Electronic Preprint for Journal of Information Processing Vol.28

Regular Paper

# Flat Folding a Strip with Parallel or Nonacute Zigzag Creases with Mountain–Valley Assignment

Erik D. Demaine[1,a)] Martin L. Demaine[1,b)] Hiro Ito[2,c)] Chie Nara[3,d)] Izumi Shirahama[4,e)] Tomohiro Tachi[5,f)] Mizuho Tomura[2,g)]

Received: January 6, 2020, Accepted: June 1, 2020

**Abstract:** Deciding flat foldability of a given mountain–valley pattern is known to be NP-complete. One special case known to be solvable in linear time is when the creases are parallel to each other and perpendicular to two sides of a rectangular piece of paper; this case reduces to a purely one-dimensional folding problem. In this paper, we give linear-time algorithms for flat foldability in two more-general special cases: (1) all creases are parallel to each other and to two sides of a parallelogram of paper, but possibly oblique to the other two sides of the parallelogram; and (2) creases form a regular zigzag whose two directions (zig and zag, again possibly oblique to the two sides of the piece of paper) form nonacute angles to each other. In the latter zigzag case, we in fact prove that every crease pattern can be folded flat, even if each crease is specified as mountain, valley, or unfolded.

**Keywords:** origami, crease pattern, flat folding, algorithm

## 1. Introduction

A classic problem in computational origami is *flat foldability*: given a crease pattern drawn on a piece of paper, can the paper be folded flat (into the plane) so as to have creases (folds by 180°) exactly as specified by the crease pattern? In the variant considered here, we are also given a *mountain–valley assignment*, that is, a specification of whether each crease should be folded in one direction (+180° or mountain) or the other (−180° or valley); together, the crease pattern and mountain–valley assignment constitute a *mountain–valley pattern*.

Both versions of the flat foldability problem — given a mountain–valley pattern or just a crease pattern — are known to be NP-complete, even when the crease pattern consists of horizontal, vertical, and diagonal creases on a rectangle of paper [2]. However, in the special case where the paper is a rectangle (intuitively, a long narrow strip) and all creases are perpendicular to two of the sides, the piece of paper effectively becomes a one-dimensional segment and flat foldability becomes tractable. Indeed, every crease pattern is flat foldable with the alternating mountain–valley assignment, and flat foldability of a specific mountain–valley pattern can be decided in linear time [3], [5]; see Section 2. We call this problem *one-dimensional flat foldability with mountain–valley assignment* (1DFF-MV for short).

In this paper, we solve two more-general special cases of flat foldability of mountain–valley patterns on a parallelogram or trapezoid of paper (intuitively, a long narrow strip). In the first case, the creases are all parallel to each other and to two sides of the parallelogram paper, forming an arbitrary angle (not necessarily 0 or 90°) to the other two sides of the paper. We show that this problem can be solved in linear time, by extending the 1DFF-MV algorithm of Refs. [3], [5]. In the second case, the creases form a zigzag pattern (with bends at edges of the strip) and there are exactly two alternating directions of creases (zig and zag) that form a fixed nonacute ($\geq 90°$) angle to each other. In this problem, we also allow some creases of the zigzag to be omitted, or equivalently, to be assigned "unfolded" instead of "mountain" or "valley". We show that such mountain–valley–unfolded patterns are always flat foldable, and thus the flat foldability problem is easy to solve (return "yes").

### 1.1 Preliminaries

A *piece of paper* can in general be any region (e.g., a polygon) [5], we focus here on the case of a parallelogram, which we assume for simplicity has the long direction aligned with the $x$-axis. A *crease* is a line segment on the piece of paper. A *crease pattern* is a set of creases that meet only at shared endpoints, that is, a planar straight-line graph on the piece of paper. A *mountain–valley assignment* is an assignment of "mountain" or "valley" to each crease in a crease pattern; a *mountain–valley–unfolded assignment* allows a third "unfolded" option. A *mountain–valley (–unfolded) pattern* is a crease pattern together with a mountain–

1 Massachusetts Institute of Technology, Cambridge, MA, USA
2 The University of Electro-Communications, Chofu, Tokyo 182–8585, Japan
3 Meiji University, Nakano, Tokyo 164–8525, Japan
4 ECBEING CORP., Shibuya, Tokyo 150–0002, Japan
5 The University of Tokyo, Meguro, Tokyo 153–8902, Japan
a) edemaine@mit.edu
b) mdemaine@mit.edu
c) itohiro@uec.ac.jp
d) cnara@jeans.ocn.ne.jp
e) izmrhythm@gmail.com
f) tachi@idea.c.u-tokyo.ac.jp
g) mizuho.tomura1012@gmail.com

valley(–unfolded) assignment.

A *flat folding* or *flat folded state* of a piece of paper consists of a *geometry function* mapping each point of paper to a point in 2D and an *ordering function* specifying, for each pair of noncrease points mapping to the same 2D point, which point is on top [5]. The geometry function must be *isometric* meaning that paths have the same length before and after mapping; the ordering function must avoid *crossings*. A *crease point* is a point where the geometry function is not locally flat; the set of such points is the crease pattern of the flat folding. Each crease of a flat folding can be distinguished as *mountain* or *valley* according to whether the top sides or bottom sides, respectively, of the paper come together locally at that crease (as determined by the ordering function).

A *flat folding of a given crease pattern* is a flat folding whose associated crease pattern is exactly the given crease pattern, i.e., a folding that is creased exactly where specified. In this case, the crease pattern is called *flat foldable*. The geometry function of the flat folding is easily determined by the crease pattern by reflecting at each crease; the difficult part is finding a valid ordering function. Similarly, we define flat foldings of a given mountain–valley pattern.

In figures, we illustrate a mountain crease by a dot–dashed line, and a valley crease by a dashed line, following [5], [6]. When there is no possibility of misunderstanding or when it is not necessary to distinguish, we sometimes draw a crease as a solid line.

### 1.2 Related Work

Origami mathematical research has been done since the 1960s [9], [10]; see Ref. [5]. In recent years, the problem of flattening solids has been considered for engineering applications [11]. A recent survey paper on flat folding is Ref. [8]. The two classic flat foldability problems are as follows:

**Problem:** Flat folding (FF)
**Input:** Crease pattern
**Question:** Is the specified crease pattern flat foldable?

**Problem:** Flat folding with mountain–valley assignment (FF-MV)
**Input:** Mountain–valley pattern
**Question:** Is the specified mountain–valley pattern flat foldable?

One of the first studied variations on these problems is *local* flat foldability, which is about flat foldability of crease patterns with a single vertex and $n$ creases emanating from that vertex. Kawasaki's Theorem [5], [6], [8], [9], [10] characterizes which single-vertex crease patterns are flat foldable. Single-vertex mountain-valley patterns are more difficult, but can also be characterized with a linear-time algorithm. Maekawa's Theorem [5], [8], [11] gives a necessary condition, while a complete algorithm was first given by Hull [5], [7].

Deciding flat foldability of general crease patterns or mountain–valley patterns is strongly NP-complete [2] (This result was originally claimed in 1996 [4], with a flaw and correction twenty years later [2]). For 1DFF-MV, where all creases are vertical in a horizontal rectangular strip of paper, a linear-time algorithm was first given by Arkin et al. [3], [5]. The problems solved in this paper are generalizations of this problem.

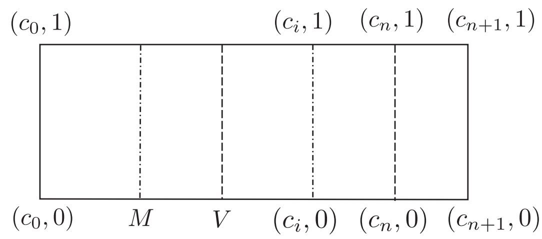



**Fig. 1** An instance of 1DFF-MV.

One interesting but still open special case is *map folding*, where the crease pattern is an $n_1 \times n_2$ grid of squares. It is unknown whether there is a polynomial-time algorithm to decide the flat foldability of a given mountain–valley assignment to these creases [3], [5]. On the other hand, if we restrict the flat folding to "simple foldings", there are efficient algorithms [1], [3].

## 2. Previous Algorithm for 1DFF-MV

Our problems generalize one-dimensional flat foldability with mountain–valley assignment (1DFF-MV), and our algorithms use the linear-time algorithm for 1DFF-MV as a subroutine [3]. We thus begin with a description of this algorithm.

Consider a 1D crease pattern specified by $n + 2$ real values $c_0$, $c_1, \ldots, c_{n+1}$ (where $c_0 < c_1 < \cdots < c_{n+1}$), and a mountain–valley assignment given by a function $L : \{1, 2, \ldots, n\} \to \{M, V\}$. The piece of paper is a rectangular strip whose 2D vertex coordinates are $(c_0, 0)$, $(c_{n+1}, 0)$, $(c_{n+1}, 1)$, and $(c_0, 1)$; see **Fig. 1**. The $y$-extent (strip width) is 1, and the $x$-extent is $c_{n+1} - c_0$; Crease $i$ has endpoints at coordinates $(c_i, 0)$ and $(c_i, 1)$. If $L(i) = M$, then $c_i$ is a mountain crease, and if $L(i) = V$, then $c_i$ is a valley crease. An instance can thus be expressed by a tuple $I = (c_0, \ldots, c_{n+1}; L)$.

1DFF-MV can be formulated as follows:

**Problem:** One-dimensional flat foldability with mountain–valley assignment (1DFF-MV)
**Input:** $I = (c_0, \ldots, c_{n+1}; L)$
**Question:** Is the specified mountain–valley pattern flat foldable?

In this problem, the width of the paper strip does not affect the result, so the essence of the problem does not change even if the strip is considered a line segment. This is why this problem is called "one-dimensional".

To solve this problem, we define two kinds of operations, "crimps" and "end-folds", and a property "mingling", as follows.

### 2.1 Crimps

If two consecutive creases $c_i$ and $c_{i+1}$ are assigned different mountain/valley labels ($\{L(c_i), L(c_{i+1}\} = \{M, V\}$), and the following inequalities hold, then they can be folded as shown in **Fig. 2**.

$$|c_{i-1} - c_i| \geq |c_i - c_{i+1}| \leq |c_{i+1} - c_{i+2}|. \qquad (1)$$

This operation is called a *crimp*. The pair $(c_i, c_{i+1})$ of creases is

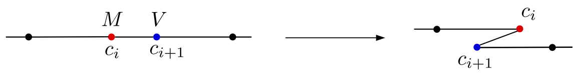



**Fig. 2** Crimp operation.

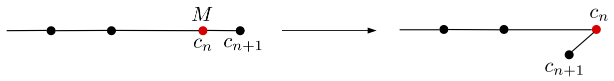



**Fig. 3** End-fold operation.

called *crimpable*. Crimping $(c_i, c_{i+1})$ removes these two creases disappear and decreases the indices of creases after $c_{i+1}$ (and the number of creases) by 2.

### 2.2 End-Folds

If the last crease satisfies the following inequality, then it can be folded as shown in **Fig. 3**.

$$|c_{n-1} - c_n| \geq |c_n - c_{n+1}| \tag{2}$$

This operation is called an *end-fold*, and $c_n$ is called a *foldable end*.

Similarly, the first crease $c_1$ is a foldable end if

$$|c_1 - c_2| \geq |c_1 - c_0|. \tag{3}$$

End-folding $c_1$ or $c_n$ removes that crease and decreases the number of creases by 1; in the latter case, it also decreases the indices of the remaining creases by 1.

### 2.3 Mingling Property

A 1D crease pattern is *mingling* if every maximal sequence $c_i, c_{i+1}, \ldots, c_j$ (where $1 \leq i < j \leq n$) of two or more consecutive creases with the same label $L(c_i) = L(c_{i+1}) = \cdots = L_{c_j}$ satisfies at least one of the following inequalities:

$$|c_{i-1} - c_i| \leq |c_i - c_{i+1}| \quad \text{or} \quad |c_{j-1} - c_j| \geq |c_j - c_{j+1}|. \tag{4}$$

Arkin et al. [3] solved 1DFF-MV by proving the following properties of mingling:

**Lemma 1** (Ref. [3]). *Every flat-foldable one-dimensional mountain–valley pattern is mingling.*

**Lemma 2** (Ref. [3]). *Any mingling one-dimensional mountain–valley pattern has either a crimpable pair or a foldable end.*

**Lemma 3** (Ref. [3]). *Folding a foldable end and crimping a crimpable pair preserve flat foldability.*

Combining Lemmas 1–3, we obtain the following lemma:

**Lemma 4** (Ref. [3]). *Any flat-foldable one-dimensional mountain–valley pattern can be folded flat by a sequence of crimps and end-folds.*

### 2.4 Algorithm

Arkin et al. showed the following theorem using Lemma 4.

**Theorem 1** (Ref. [3]). *1DFF-MV can be solved in $O(n)$ time.*

We now detail Theorem 1's algorithm for 1DFF-MV, which we call *CRIMP*, for determining whether a given instance $I = (c_0, \ldots, c_{n+1}; L)$ is flat foldable. In this algorithm, we maintain a list $\mathcal{F}$ of foldable creases, containing all crimpable pairs $(c_i, c_j)$ and all end-foldable creases [*1].

```
procedure CRIMP(I)
begin
    initialize F
    while F is not empty do
        choose an element of F and crimp or end-fold it
        update F
    end while
    if all creases were folded then then
        output "yes"
    else
        output "no"
    end if
end
```

It takes constant time to check whether a pair of consecutive creases are crimpable or whether an end crease is foldable. Thus we can initialize $\mathcal{F}$ in linear time. When we apply a crimp or an end-fold, only a constant number of pairs of consecutive creases or end creases can change their crimpable or end-foldable status. Thus we can update $\mathcal{F}$ in constant time per fold. Each operation decreases the number of creases by at least 1, so the number of iterations is at most $n$, for a total of $O(n)$ time.

## 3. Strip Flat Folding with Parallel Creases

In this section, we solve our first special case of flat foldability, in which all creases are parallel, but their angle to two parallel sides of a sheet of paper may not be 90°. In this case, we consider the piece of paper as a parallelogram whose ends are parallel to creases instead of a rectangle [*2].

### 3.1 Problem

Such a mountain–valley pattern with $n$ creases can be specified by $n + 3$ real values $\theta$, $c_0$, $c_1$, $\ldots$, $c_{n+1}$ (where $0 < \theta \leq 90°$ and $c_0 < c_1 < \cdots < c_{n+1}$) and a mountain–valley assignment $L : \{1, 2, \ldots, n\} \to \{M, V\}$ without loss of generality. The piece of paper is a parallelogram strip whose 2D vertex coordinates are $(c_0, 0)$, $(c_{n+1}, 0)$, $(c_{n+1} + \cot\theta, 1)$, and $(c_0 + \cot\theta, 1)$; see **Fig. 4**. The $y$-extent (strip width) is 1. Crease $i$ has endpoints at coordinates $(c_i, 0)$ and $(c_i + \cot\theta, 1)$. If $L(i) = M$, then $c_i$ is a mountain crease, and if $L(i) = V$, then $c_i$ is a valley crease.

This mountain–valley pattern has $n$ creases, and the angle formed by each crease and the long (horizontal) edge of the strip is $\theta$. The short (nonhorizontal) edges of the strip and all of the creases are parallel.

An instance can be expressed by a tuple $I = I(\theta) = (c_0, \ldots, c_{n+1}; L, \theta)$. We use the notation $I(\theta)$ to make it easy to consider alternate angles $\theta'$.

The problem can be formulated as follows:

[*1] Note that separated creases $c_i$ and $c_j$ with $j \geq i + 2$ may later become a crimpable pair when all creases between them have been folded.

[*2] If we were to use rectangular paper, we could handle the irregularity at the two ends in a constant time, without affecting the efficiency of the algorithm.

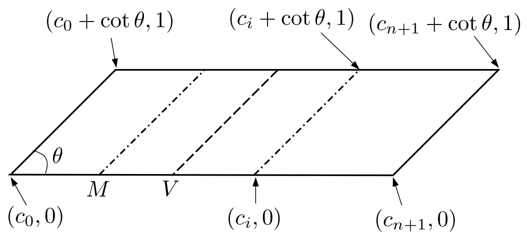



**Fig. 4** An instance of StripFFP-MV.

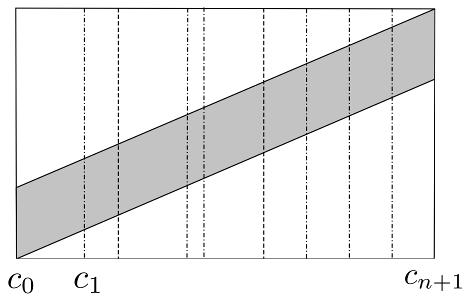



**Fig. 5** Instance $I$ and the universal sheet of $I$.

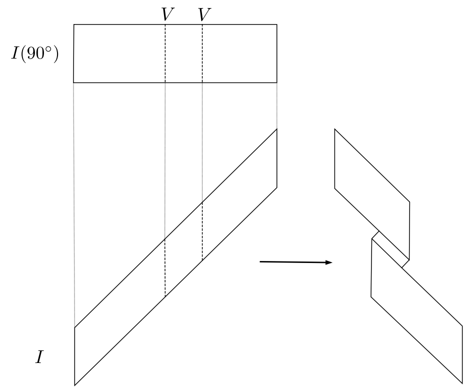



**Fig. 6** An example where $I$ is flat foldable but $I(90°)$ is not flat foldable.

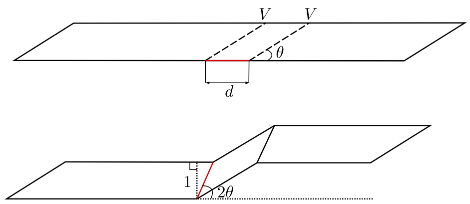



**Fig. 7** A case where the distance $d$ between creases is $1/\sin 2\theta$.

**Problem:** Strip flat folding problem with parallel creases and mountain–valley assignment (StripFFP-MV)
**Input:** $I = (c_0, \ldots, c_{n+1}; L, \theta)$
**Question:** Is the specified mountain–valley pattern flat foldable?

### 3.2 Universal Sheet

For an instance $I = (c_0, \ldots, c_{n+1}; L, \theta)$, we define the *universal sheet* of $I$ to be the rectangle that contains $I$'s parallelogram piece of paper and has two sides parallel to the creases; see **Fig. 5**. This universal sheet together with the creases of $I$ can be regarded as an instance of 1DFF-MV. We use this concept to prove the following property:

**Lemma 5.** *For any input $I$ of StripFFP-MV, $I$ is flat foldable if $I(90°) = (c_0, \ldots, c_{n+1}; L, 90°)$ is flat foldable.*

*Proof.* The intervals between the creases on the universal sheet of $I$ is $\sin\theta$ times the intervals of $I(90°)$. The 1DFF-MV problem has the same solution even if intervals of creases are enlarged by the same magnification, so universal sheet of $I$ is equivalent to $I(90°)$. Therefore, if $I(90°)$ is flat foldable, the universal sheet of $I$ is flat foldable. The universal sheet of $I$ contains $I$, so if the universal sheet of $I$ is flat foldable, then $I$ is also flat foldable. □

By Lemma 5, the CRIMP algorithm for 1DFF-MV gives a partial answer to StripFFP-MV. Given an instance $I(90°)$ of 1DFF-MV, call CRIMP to determine whether it is flat foldable. If the output is "yes", then $I$ is also flat foldable, and we have solved StripFFP-MV. However, if CRIMP's output is "no", then we do not know about StripFFP-MV. The reverse of Lemma 5 is generally not true: **Fig. 6** is an example where $I$ is flat foldable but $I(90°)$ is not flat foldable.

### 3.3 Separating Big Gaps

Intuitively, the part that collides in $I(90°)$ may avoid collision by moving diagonally in $I(\theta)$. Let $d$ be the distance between two adjacent creases. As can be seen in the example of **Fig. 7**, when $d$ is more than $1/\sin 2\theta$, no collision occurs. The following lemma gives theoretical support for this argument:

**Lemma 6.** *Let $I = (c_0, \ldots, c_{n+1}; L, \theta)$ be an instance of StripFFP-MV. Suppose that $k \in \{1, \ldots, n-1\}$ satisfies the following inequality:*

$$c_{k+1} - c_k \geq \frac{1}{\sin 2\theta}. \tag{5}$$

*Then $I$ is flat foldable if and only if two instances of StripFFP-MV, $I_1 = (c_0, \ldots, c_{k+1}; L, \theta)$ and $I_2 = (c_k, \ldots, c_{n+1}; L, \theta)$, are both flat foldable.*

*Proof.* If $I$ is flat foldable, then so are $I_1$ and $I_2$: any flat folding can be restricted to a subset of the paper. Now assume that $I_1$ and $I_2$ are flat foldable. From the inequality $c_{k+1} - c_k \geq 1/\sin 2\theta$, when $c_k$ and $c_{k+1}$ are both folded, the strip corresponding to the right side of $c_{k+1}$ (referred to as the right strip) is located above the strip corresponding to the left side of $c_k$ (referred to as the left strip); see Fig. 7. Even if it is folded afterwards, no part of the right strip will come below, and no part of the left strip will come above, so they do not interfere. Therefore, a flat folding of $I$ can be obtained by joining flat foldings of $I_1$ and $I_2$ by identifying the parallelogram between $c_k$ and $c_{k+1}$. □

From this lemma, we define the *separate* operation. Given an instance $I$ and a value $k$ satisfying the condition of Lemma 6, SEPARATE$(I, k)$ outputs $I_1$ and $I_2$ as defined in Lemma 6; see **Fig. 8**.

The basic strategy of our algorithm for solving StripFFP-MV is to apply SEPARATE or CRIMP recursively whenever they can be applied. If the algorithm stops before all creases are folded, we will show that the instance is not flat foldable.

Before we proceed, we need to generalize the shapes handled by Lemma 6. This is because, when we apply CRIMP, the shape of the instance will no longer be a parallelogram; see **Fig. 9**. In this case, SEPARATE can be applied under a condition different

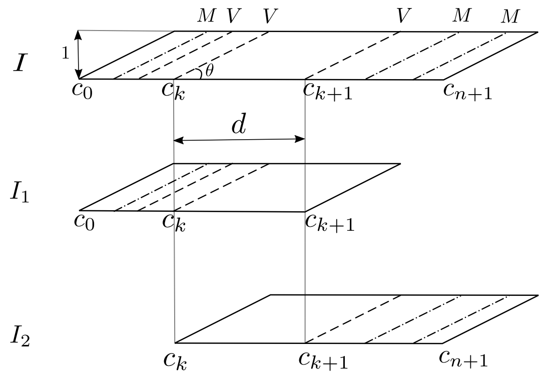



**Fig. 8** Separating instance $I$ into $I_1$ and $I_2$.

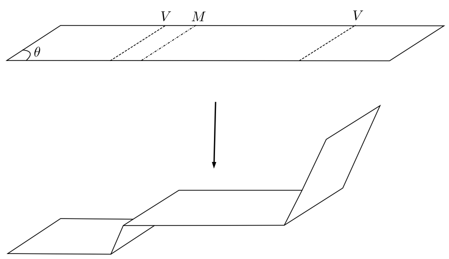



**Fig. 9** Shape change of an instance.

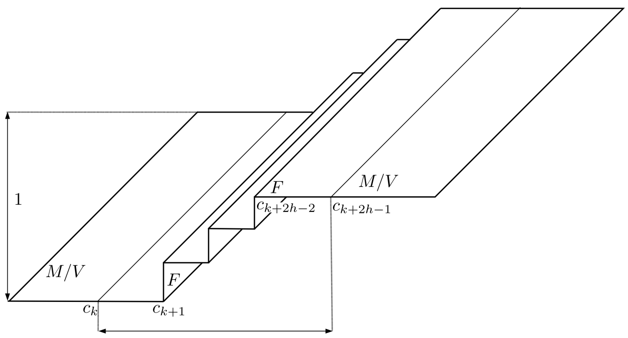



**Fig. 10** A separable pair of creases such that all creases between them are folded.

from (5). In order to treat this case, we must determine what shape the strip has in each step of the algorithm.

Define an *edge* $(c_k, c_{k+1})$ to be the segment between the lower endpoints $(c_k, 0), (c_{k+1}, 0)$ of two consecutive creases $c_k, c_{k+1}$. After crease $c_k$ is folded, the angle between edge $(c_k, c_{k+1})$ and the $x$-axis becomes $2\theta$.

We change the label of folded creases to $F$, which means "folded". In 1DFF-MV, folded creases effectively disappear, but in this problem, we need to keep track of folded creases because they change the shape. Thus we obtain three types of crease labels: $\{M, V, F\}$. Hereafter, we allow "an instance $I$" to include an intermediate state of the algorithm, where the function $L$ can output $F$ in addition to $M$ and $V$.

Based on the above discussion, we obtain the following extension of Lemma 6; see **Fig. 10**.

**Lemma 7.** *Let $I = (c_0, \ldots, c_{n+1}; L, \theta)$ be an instance of StripFFP-MV. Suppose there are integers $1 \leq k \leq n-1$ and $h \geq 1$ that satisfy the following conditions:*

$$\left.\begin{array}{l} L(k) \neq F, \\ L(k+2h-1) \neq F, \\ L(i) = F \quad (\forall i \in \{k+1, \ldots, k+2h-2\}), \\ (c_{k+1} - c_k) + (c_{k+3} - c_{k+2}) + \cdots + (c_{k+2h-1} - c_{k+2h-2}) \geq \dfrac{1}{\sin 2\theta} \end{array}\right\} \quad (6)$$

*Then $I$ is flat foldable if and only if two instances of StripFFP-MV, $I_1 = (c_0, \ldots, c_{k+2h-1}; L, \theta)$ and $I_2 = (c_k, \ldots, c_{n+1}; L, \theta)$, are both flat foldable.*

*Proof.* The proof is a simple extension of the proof of Lemma 6. □

We define SEPARATE$(I, k, h) = (I_1, I_2)$ according to the constraints and definitions of Lemma 7. When integers $1 \leq k \leq n-1$ and $h \geq 1$ satisfy the conditions (6), $(c_k, c_{k+2h-1})$ is called a *separable pair*.

### 3.4 Algorithm

The basic idea of the algorithm is to apply CRIMP or SEPARATE as much as possible, and to determine that the pattern is not flat foldable if they cannot be applied before every crease is folded. If this idea is implemented directly, the computation time is $\Theta(n^2)$. We show how to speed up this algorithm to run in linear time. The method is to apply CRIMP first as far as possible and then to apply SEPARATE. The validity of this method is supported by the following lemmas.

**Lemma 8.** *When instance $I = (c_0, \ldots, c_{n+1}; L, \theta)$ of StripFFP-MV is not crimpable anywhere, separating does not make anything crimpable.*

*Proof.* Let the instances obtained by separating $I$ by SEPARATE$(I, k, h)$ be $I_1 = (c_0, \ldots, c_{k+2h-1}; L, \theta)$ and $I_2 = (c_k, \ldots, c_{n+1}; L, \theta)$. By symmetry, suppose that $I_1$ is crimpable at a pair of creases $c_i$ and $c_{i+1}$. Because the $L$ labels of $c_{k+1}, \ldots, c_{k+2h-2}$ are all $F$, $i$ must be in $\{1, \ldots, k-1\}$. By crimpability, we know that the crimp condition (1) holds. However, this condition also holds for $I$, and the labels of $c_i$ and $c_{i+1}$ do not change, so $c_i$ and $c_{i+1}$ are crimpable in $I$, a contradiction. □

**Lemma 9.** *When instance $I = (c_0, \ldots, c_{n+1}; L, \theta)$ of StripFFP-MV is not separable anywhere, end-folding does not make it separable.*

*Proof.* Let the instance obtained by end-folding $I$ be $I' = (c_0, \ldots, c_n; L, \theta)$. Suppose that $I'$ is separable at a pair of creases $c_k$ and $c_{k+2h-1}$. By Lemma 7, the conditions (6) hold for $I'$. For all $i \in \{k+1, \ldots, k+2h-2\}$, the label $L(i)$ is not changed by end-folding because end-folding only changes the label that is closest to the end of the strip (to $F$). Thus, if there is a section whose label is $F$ between two non-$F$ creases, it was $F$ before end-folding. Therefore, $(c_k, c_{k+2h-1})$ is also a separable pair in $I$. □

By calling SEPARATE after crimping all the crimpable pairs from Lemma 8, we obtain the DIAGONAL_CRIMP algorithm for StripFFP-MV. In it, we maintain a list of separable pairs $\mathcal{S}$ containing all separable pairs $(c_k, c_{k+2h-1})$.

We explain the above procedure further as follows. First, when CRIMP is applied to $I$ in Line 2, it is treated as its universal sheet. If the output of this call is "yes", then this algorithm outputs "yes" and stops. If the output of this call is "no", then we separate $I$ at all separable pairs (Line 12). If there is not a separable pair, this algorithm outputs "no" and stops. Otherwise, we apply CRIMP for all separated subinstances (Line 14). At this point, all crimpable pairs were already crimped by CRIMP in Line 2, so by

**procedure** DIAGONAL_CRIMP($I$)

```
 1: begin
 2:     call CRIMP(I), folding I as much as possible
 3:     if the output of the above call is "yes" then
 4:         output "yes"
 5:         stop
 6:     end if
 7:     Make a list S of separable pairs
 8:     if S is empty then
 9:         output "no"
10:         stop
11:     end if
12:     Separate I for all the elements (separable pairs) of S, and let the ob-
        tained subproblems be I_1, ..., I_k (where k = |S| + 1)
13:     for each subinstance I_l (1 ≤ l ≤ k) do
14:         call CRIMP(I_l)
15:         if the output of the above call is "no" then
16:             output "no"
17:             stop
18:         end if
19:     end for
20:     output "yes"
21: end
```

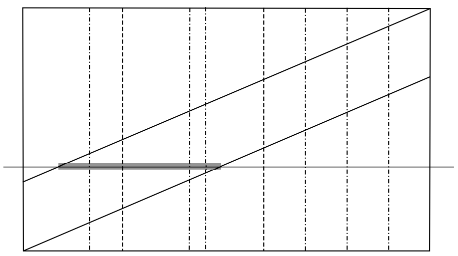



**Fig. 11** An essential subinstance.

Lemma 8, only end-folding is performed by CRIMP in Line 14.

### 3.5 Correctness

This section proves that DIAGONAL_CRIMP satisfies both soundness (non-flat-foldable instances are always rejected) and completeness (flat-foldable instances are always accepted).

Soundness can be proved easily as follows:

**Lemma 10.** *DIAGONAL_CRIMP satisfies soundness.*

*Proof.* Instances that are determined to be flat foldable by the procedure DIAGONAL_CRIMP are flat foldable by Lemma 7 and the definition of CRIMP. □

Next, we show completeness as follows. We define some terms and give a lemma that are used for the proof.

Consider the universal sheet of an instance $I$ of StripFFP-MV. On the universal sheet, consider an arbitrary horizontal line that intersects the creases of $I$ at a right angle; see **Fig. 11**. The line segment where this horizontal line intersects with $I$ can be regarded as an instance of 1DFF-MV in combination with creases that intersect it. These instances are called *essential subinstances* of $I$. Note that there are infinite number of essential subinstances for a single $I$.

**Lemma 11.** *If there is an essential subinstance of I that is not flat foldable, then I is not flat foldable.*

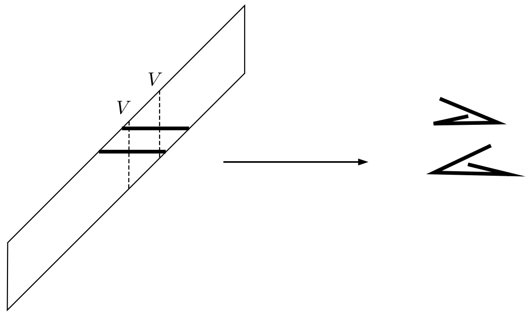



**Fig. 12** An example where the strip is not flat foldable even though all essential subinstances are flat foldable.

*Proof.* Any flat folding of $I$ can be restricted to any essential subinstance of $I$. □

However, the reverse is not generally true: **Fig. 12** shows an example. Although this instance is not flat foldable, all essential subinstances are flat foldable by performing end-folds at most twice as shown in the figure. However, because the vertical relationship between the left and right papers is different, they cannot be folded without contradiction. DIAGONAL_CRIMP correctly determines the not-flat-foldability of this instance, because the universal sheet of this instance cannot be crimped, end-folded, nor separated anywhere, and it is rejected at Line 9.

The proof of completeness can now be shown as follows:

**Lemma 12.** *DIAGONAL_CRIMP satisfies completeness.*

*Proof.* First, (i) if there is an essential subinstance that is not flat foldable, then the instance is not flat foldable by Lemma 11. Thus, in the following, we assume that (ii) all essential subinstances are flat foldable.

Assume that a flat-foldable instance is determined by DIAGONAL_CRIMP to be not flat foldable. This is determined at Lines 9 or 16. In this case, there is a subinstance that cannot be further separated, crimped, nor end-folded. This means that this subinstance is not mingling by Lemma 2. Therefore, there is a maximal sequence $c_p, c_{p+1}, \ldots, c_{q-1}, c_q$ of consecutive creases with the same labels that satisfy both the following inequalities:

$$|c_{p-1} - c_p| > |c_p - c_{p+1}|, \tag{7}$$

$$|c_{q-1} - c_q| < |c_q - c_{q+1}|. \tag{8}$$

If $c_{p-1}, c_p, \ldots, c_{q+1}$ are included in one essential subinstance, then the essential subinstance is not mingling and must be considered in (i). Thus we assume that $c_{p-1}, c_p, \ldots, c_{q+1}$ are not included in one essential subinstance. From the inequalities (7) and (8), there is a sequence $c_{p'}, \ldots, c_{q'}$ ($p \le p', q' \le q$) of consecutive creases with the same label that satisfy

$$\begin{aligned} &|c_{p'-1} - c_{p'}| \\ &> |c_{p'} - c_{p'+1}| = \cdots = |c_r - c_{r+1}| = \cdots = |c_{q'-1} - c_{q'}| \\ &< |c_{q'}\ c_{q'+1}|. \end{aligned}$$

Because it cannot be further separated, there is the essential subinstance starting from $c_{p'-1}$ and including $c_{p'}$ and $c_{p'+1}$ (from the condition, it also includes a part of edge $(c_{p'+1}, c_{p'+2})$). By the assumption that it is flat foldable and by Lemma 11, every essential subinstance is flat foldable. From $|c_{p'-1} - c_{p'}| > |c_{p'} - c_{p'+1}|$, $c_{p'-1}$, $c_{p'}$, and $c_{p'+1}$ form a spiral structure with $(c_{p'-1}, c_{p'})$ on

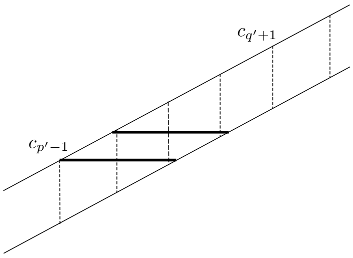



**Fig. 13** The essential subinstances that have a spiral structure with the left side outside.

the outermost side. Next, consider an essential subinstance including three creases $c_{p'}$, $c_{p'+1}$, and $c_{p'+2}$ and a part of edges $(c_{p'-1}, c_{p'})$ and $(c_{p'+2}, c_{p'+3})$. From the consistency of the former essential subinstance, this also must form a spiral structure with $(c_{p'-1}, c_{p'})$ on the outermost side. **Figure 13** is an example showing these two essential subinstances. Inductively, for any $r \in \{p'-1, \dots, q'-1\}$, $(c_r, c_{r+1})$ is located outside of the spiral structure rather than $(c_{r+1}, c_{r+2})$.

By applying the same argument from the right side, for any $r \in \{p'-1, \dots, q'-1\}$, $(c_r, c_{r+1})$ is located inside the spiral structure rather than $(c_{r+1}, c_{r+2})$. This is a contradiction. Therefore, this instance is not flat foldable.

By the above argument, the instance is not flat foldable in both cases (i) and (ii). Therefore, if $I$ is determined to be not flat foldable by DIAGONAL_CRIMP, then $I$ is not flat foldable, so DIAGONAL_CRIMP satisfies completeness. □

The correctness of DIAGONAL_CRIMP now follows from Lemmas 10 and 12:

**Lemma 13.** *The procedure DIAGONAL_CRIMP correctly solves StripFFP-MV.*

### 3.6 Running Time

Finally, we show that DIAGONAL_CRIMP runs efficiently:

**Theorem 2.** *StripFFP-MV can be solved in $O(n)$ time by DIAGONAL_CRIMP.*

*Proof.* By Lemma 13, the procedure DIAGONAL_CRIMP correctly solves StripFFP-MV, so it only remains to bound the running time. First, CRIMP runs in linear time by Theorem 1, bounding the time for Step 2. Each crease is included in at most two subinstances made in Line 12 in DIAGONAL_CRIMP. Thus the total time in all calls to CRIMP in Line 14 of DIAGONAL_CRIMP is also linear. Computing the list $\mathcal{S}$ of separable pairs in Line 7 can be done in linear time by checking each pair of consecutive non-F creases for separability. Therefore, DIAGONAL_CRIMP can be performed in $O(n)$ time. □

## 4. Strip Flat Folding with Nonacute Zigzag Creases

In this section, we solve our second special case of flat foldability, in which the crease pattern consists of an alternating sequence of two parallel families of creases, where the angle between consecutive creases is nonacute ($\geq 90°$). In this problem, we allow a label of a crease "unfold" ($U$) in addition to mountain ($M$) and valley ($V$). If the label of a crease is $U$, the crease is not to be folded; in other words, it is omitted from the crease pattern, and just serves as a reference line segment.

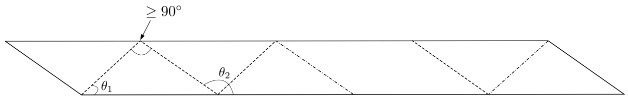



**Fig. 14** An instance of StripFF-zz-MVU($\geq 90°$).

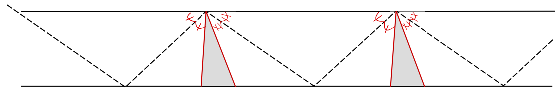


**Fig. 15** Reflection lines.

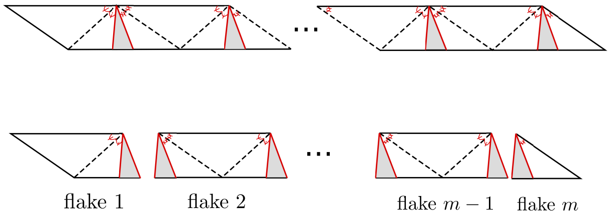



**Fig. 16** Separating into flakes (labeled) and shared ribs (shaded grey).

### 4.1 Problem

Such a mountain–valley–unfolded pattern can be specified by an integer $n$, which is the number of creases, two angles $\theta_1, \theta_2$ satisfying $0 < \theta_1 < \theta_2 < 180°$, and a mountain–valley–unfolded assignment $L : \{1, \dots, n\} \to \{M, V, U\}$. The piece of paper is a parallelogram strip if $n$ is odd, and a trapezoid if $n$ is even. For both shapes, the $y$-extent is 1. The creases form the zigzag pattern shown in **Fig. 14**. The angle that odd creases make with the bottom long edge (base) of the strip is $\theta_1$, while the angle that even creases make with the base is $\theta_2 = 180° - \theta_1$. The angle between consecutive creases is $\theta_2 - \theta_1$, so for the nonacuteness property, we require

$$\theta_2 - \theta_1 \geq 90°. \tag{9}$$

An instance can be expressed by a tuple $I = (n, L, \theta_1, \theta_2)$. The problem can then be formulated as follows:

> **Problem:** Strip flat folding problem with nonacute zigzag creases and mountain–valley–unfolded assignment (StripFF-zz-MVU($\geq 90°$))
> **Input:** $I = (n, L, \theta_1, \theta_2)$
> **Question:** Is the specified mountain–valley pattern flat foldable?

We prove that the answer to this question is always "yes":

**Theorem 3.** *Every instance of StripFF-zz-MVU($\geq 90°$) is flat foldable.*

*Proof.* Consider the reflection lines drawn in red in **Fig. 15**, which result from reflecting the top edge of the strip across the incident creases. As shown in **Fig. 16**, these lines separate the instance into $m = \lceil n/2 \rceil + 1$ strips, called *flakes* [*3], each containing two creases and four reflection lines. Each triangular part sandwiched between two adjacent reflection lines, called a *rib*, is shared by (the overlap of) two consecutive flakes. (If the angle

[*3] Imagine flaking a fish from its ribcage.

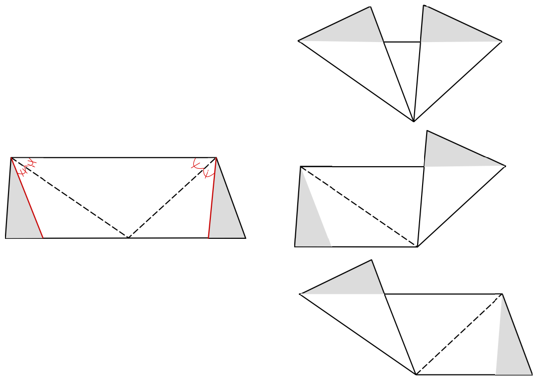


**Fig. 17** A flake and its possible flat foldings.

$\theta_2 - \theta_1$ between two creases is exactly 90°, each rib degenerates to a zero-area line segment.) We number the flakes $1, 2, \ldots, m$ from left to right.

Because the angle $\theta_2 - \theta_1$ between the two creases is $\geq 90°$ by (9), each of the flakes is flat foldable regardless of the mountain–valley–unfolded assignment. **Figure 17** shows the possible foldings of a flake depending on which creases are (un)folded. After such folding, in each flake, each rib does not overlap any other part of the paper and is exposed both above and below.

Now consider joining the folded flakes together: put each folded flake $i$ in its own layer $i$, translate these foldings to overlap corresponding ribs, and then fuse together corresponding ribs. Because each flake $i$ has both ribs exposed both above and below, one rib can be fused to layer $i + 1$ and the other rib can be fused to layer $i - 1$ without causing intersection. Therefore, the result is a flat folding of the entire mountain–valley–unfolded pattern. □

## 5. Conclusion

We solved two new strip-folding problems. One is the strip flat folding problem with parallel creases and mountain–valley assignment (StripFFP-MV), which is a generalization of the one-dimensional flat folding problem with mountain–valley assignment solved in Ref. [3]. We gave a linear-time algorithm for this problem. The other is the strip flat folding problem with nonacute zigzag creases and mountain–valley–unfolded assignment (StripFF-zz-MVU ($\geq 90°$)). We showed that, assuming the angle between adjacent creases is $\geq 90°$, every instance is flat foldable.

An interesting open problem would be to analyze the acute zigzag case, where the angle between adjacent creases is $< 90°$. A broader open question is to determine the computational complexity of flat foldability for any pattern of noncrossing creases in a strip (rectangle, parallelogram, or trapezoid).

**Acknowledgments** We would like to acknowledge the participants of the University of Tokyo Workshop on Geometric Folding Algorithms in May 2016 where we had a preliminary discussion on these problems. Hiro Ito is supported by JST CREST JPMJCR1402 and JSPS KAKENHI 15K11985. Chie Nara is supported by MEXT/JSPS KAKENHI 20K03726. Tomohiro Tachi is supported by JSPS KAKENHI 16H06106.

### References

[1] Akitaya, H., Avery, C., Bergeron, J., Demaine, E.D., Kopinsky, J. and Ku, J.: Infinite All-Layers Simple Foldability, *Graphs and Combinatorics*, Vol.36, pp.231–244 (2020).

[2] Akitaya, H.A., Cheung, K.C., Demaine, E.D., Horiyama, T., Hull, T.C., Ku, J.S., Tachi, T. and Uehara, R.: Box pleating is hard, *Revised Papers from the 18th Japan Conference on Discrete and Computational Geometry and Graphs* (*JCDCGG 2015*), Lecture Notes in Computer Science, Vol.9943, pp.167–179 (2015).

[3] Arkin, E.M., Bender, M.A., Demaine, E.D., Demaine, M.L., Mitchell, J.S.B., Sethia, S. and Skiena, S.S.: When Can You Fold a Map?, *Computational Geometry: Theory and Applications*, Vol.29, No.1, pp.23–46 (2004).

[4] Bern, M. and Hayes, B.: The Complexity of Flat Origami, *Proc. 7th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp.175–183 (1996).

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

[6] Hull, T.: On the Mathematics of Flat Origamis, *Congressus Numerantium*, Vol.100, pp.215–224 (1994).

[7] Hull, T.: Counting Mountain-Valley Assignments for Flat Folds, *Ars Combinatoria*, Vol.67, pp.175–187 (2003) (online), available from ⟨http://web.merrimack.edu/˜thull/papers/CountingFolds.pdf⟩ .

[8] Matsukawa, Y. and Mitani, J.: Mathematics of Flat Origami, *Trans. Japan Society for Industrial and Applied Mathematics*, Vol.27, No.4, pp.333–353 (2017).

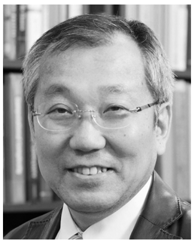
[9] Murata, S.: A Theory of Paper Sculpture, *Bulletin of Oita prefectural Junior College of Art*, Vol.4, pp.61–66 (1966).

[10] Murata, S.: A Theory of Paper Sculpture: II, *Bulletin of Oita prefectural Junior College of Art*, Vol.5, pp.29–37 (1966).

[11] Nara, C.: Approaches to the Study of Origami using Mathematical Sciences, *Journal of the Japan Society of Mechanical Engineers*, Vol.119, No.1175, pp.548–551 (2016).

**Erik D. Demaine** received a B.Sc. degree from Dalhousie University in 1995, and M.Math. and Ph.D. degrees from the University of Waterloo in 1996 and 2001, respectively. Since 2001, he has been a professor in computer science at the Massachusetts Institute of Technology. His research interests range throughout algorithms, from data structures for improving web searches to the geometry of understanding how proteins fold to the computational difficulty of playing games. In 2003, he received a MacArthur Fellowship as a "computational geometer tackling and solving difficult problems related to folding and bending-moving readily between the theoretical and the playful, with a keen eye to revealing the former in the latter". He cowrote a book about the theory of folding, together with Joseph O'Rourke (*Geometric Folding Algorithms*, 2007), and a book about the computational complexity of games, together with Robert Hearn (*Games, Puzzles, and Computation*, 2009). With his father Martin, his interests span the connections between mathematics and art.

**Martin L. Demaine** is an artist and computer scientist. He started the first private hot glass studio in Canada and has been called the father of Canadian glass. Since 2005, he has been the Angelika and Barton Weller Artist-in-Residence at the Massachusetts Institute of Technology. Martin works together with his son Erik in paper, glass, and other material. Their artistic work includes curved origami sculptures in the permanent collections of the Museum of Modern Art in New York, and the Renwick Gallery in the Smithsonian. Their scientific work includes over 60 published joint papers, including several about combining mathematics and art.



**Hiro Ito** received B.E., M.E., and Ph.D. degrees in the Department of Applied Mathematics and Physics from the Faculty of Engineering, Kyoto University in 1985, 1987, and 1995, respectively. 1987–1996, 1996–2001, and 2001–2012, he was a member of NTT Laboratories, Toyohashi University of Technology, and Kyoto University, respectively. Since 2012, he has been a full professor in School of Informatics and Engineering at The University of Electro-Communications (UEC). He has been engaged in research on discrete algorithms mainly on graphs and networks, discrete mathematics, recreational mathematics, and algorithms for big data. Dr. Ito is a member of IEICE, the Operations Research Society of Japan, the Information Processing Society of Japan, and the European Association for Theoretical Computer Science. He is also a member of the steering committee of JCDCG[3].

**Chie Nara** She received her B.A., M.S. and Ph.D. degrees from Ochanomizu University. She served as a professor at Tokai University before taking up the current research position at Meiji University. Recently, she is also engaged in Origami-engineering to develop foldable products as application of her research.

**Izumi Shirahama** received her B.Eng. in Informatics and Network Engineering at the University of Electro- Communications (UEC Tokyo) in 2017. She worked on initial consideration for strip flat problem.

**Tomohiro Tachi** received his B.Eng, M.Eng, and Ph.D. degrees from the University of Tokyo in 2005, 2007, and 2010 respectively. He is currently an associate professor in Graphic and Computer Sciences at the Graduate School of Arts and Sciences, the University of Tokyo. He keeps exploring three-dimensional and kinematic origami through computation and developed software tools including "rigid origami simulator", "origamizer", and "freeform origami", which are available from his website. His research interests include origami, structural morphology, architectural geometry, and computational fabrication.

**Mizuho Tomura** received her B.Eng. and M.Eng. in Informatics and Network Engineering at the University of Electro-Communications (UEC Tokyo) in 2018 and 2020, respectively. She worked on algorithms for the strip flat folding problem.