assignment of "mountain" and "valley" and angles to the binary string. The first step can be described as follows:

(2a) For each adjacent pair of 0s, we replace 1s between them by the number of 1s plus 1. For example, the string 00011011 in Fig. 2 is replaced by $0\underline{1}0\underline{1}0\underline{3}0\underline{3}$, where the positive (underlined) numbers describe the number of unit angles there.

Then we assign mountain ($= M$) and valley ($= V$) to each of 0, but here we only consider the assignments that satisfies Maekawa Theorem. Maekawa Theorem says that the number of Ms and the number of Vs differ 2. To avoid symmetry case, we can assume that (the number of Ms)$-$(the number of Vs)$= 2$. Thus next step is described as follows:

(2b) For the resulting string over $\{0, 1, 2, \ldots\}$, assign all possible Ms and Vs to each of 0 such that the number of Ms is two larger than the number of Vs. For example, for the string 01010303, we obtain the set of strings $\{V1M1M3M3, M1V1M3M3, M1M1V3M3, M1M1M3V3\}$.

In the resulting set of strings, we can have equivalent crease pattern. Precisely, if the original crease pattern (or string s) has equivalent pattern under rotation and reversal, they produce equivalent crease pattern. For example, in the set of strings $\{V1M1M3M3, M1V1M3M3, M1M1V3M3, M1M1M3V3\}$, we can observe that $V1M1M3M3$ is a crease pattern which is the mirror image of a crease pattern $M1M1V3M3$, hence we consider they are equivalent. (In Fig. 2, after Phase 2, the central crease pattern has its mirror image, and it should be omitted.) To avoid this equivalent patterns, we perform the following:

(2c) For the resulting string s' over $\{M, V, 1, 2, \ldots\}$ after phase 2b, we check whether s' is the lexicographically smallest element under rotation and reversal.

In the phase 2c, we have to check whether s' starts from the right position that gives the lexicographically smallest element among the set of strings obtained by rotation and reversal. For solving this nontrivial problem efficiently, Booth gives a linear time algorithm in [2]. Precisely, Booth's algorithm finds the right index that gives the lexicographically smallest string for a given circular string of length n in linear time. To deal with rotation and reversal, the step 2c can be implemented as follows;

(2c-1) For the resulting string s' over $\{M, V, 1, 2, \ldots\}$ after phase 2b, let s'^R is the reverse string of s'.

(2c-2) Find the right index i of circular string s' such that the string starting from the index i is the lexicographically smallest string among all ones given by s'. If i is not the first letter in s', we discard this s' since it is redundant.

(2c-3) Find the right index j of circular string s'^R. If the obtained string from s'^R by starting from the index j is lexicographically smaller than s', we discard s'.

(2c-4) If $s' > s'^R$ in lexicographically order, we discard s'; otherwise, this s' goes to phase 3 to be processed.

Summarizing up, we have the following theorem:

Theorem 6. *For a given crease pattern from phase 1 based on n unit angles, we can generate all distinct assignment of mountain and valley that satisfies Maekawa Theorem in $O(nC(n))$ time, where $C(n)$ is $\binom{n}{n/2-1}$.*

Proof. The number of lines in the crease pattern is at most n, and the number of M is 2 larger than the number of V. Thus, the number of strings s' over $\{M, V, 1, 2, \ldots\}$ of length at most n with the constraint for the number of Ms and Vs is at most $\binom{n}{n/2-1}$. The other management can be done in linear time, which implies the theorem. □

3.4 Phase 3: Test of Flat-Foldability

In this phase, we check if the resulting string s' over $\{M, V, 1, 2, \ldots\}$ is flat-foldable or not. For this problem, Demaine and O'Rourke give a linear time algorithm [3, Chap 12]. Therefore we can done this phase in linear time. Roughly, the algorithm works simple; it finds a local minimal angle, folds two creases on the boundary of this small fan-shape, glues it, and repeats until all creases are folded. However, the correctness of this simple algorithm is not easy; as mentioned at the footnote in [3, page 204], the rigorous proof is first done by Demaine and O'Rourke in [3, Chap 12].

3.5 Analysis of Algorithm

The correctness of our algorithm relies on the algorithms used in each phase as described above. Therefore, we consider its time complexity since space complexity is clearly $O(n)$. Our main theorem is the following:

Theorem 7. *For a given n, all distinct flat-foldable mountain and valley assignments of unit angle $(360/n)^\circ$ can be done in $O(n^2 B(n)\binom{n}{n/2-1}))$ time, where $B(n)$ is the number of bracelets of length n (see Equation 1).*

Table 1. The number of enumerated patterns. The number of lines in a pattern is even number from 2 to n.

n	Phase 1	Phase 2	Phase 3
4	2	2	2
6	3	7	6
8	7	27	20
10	13	143	87
12	35	837	420
14	85	5529	2254
16	257	38305	12676
18	765	276441	73819
20	2518	2042990	438795
22	8359	15396071	2649555
24	28968	117761000	16188915
26	101340	912100793	99888892
28	361270	7139581543	621428188
30	1297879	56400579759	3893646748
32	4707969	449129924559	24548337096
34	17179435	-	-
36	63068876	-	-
38	232615771	-	-
40	861725794	-	-
42	3204236779	-	-

Table 2. Distribution of the patterns obtained at Phase 1.

n	#line of each pattern										sum
	2	4	6	8	10	12	14	16	18	20	
4	1	1									2
6	1	1	1								3
8	1	3	2	1							7
10	1	3	6	2	1						13
12	1	6	13	11	3	1					35
14	1	6	26	30	18	3	1				85
16	1	10	46	93	74	28	4	1			257
18	1	10	79	210	275	145	40	4	1		765
20	1	15	124	479	841	716	280	56	5	1	2518

4 Experimental Results

As shown in Theorem 7, the upper bound of the number of distinct flat-foldable maintain and valley assignment of unit angle $(360/n)^\circ$ is exponential, and exact values for each n are difficult to estimate theoretically. Therefore, we here show experimental results.

4.1 The Number of Crease Patterns

Table 1 and Fig. 3 show the exact number of distinct patterns obtained at each phase. As mentioned in Section 3.2, the result of Phase 1, which enumerates "crease"/"flat" assignments satisfying Kawasaki theorem, coincide with the sequence of the sequence labeled as A006840 in OEIS. The counting results at the other phases are different from any existing sequences in OEIS, that is, we find totally new sequences in this study.

4.2 Solution Space

We measure the rate of number of solutions against that of possible patterns at each phase (see Table 3 and Fig. 4), which suggests how difficult the problems

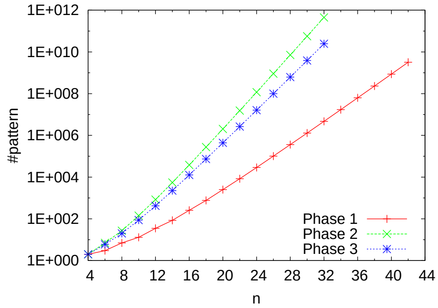

Fig. 3. The number of enumerated patterns. The number of lines in a pattern is even number from 2 to n.

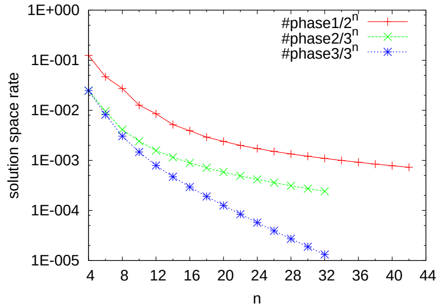

Fig. 4. The rate of solutions against possible patterns at each phase.

are. We can see that the solution spaces are very sparse for all phases. There are 2^n possible "crease"/"flat" assignments at Phase 1. Only about 4.7% is the solution for Phase 1 if $n = 6$. It decreases significantly and gets less than 1% for $n \geq 12$. The rates at Phase 2 and Phase 3 are against 3^n since we consider "mountain"/"valley"/"flat" assignment at that phases. The two rates tend to decrease similarly to that of Phase 1, and are much smaller, e.g., 2.5% for Phase 2 when $n = 6$. Such rate at every phase seems to be exponential to n according to Fig. 4.

5 Concluding Remarks

We develop the first algorithm for enumerating distinct flat-foldable single vertex crease patterns. We also show by experiments how many such patterns there are, which is done the first time as well. Improving the algorithm and investigating further for the counting problems are the future works. For example, rather than Sawada's algorithm in Theorem 3, enumeration of the sequences stated in Theorem 5 directly seems to improve the running time of our algorithm drastically.

We also examine the rates in each phase; experimentally, they seem decrease exponentially. Nevertheless, we conjecture that there are exponentially many flat-foldable crease patterns. Showing theoretical lower bounds and upper bounds are also remained open.

Acknowledgement

We would like to thank Yota Otachi for his fruitful discussions and comments.

Table 3. #solution/#possible at each phase.

n	#Phase1/2^n	#Phase2/3^n	#Phase3/3^n
4	0.125	0.024691358	0.024691358
6	0.046875	0.009602195	0.008230453
8	0.02734375	0.004115226	0.003048316
10	0.012695313	0.002421718	0.001473353
12	0.008544922	0.001574963	0.000790304
14	0.005187988	0.001155977	0.000471255
16	0.003921509	0.000889847	0.000294471
18	0.002918243	0.000713543	0.00019054
20	0.002401352	0.000585924	0.000125845
22	0.001992941	0.000490617	8.44317E-05
24	0.001726627	0.000416957	5.73202E-05
26	0.001510084	0.000358831	3.92975E-05
28	0.001345836	0.000312088	2.71641E-05
30	0.001208744	0.000273934	1.89112E-05
32	0.001096159	0.000242377	1.32477E-05
34	0.000999975	-	-
36	0.000917773	-	-
38	0.000846251	-	-
40	0.000783735	-	-
42	0.000728559	-	-

References

1. E. M. Arkin, M. A. Bender, E. D. Demaine, M. L. Demaine, J. S. B. Mitchell, S. Sethia, and S. S. Skiena. When can you fold a map? *Comput. Geom. Theory Appl.*, 29(1):23–46, 2004.
2. K. S. Booth. Lexicographically Least Circular Substrings. *Information Processing Letters*, 4,5:240–242, 1980.
3. E. D. Demaine and J. O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra.* Cambridge University Press, 2007.
4. W. D. Hoskins and A. P. Street. Twills on a Given Number of Harnesses. *J. Austral. Math. Soc. (Series A)*, 33:1–15, 1982.
5. T. Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatoria*, 67:175–187, 2003.
6. J. Sawada. Generating Bracelets in Constant Amortized Time. *SIAM J. Comput.*, 31(1):259–268, 2001.