Journal for Geometry and Graphics
Volume 15 (2011), No. 2, 195–201.

A Method for Designing Crease Patterns for Flat-Foldable Origami with Numerical Optimization

Jun Mitani

Graduate School of Systems and Information Engineering,
University of Tsukuba / JST ERATO, Tsukuba Ibaraki 305-8573 Japan
email: mitani@cs.tsukuba.ac.jp

Abstract. Origami is the art and science of making various shapes by simply folding a sheet of paper. When origami is studied as a geometrical problem, it is often assumed that the origami is folded flat. This flat foldability is an important property when we consider applying the techniques of origami to increase the portability and storage space of industrial products. We propose a method to generate new crease patterns that are flat-foldable by using a computer with the algorithm of numerical optimization. In this paper, we describe the five steps of our method. Although the numerical optimization sometimes fails to converge and we have to go back to the first step, we can generate many flat-foldable crease patterns containing randomly placed vertices. Our method can generate patterns that can be used not only as new origami designs but also as problems for the origami puzzle *"Fold Me Up"*.

Key Words: Origami, Crease Pattern, Numerical Optimization, Flat Foldability
MSC 2010: 52B70, 68U05

1. Introduction

Origami is the art and science of making various shapes by simply folding a sheet of paper. Numerous mathematicians in the field of geometry have studied origami. Novel methods for designing origami by using a computer have been proposed in recent decades [1]. It was found that the theories of origami are applicable in the field of engineering.

When origami is studied as a geometrical problem, it is often assumed that the origami is folded into flat pieces. Most origami pieces commonly seen are actually flat folded. This flat foldability is an important property when we consider applying the techniques of origami to industrial products from the viewpoint of portability and storage space.

Typically, we focus on the *crease pattern* when studying the properties of a piece of origami. Here we define the crease pattern as follows.

ISSN 1433-8157/$ 2.50 © 2011 Heldermann Verlag

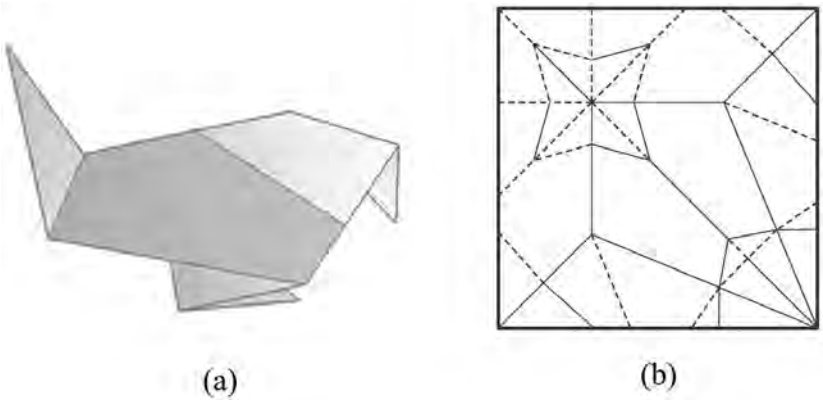

Figure 1: (a) An origami bird. (b) Crease pattern of (a).

The crease pattern is a set of mountain folded lines and valley folded lines appearing on a sheet of paper when a piece of origami is opened flat.

Figure 1 shows an example of an origami bird and the crease pattern. The solid lines and dashed lines represent mountain and valley lines, respectively.

Most recent complex origami artworks are designed from the crease patterns based on known rules. The rules are summarized in the origami design book [2]. Although most of existing origami pieces are flat foldable, randomly generated crease patterns are rarely flat foldable.

We can design flat-foldable crease patterns simply by re-arranging known patterns, such as the Miura pattern and the Waterbomb pattern (see Fig. 2(a), (b)). However, it is not straightforward to make completely new patterns that have no steady repetitions, such as the pattern shown in Fig. 2(c).

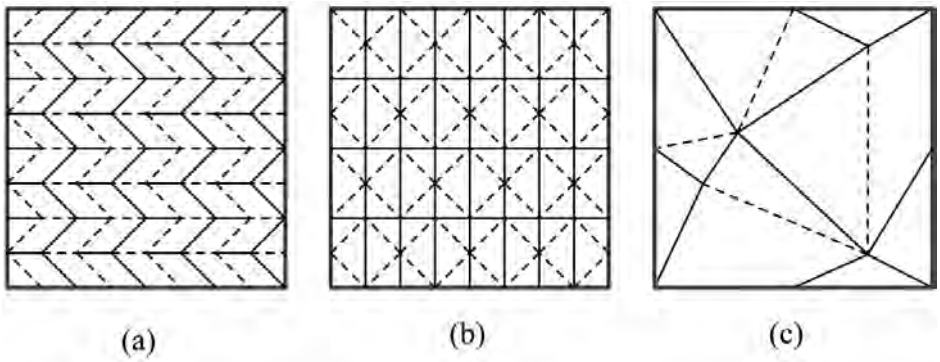

Figure 2: (a) The Miura pattern. (b) The Waterbomb pattern.
(c) A pattern generated from random points with our method.

In this paper, we propose a method to make new flat-foldable crease patterns such as the one shown in Fig. 2(c) by using a computer with the algorithm of numerical optimization. We believe that our method can lead to new origami designs.

An important problem to be solved is to design crease patterns that satisfy the conditions for which a crease pattern is flat foldable. A huge amount of studies related to origami

were surveyed and summarized in a book written by E.D. Demaine and J. O'Rourke [3]. Topics about flat foldability are discussed in Chapter 12 in this book. As stated in this chapter, the following, called Kawasaki's Theorem, is a necessary and sufficient condition for making a crease pattern that is locally flat foldable without distinguishing between mountain and valley folds.

Condition 1

1. The number of lines connecting to a single inner vertex is even.
2. The sums of alternating angles are 180 degrees.

The word *vertex* in condition 1.1 refers to the point where lines meet. Condition 1.2 is illustrated in Fig. 3.

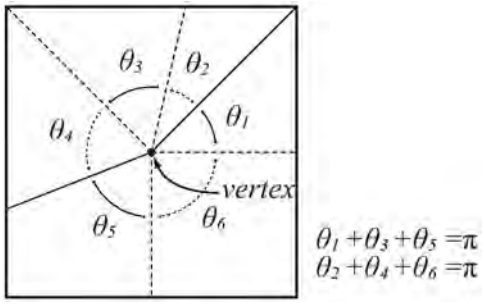

Figure 3: An example of a simple crease pattern which satisfies condition 1.2

Since condition 1 is only valid locally, even when the condition is satisfied for all inner vertices, it is possible that the crease pattern cannot be folded flat globally due to collisions of the parts of origami during assembly. It has been proved that the complexity of the problem of judging whether a crease pattern can be folded flat globally is NP hard [4]. Therefore, designing new crease patterns that are globally flat foldable seems difficult. It is not guaranteed that the patterns generated by our method satisfy the global foldability. We manually confirm whether the pattern is globally flat-foldable or not.

2. Proposed method

One possible solution for making new flat-foldable crease patterns is to fold a sheet of paper random by hand, open it, then trace the folded lines. Although this is a very simple approach, we propose a method for making new flat-foldable patterns systematically using a computer.

The flow of our method consists of the following five steps.

Step 1. Place vertices on edges and inside a sheet of paper.

Step 2. Generate lines that connect two vertices.

Step 3. Adjust the number of lines so that condition 1.1 is satisfied.

Step 4. Move the positions of the vertices so that condition 1.2 is satisfied.

Step 5. Validate that the pattern is globally flat foldable.

The details of each step are described next.

Step 1

We place a predefined number of vertices on edges and inside of a square sheet of paper. Four vertices are placed at the corners. The spaces between neighboring vertices are uniform in our implementation. Then, some vertices are placed inside the square. We randomly decide the positions. As the number of vertices increases, the pattern becomes more complex.

Step 2

We generate lines that connect the vertices. Since a line cannot cross another line, we use the Delaunay triangulation method to generate the lines. This method divides an area into triangles with a set of lines that connect two of the given vertices.

Step 3

We adjust the number of lines so that condition 1.1 is satisfied. Since it is difficult to add a line so that no line crosses any other line, we simply remove lines in this step. When the number of lines connecting to a single vertex is odd, we remove one line. Because condition 1.1 is not applied to the vertices placed on edges, we adjust from the center of the sheet of paper toward the edges. This operation decreases the total number of lines and generates polygonal areas surrounded by more than three lines.

Step 4

We move the positions of the vertices so that condition 1.2 is satisfied. We consider this as a problem of optimization with constraints. The constraints are expressed in the following equation.

$$\sum \theta_{k,2j} = \sum \theta_{k,2j+1} = \pi \tag{1}$$

The first term is the sum of the even angles of k-th inner vertex. The second term is the sum of the odd angles. We express the optimized value E as

$$E = \sum_i w_i |V_i - V_i'|^2 \tag{2}$$

where V_i is a vector that shows the coordinates of i-th vertex, V_i' is a vector that shows the coordinates of the i-th vertex after the position is changed, and w_i is the weight given to i-th vertex. We can say that E is the sum of the weighted squared distances between the positions before and after modification of all vertices. The value of w_i is set to 1 for all inner vertices and an infinite value for vertices placed on edges. By setting these weights, the vertices placed on edges are forced to remain stationary. The equation for optimization with constraints is solved by the method of Lagrange multipliers which is expressed as

$$F(\lambda) = E + \lambda C \tag{3}$$

where constraint C is expressed as

$$C = \pi - \sum \theta_{k,2j} = 0 \tag{4}$$

By solving this problem, we obtain the altered positions of vertices which satisfy condition 1.2 while requiring a minimal amount of modification from Step 3. To solve the optimization problem, we use Newton's method, which is an iterative approach. This approach sometimes fails to converge, in which case we go back to Step 1.

Step 5

Finally, we verify that the pattern is globally flat foldable. To do this we assign a mountain-valley flag to each line to fold the pattern. For a locally flat-foldable pattern, the following necessary condition must be satisfied for mountain-valley assignment [3].

Condition 2 A mountain-valley assignment is flat foldable if and only if, for every strict local minimum angle, the incident creases have opposite mountain-valley assignments and the result of crimping these creases is flat foldable.

Finding valid mountain-valley assignments that can be folded flat by trial-and-error by a human hand is time consuming. Therefore, we assign mountains and valleys which satisfies condition 2 with a computer by a brute force approach. After the flags are set, the pattern is checked to see whether it is globally flat foldable by hand.

3. Result of implementation

We implemented the flow described in the previous section on a PC. We show the results of each step in this section. Generation of vertices and lines in steps 1 and 2 is completely automatic. The generation of vertices is trivial and the generation of lines by using Delaunay triangulation is completed by using an existing geometric library. These steps rarely affect the total computational time. Figure 4(a) shows the result of these steps. We firstly placed vertices at the corners and middle positions of the edges. The number of randomly placed inner vertices is 4.

Step 3 adjusts the number of lines by removing extra edges to satisfy condition 1.1. We did not prove that we can obtain a valid pattern with this approach every time. However, most of the time we did successfully obtain a valid pattern. If we do not generate a valid pattern, we go back to Step 1. Figure 4(b) shows the result of Step 3 applied to Fig. 4(a). Since the leftmost inner vertex in Fig. 4 has an odd number (five) of connected lines, the line connected to the left bottom corner was removed.

Step 4 moves the positions of the vertices so that condition 1.2 is satisfied. Depending on the initial state, the iterative convergence calculation often fails. Even when the calculation successfully converges, intersections between the lines may exist in some cases. In these cases, we go back to Step 1. Figure 4 shows the result of this Step 4 applied to Fig. 4(c). After 74

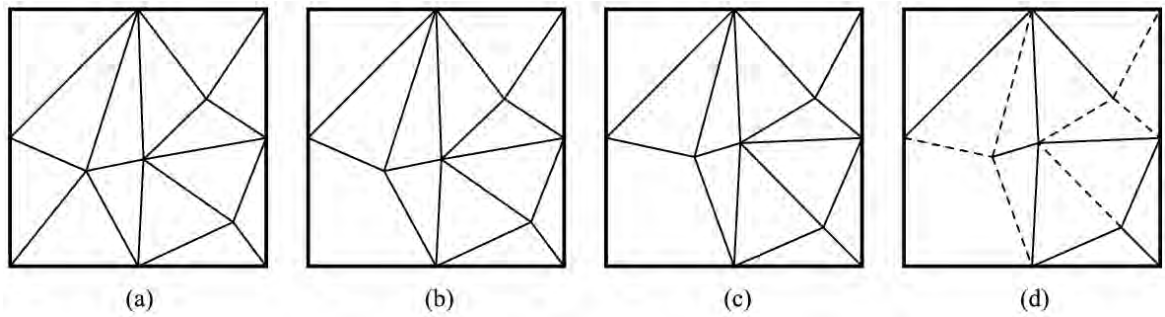

Figure 4: Flow of our method. (a) A square is triangulated with randomly aligned vertices. (b) The number of lines connected to an inner vertex is adjusted to be even. (c) The pattern is modified so that it satisfies condition 1.2. (d) Mountain-valley flags are assigned to lines.

iterations, the positions of vertices converged in less than 1 sec. We set the threshold amount of the error of condition 1.2 to 0.01 degrees.

Step 5 assigns a mountain-valley flag to each line so that the pattern can be flat folded. Although the locally valid assignment finishes in a reasonable time, it is difficult to judge whether the pattern is globally flat foldable. As mentioned in Section 1, no efficient methods exist to determine this. We checked the possibility by folding lines by hand. This is a time-consuming process that often fails. If we cannot globally fold the pattern into a flat model, we assign the mountain-valley flags again. If we still cannot fold the pattern after a certain number of trials, we return to Step 1.

4. Examples

We show some examples generated by our method in Fig. 5. As stated in the previous section, steps 1 and 2 were performed automatically without failure. Although steps 3 and 4 sometimes failed, they are computational operations that are not time consuming. Step 5 is problematic, since verification by human hand is needed, and sometimes this verification process fails. In our experience, as the number of vertices increases, Step 5 rapidly becomes less successful.

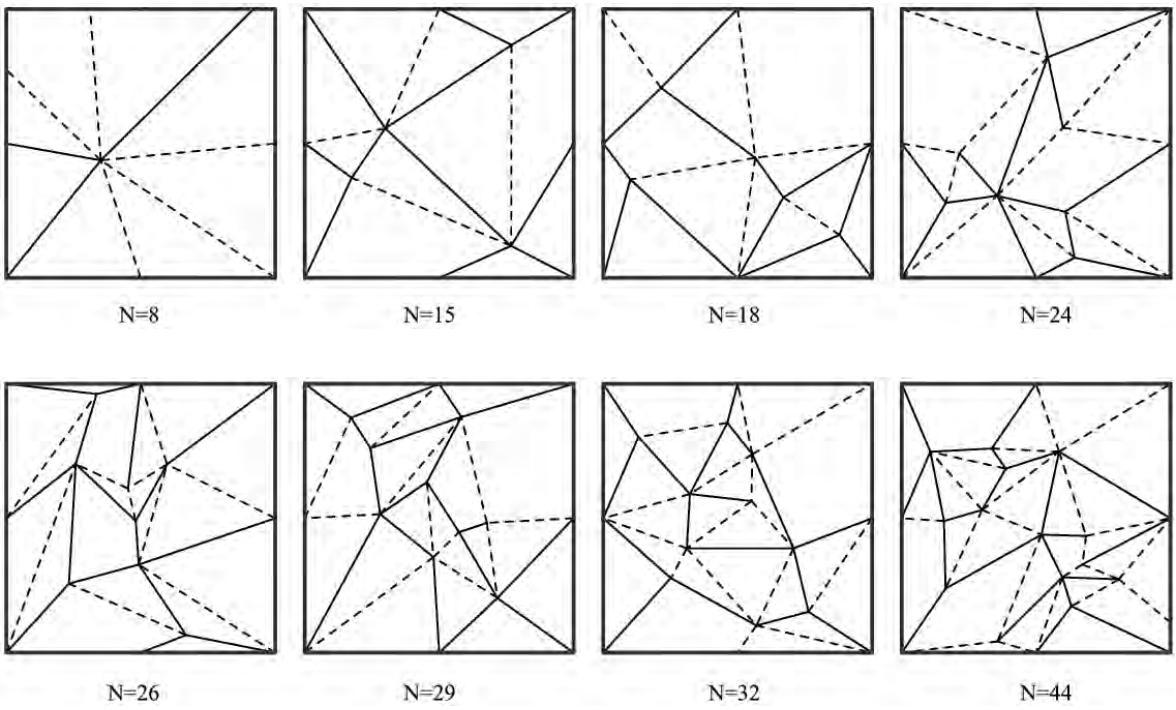

Figure 5: Examples of flat-foldable patterns generated with our method. N is the number of lines included in the crease pattern.

5. Summary

We proposed a method for generating a new crease pattern that is flat foldable. Although some problems still remain, we could generate a variety of flat-foldable patterns with this method. This will help us to find new valuable crease patterns. The open problems our approach has are:

- There is no guarantee that the line-removal in Step 3 will produce a valid connectivity for the crease pattern. Moreover, it is not proved that a valid pattern is always obtained starting from any triangulated square.
- The numerical solution of the optimization problem in Step 4 sometimes fails on non-convergence or no avoidance of intersecting of line segments. We did not attempt to investigate further the nature of the optimization problem. More promising strategies may exist.
- The test for global flat-foldability in the last step is done manually. Since the obtained pattern has numerical errors according to the threshold of convergence, it will be difficult to check the global flat-foldability of the pattern with a simple approach.

Optionally, when mountain-valley flags are hidden in a crease pattern, it becomes the origami puzzle *"Fold Me Up"* introduced in [5]. Our method can also be used to generate problems for this kind of puzzle.

References

[1] P. Wang-Iverson, R.J. Lang, M. Yim: *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*. A K Peters Ltd. 2011.

[2] R.J. Lang: *Origami Design Secrets: Mathematical Methods for an Ancient Art*. A K Peters Ltd. 2003.

[3] E.D. Demaine, J. O'Rourke: *Geometric folding algorithms*. Cambridge University press. 2007.

[4] M. Bern, B. Hayes: *The complexity of flat origami*. In Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms 1996, pp. 175–183.

[5] T. Hull: *Project origami — Activities for Exploring Mathematics*. A K Peters Ltd. 2006.

Received August 7, 2010; final form August 23, 2011