Map Folding

Rahnuma Islam Nishat* Sue Whitesides *†

Abstract

A crease pattern is an embedded planar graph on a piece of paper. An $m \times n$ map is a rectangular piece of paper with a crease pattern that partitions the paper into an $m \times n$ regular grid of unit squares. If a map has a configuration such that all the faces of the map are stacked on a unit square and the paper does not self-intersect, then it is flat foldable, and the linear ordering of the faces is called a valid linear ordering. Otherwise, the map is unfoldable. In this paper, we show that given a linear ordering of the faces of an $m \times n$ map, we can decide in linear time whether it is a valid linear ordering, which improves the quadratic time algorithm of Morgan. We also define a class of unfoldable $2 \times n$ mountain-valley patterns for every $n \geq 5$.

1 Introduction

A *piece of paper* is a connected polygon in $\mathbb{R}^2$, with or without holes. A paper has a *light side* and a *dark side*. A *crease pattern* is an embedded planar graph on a piece of paper. Each edge of a crease pattern that is not on the boundary of the paper is called a *crease*. The crease pattern divides the surface of the paper into a set of bounded regions called *faces*. Each face is bounded by a set of creases and possibly by part of the boundary of the paper. Each crease is incident to exactly two faces. A *vertex* of a crease pattern is an endpoint of a crease that is not on the boundary of the paper.

If a crease pattern partitions a rectangular piece of paper without holes into an $m \times n$ regular grid of unit squares, then the piece of paper is called an $m \times n$ *grid paper* or an $m \times n$ *map*. A crease pattern on an $m \times n$ map is called an $m \times n$ *crease pattern*. When we fold a piece of paper with a given crease pattern, we are restricted to fold the paper only along the creases. A crease can be folded either as a *mountain* or as a *valley*. A *mountain fold* folds the paper such that the two faces incident to the crease touch each other on the dark side after the fold. Similarly, a *valley fold* folds the paper such that the two faces incident to the crease touch each other on the light side after the fold.

*Department of Computer Science, University of Victoria, BC, Canada, rnishat@uvic.ca, sue@uvic.ca

†Supported by an NSERC Discovery Grant and the University of Victoria. Results here appear in the first author's MSc thesis [9].

A *mountain-valley assignment* is a many-to-one function from the creases in a crease pattern to a label set $\{M, V\}$. A *mountain-valley pattern* is a crease pattern together with a mountain-valley assignment. Figure 1(a) shows a mountain-valley pattern on a 3×3 map. The valley creases are denoted by triple-dot dashed (red) lines and the mountain creases are denoted by dashed (blue) lines.

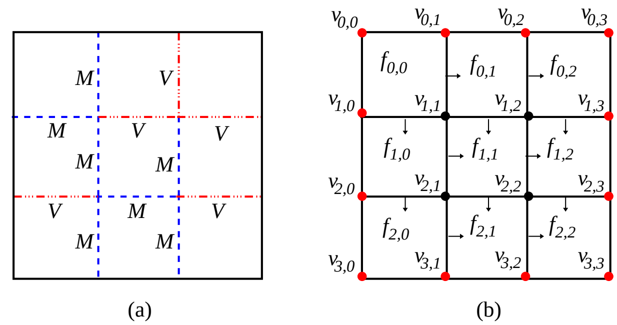

Figure 1: (a) A mountain-valley pattern on a 3×3 map. (b) A 3×3 map with crease pattern $\mathbb{C}$, where the vertices of $\mathbb{C}$ are shown as black disks and the dummy vertices are shown as red disks.

Hull [4] gave upper and lower bounds on the number of flat foldable mountain-valley assignments on a single-vertex crease pattern on a disk. Researchers have also been interested in combinatorial problems in origami. Justin [5] enumerated a number of unfoldable mountain-valley patterns on 2×5, 2×6 and 2×7 maps. Uehara [11] showed that any mountain-valley $1 \times n$ pattern is flat foldable, and gave new upper and lower bounds on the number of flat folded states for that case. Jack Edmonds posed the following open problem in 1997 [3].

Open Problem: *What is the complexity of deciding whether an $m \times n$ map with a given mountain-valley pattern is* flat foldable*?*

In an attempt to answer the above question, Arkin *et al.* [1] introduced "simple folding" techniques. They showed that any flat foldable 1D mountain-valley pattern is flat foldable using simple folding. Recently, Morgan [8] has given an $O(n^9)$ algorithm for $2 \times n$ mountain-valley patterns and an exponential time algorithm for $m \times n$ mountain-valley patterns. Bern and Hayes [2] proved that both the *flat foldability* and the *assigned flat foldability* problems are NP-complete. The *flat foldability* problem asks whether a paper with a given crease pattern has a final flat folded state, where the creases are not necessarily labeled and the crease pattern is "locally flat foldable". In an assigned flat foldability problem, each crease is labeled either mountain or valley.

In this paper, we give an exponential time algorithm to determine whether a given $m \times n$ mountain-valley pattern is flat foldable. We also investigate the combinatorial properties of mountain-valley patterns. The main results of the paper are as follows.

In Section 3, we show that given a linear ordering of the faces of an $m \times n$ mountain-valley pattern, we can decide in linear time whether it is a valid linear ordering or not. In Section 4, we give an exponential time algorithm to decide flat foldability of an $m \times n$ mountain-valley pattern. In Section 5, we show that there is an unfoldable $2 \times n$ mountain-valley pattern for each and every $n \geq 5$ and define a class of unfoldable $2 \times n$ mountain-valley patterns for every $n \geq 5$.

2 Preliminaries

In this section, we define the terminology used throughout the paper. We also mention some previous results that we use.

Let P be a piece of paper. Let $\mathbb{C}$ be a crease pattern on P such that P is flat foldable with respect to $\mathbb{C}$. Let v be a vertex of $\mathbb{C}$. Suppose we draw any circle r around v such that no other vertex of $\mathbb{C}$ is on r or inside r. Since P is flat foldable with respect to $\mathbb{C}$, the disk bounded by r is also flat foldable. The following results are known for a crease pattern on a disk with a single vertex v at the center of the disk.

Lemma 1[7] *The difference between the number of creases with the label mountain and the number of creases with the label valley meeting at v is* 2.

Let P be a map with crease pattern $\mathbb{C}$. It easily follows from Lemma 1 that each vertex of $\mathbb{C}$ must have either three mountain creases and one valley crease or three valley creases and one mountain crease incident to it when P is flat foldable. If the conditions stated in Lemma 1 is satisfied for all the vertices in $\mathbb{C}$, we say that the crease pattern $\mathbb{C}$ is *locally flat foldable.*

A *fragment* of P is a subset of faces of $\mathbb{C}$ that form a connected rectangular region (without a hole). A *flat folded state* of P is a stack of disjoint fragments of P that are parallel to each other, connected along the creases of $\mathbb{C}$, and such that the union of all the fragments is P. Each fragment in the stack is called a *layer.* A *final flat-folded state* of P is a flat folded state where each layer consists of exactly one face of $\mathbb{C}$. If P has a final flat-folded state, then P is *flat foldable.* A final flat-folded state of P is also called a final flat-folded state of $\mathbb{C}$. Figure 2 shows an example of a flat folded state of a 6×8 map.

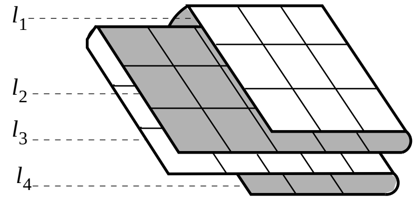

Figure 2: A flat folded state of P with four layers l_1, l_2, l_3 and l_4.

A *vertex of* $\mathbb{C}$ is an endpoint of a crease of $\mathbb{C}$ that is not on the boundary of the paper. A *vertex of* P is either an endpoint of a crease or a corner of the boundary of P. We call a vertex of P that is not a vertex of $\mathbb{C}$ a *dummy vertex.* Figure 1(b) shows the vertices of a 3×3 map.

We denote by $f_{i,j}$ a face of $\mathbb{C}$ that has the vertices $v_{i,j}, v_{i+1,j}, v_{i,j+1}, v_{i+1,j+1}$ on its boundary as shown in Figure 1(b). For each face $f_{i,j}$ of $\mathbb{C}$, we associate the creases $(v_{i,j}, v_{i+1,j})$ (left side of the unit square), where $0 < j < n$, and $(v_{i,j}, v_{i,j+1})$ (top of the unit square), where $0 < i < m$, to $f_{i,j}$. The creases associated with each face are shown in Figure 1(b).

A *column* c_j of $\mathbb{C}$ is a set of m faces $f_{0,j}, f_{1,j}, \ldots, f_{m-1,j}$, where $0 \leq j \leq n-1$. A *row* r_i of $\mathbb{C}$ is a set of n faces $f_{i,0}, f_{i,1}, \ldots, f_{i,n-1}$, where $0 \leq i \leq m-1$. The creases associated with a column c_j are the creases associated with the faces in c_j. Similarly, the creases associated with a row r_i are the creases associated with the faces in r_i.

2.1 Checkerboard Pattern

Let us assume that P is flat foldable and let L be the linear ordering of the faces of $\mathbb{C}$ in a final flat folded state S_f of P. Without loss of generality we assume that the face $f_{0,0}$ is facing light side up in S_f and the vertex $v_{0,0}$ is incident to the top-left corner of the unit square on which the faces are stacked. It is easy to observe that the faces that share an edge with $f_{0,0}$ must face dark side up. In a similar way, if a face $f_{i,j}$, $0 \leq i \leq m-1$ and $0 \leq j \leq n-1$, is facing light side (respectively, dark side) up, then all the faces that share an edge with $f_{i,j}$ must face dark side (respectively, light side) up. So the faces of $\mathbb{C}$ form a *checkerboard pattern*, where the color of a face f depends on which side of f must face up in any final flat folded state of P (under our assumption), as shown in Figure 4(a).

2.2 Butterflies

A *butterfly* B is a pair of faces f, f' of $\mathbb{C}$ incident to the same crease e. We call f and f' the *wings* of B and the crease e the *hinge* of B. A *pair of butterflies* is a set of two butterflies B_1 and B_2 with no wing in common.

Let S be any flat folded state of P and let B_1, B_2 be a pair of butterflies such that the wings of B_1, B_2 lie above the same unit square u on the XY-plane and the hinges of B_1, B_2 lie above the same edge of u. Let the wings of B_1 and B_2 be f_1, f_1' and f_2, f_2', respectively. Here, f_1 and f_2 denote the lower wings and f_1' and f_2' denote the upper wings of their respective butterflies. Then the ordering of the four wings from bottom to top must be one of the following: (f_1, f_1', f_2, f_2'), (f_2, f_2', f_1, f_1'), (f_2, f_1, f_1', f_2') or (f_1, f_2, f_2', f_1'), as shown in Figure 3(a)–(d), respectively. Note that the ordering of the wings cannot be (f_1, f_2, f_1', f_2') or (f_2, f_1, f_2', f_1') as P would self-intersect. If the order of the wings is as in Figure 3(a) or (b), we say that B_1 and B_2 *stack*. Otherwise, we say that B_1 and B_2 *nest*.

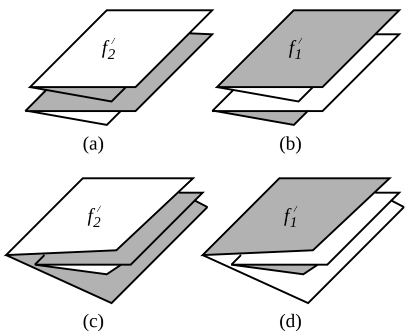

Figure 3: A pair of butterflies B_1, B_2, where (a) B_2 is stacked on B_1, (b) B_1 is stacked on B_2, (c) B_1 nests in B_2, and (d) B_2 nests in B_1.

If P is flat foldable, then there exists a final flat folded state of P where all the faces of $\mathbb{C}$ lie above a unit square u on the XY-plane. Since we assume that in any final flat folded state (if one exists) of P, $v_{0,0}$ is incident to the top-left corner of u, the horizontal creases in row r_i, $0 \le i \le m-1$, lie above the top edge of u when i is even. We call the butterflies that have these creases as hinges the *north butterflies*. Similarly, the horizontal creases in row r_i, $0 \le i \le m-1$, lie above the bottom edge of u when i is odd. We call butterflies with these hinges the *south butterflies*. The vertical edges in column c_j, $0 \le j \le n-1$, lie above the left edge of u when j is even and they lie above the right edge of u when j is odd. We call butterflies with those hinges the *west butterflies* and the *east butterflies*, respectively. A pair of butterflies B_1 and B_2 is called a pair of *twin butterflies* if both of them are north or south or east or west butterflies.

2.3 Directed Network

Let B be a butterfly of P with hinge e and wings f, f'. Since f and f' are adjacent faces, exactly one of them has light side up in the checkerboard pattern. Without loss of generality, we assume that f has the light side up. Then the label of e (mountain or valley) determines the *ordering* of f and f'. If e has the label mountain, then f (the face with the light side up) comes above f' (the face with dark side up). We denote the ordering by $f \prec f'$, where $\prec$ means 'comes above'. On the other hand, if e has the label valley, then the ordering is $f' \prec f$. The labels of all the creases give a *directed network* (of the faces) of $\mathbb{C}$. For example, Figure 4(a) shows a 2×2 mountain-valley pattern. The creases

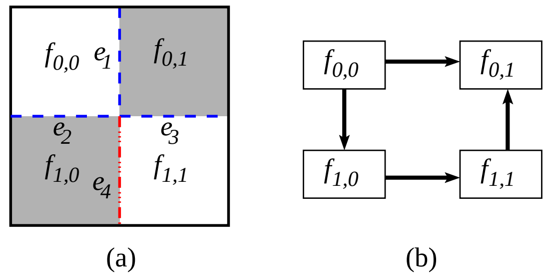

Figure 4: (a) A 2×2 mountain-valley pattern, and (b) the directed network.

e_1, e_2, e_3 and e_4 impose the following directed network. (See Figure 4(b).) $f_{1,1}$ is facing light side up and $f_{1,0}$ is facing dark side up. Since e_4 has the label valley, then $f_{1,0} \prec f_{1,1}$. $f_{0,0}$ is facing light side up and $f_{1,0}$ is facing dark side up. Since e_2 has the label mountain, then $f_{0,0} \prec f_{1,0}$. $f_{0,1}$ is facing dark side up and $f_{0,0}$ is facing light side up. Since e_1 has the label mountain, then $f_{0,0} \prec f_{0,1}$. In a similar way, $f_{1,1} \prec f_{0,1}$.

Since $f_{0,0}$ comes above all other faces, it must be the topmost face. Similarly, the bottommost face must be $f_{0,1}$. In fact, in this particular example, the directed network gives a unique candidate for a valid linear ordering of the faces of $\mathbb{C}$, which is $L = (f_{0,0}, f_{1,0}, f_{1,1}, f_{0,1})$ from top to bottom. Notice that the directed network in Figure 4(c) is a directed acyclic graph (DAG).

We claim that the directed network of any flat foldable $m \times n$ mountain-valley pattern must be a DAG. Our approach is independent of, but similar to [8].

Lemma 2 *Let $\mathbb{C}$ be an $m \times n$ mountain-valley pattern. If $\mathbb{C}$ is flat foldable, then its directed network $\mathcal{N}$ is a directed acyclic graph.*

3 Recognizing Valid Linear Orderings

In this section, we give an algorithm to decide whether a given linear ordering of the faces of a mountain-valley pattern $\mathbb{C}$ is a valid linear ordering of $\mathbb{C}$.

Here is an outline of our algorithm, which is essentially the method of [8]. Let L be any linear ordering of the faces of $\mathbb{C}$. For each pair of twin butterflies B_1, B_2 in $\mathbb{C}$, we check whether B_1, B_2 nest, stack or intersect in L. If they either stack or nest, then we check whether the ordering of the wings of B_1 and B_2 satisfies the ordering in the directed network. If each pair of twin butterflies satisfies the ordering in the directed network and does not intersect, then L is a valid linear ordering. Otherwise, it is not a valid linear ordering.

We now prove the correctness the algorithm.

Theorem 3 *Let P be an $m \times n$ map with the mountain-valley pattern $\mathbb{C}$. Let L be a linear ordering of the faces of $\mathbb{C}$. Then L is a valid linear ordering if and only if (a)–(b) hold: (a) every pair of twin butterflies either stacks or nests in L (i.e., satisfies the Butterfly Condition), and (b) L satisfies the directed network $\mathcal{N}$ of $\mathbb{C}$.*

Proof. We first assume that L is a valid linear ordering of $\mathbb{C}$. Then Conditions (a) and (b) must hold. Therefore, we assume that Conditions (a) and (b) hold. We first decompose P into $m \times n$ distinct unit squares, where each square is a face of $\mathbb{C}$. Each of these squares has a light side and a dark side. We stack these squares on a unit square u according to the linear ordering L. The checkerboard pattern of $\mathbb{C}$ decides for each face whether it faces dark or light side up. For each north butterfly B in P, we join its two wings (along the hinge of B) such that its hinge lies above the top edge of u. Since any two north butterflies either nest or stack, there will be no intersection of butterflies. We join the wings of the south, east and west butterflies along the bottom, right and left edge of u in a similar way. In this way, we construct a final flat folded state S_f of P and L is the linear ordering of the faces of $\mathbb{C}$ in S_f. Therefore, L is a valid linear ordering. □

We now calculate the running time of the algorithm.

Theorem 4 *The running time of the algorithm above is $O(m^2n^2)$. With a careful implementation, the running time can be reduced to $O(mn)$ which is linear in the size of the input.*

Proof. Since there are $O(m^2n^2)$ pairs of twin butterflies, and it takes $O(1)$ time to check whether a pair of twin butterflies intersect and whether the order of the wings of each butterfly satisfies the ordering given by the directed network, the total running time is $O(m^2n^2) \times O(1) = O(m^2n^2)$.

We now show a careful implementation to reduce the time complexity. We first check for each pair of north butterflies whether they intersect or not. We take a two dimensional array $M[0 \ldots m-1][0 \ldots n-1]$ and a stack $S[1 \ldots mn]$. At first the stack is empty and each of the entries in M is 0. We preprocess M based on the directed network of $\mathbb{C}$, and M remains unchanged during the processing of the faces. Here are the rules for preprocessing M.

For each $1 \leq i \leq m-2$, where i is odd, and for each $0 \leq j \leq n-1$, we do the following:

- If $f_{i,j}$ faces light side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled mountain, then set $M[i+1, j] = 1$. This means that the face $f_{i,j}$ must occur in L before the face $f_{i+1,j}$.
- If $f_{i,j}$ faces dark side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled mountain, then set $M[i, j] = 1$.
- If $f_{i,j}$ faces light side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled valley, then set $M[i, j] = 1$.
- If $f_{i,j}$ faces dark side up in the checkerboard pattern of $\mathbb{C}$ and the crease between $f_{i,j}, f_{i+1,j}$ is labeled valley, then set $M[i+1, j] = 1$.

We now take the faces in the order given by L and process them as follows. Let the current face be $f_{x,y}$. If $x < 1$, or $x > m-2$ and m is even, then it is not a wing of a north butterfly. Therefore, we proceed to the next face in L. Otherwise, it is the wing of a north butterfly and we examine $M[x, y]$.

- If $M[x, y] = 0$, then it is the wing of a butterfly that occurs before the other wing. Push the face $f_{x,y}$ to S.
- If $M[x, y] = 1$, the other wing of the corresponding butterfly is already in the stack. In this case, we check the top of the stack. If the topmost face in the stack is $f_{x+1,y}$ (x is odd) or $f_{x-1,y}$ (x is even), then we pop the topmost face and proceed to the next face. Otherwise, there is an intersection, and hence L is not a valid linear ordering.

We stop when either we detect an intersection or we reach the end of L. Therefore, checking for intersection among the north butterflies takes $O(mn)$ time. Similarly we check the south, west and east butterflies. The running time of the algorithm is $4 \times O(mn) = O(mn)$, linear in the size of the input linear ordering. □

4 Enumerating Valid Linear Orderings

In this section, we sketch the outline of an exponential-time exact algorithm to enumerate all the valid linear orderings (if any exist) of an $m \times n$ mountain-valley pattern $\mathbb{C}$. Note that the existance of a valid linear ordering of $\mathbb{C}$ proves that $\mathbb{C}$ is flat foldable.

Since $\mathbb{C}$ is a mountain-valley pattern, there is a unique directed network $\mathcal{N}$ of $\mathbb{C}$. We assume that $\mathcal{N}$ is a directed acyclic graph; otherwise $\mathbb{C}$ is not flat foldable by Lemma 2. We now enumerate all the linear orderings of the faces of $\mathbb{C}$ using the algorithm of [10], which takes constant amortized time; i.e., the total running time of the algorithm is $O(e(\mathcal{N}))$, where $e(\mathcal{N})$ is the number of linear orderings generated by the algorithm from the partial order $\mathcal{N}$. We can decide whether a linear ordering is a valid linear ordering in

$O(mn)$ time. From Theorem 1.1 of [6], we know that $e(\mathcal{N}) \leq 2^{mn(log(mn)-H(\mathcal{N}))} \leq 2^{mnlog(mn)} = O(mn^{mn})$, where $H(\mathcal{N}) \leq \log mn$ is the entropy function of $\mathcal{N}$. Therefore, enumerating all valid linear orderings takes $O(mn) \times O(mn^{mn}) = O(mn^{mn+1})$ time.

5 Unfoldable Maps

In this section, we define a class χ_n of unfoldable $2 \times n$ mountain-valley patterns, $n \geq 5$. Note that any $2 \times n$ mountain-valley pattern is flat foldable when $n \leq 4$. We first show a subclass S_n of unfoldable $2 \times n$ mountain-valley patterns, for every $n \geq 5$. We then observe that any map with an unfoldable pattern (i.e., a pattern in S_n) as a fragment is unfoldable. Using this result, we define the class χ_n, which includes S_n as a subclass.

Let P be a $2 \times n$ map with a mountain-valley pattern $\mathbb{C}$. By definition, there are n horizontal creases. We call each of these creases a *spinal crease* and collectively we call these creases the *spine*. We call the $n-1$ vertical creases above the spine the *upper ribs* and the remaining creases the *lower ribs*. We denote the upper ribs in $\mathbb{C}$ by $u_1, u_2, \ldots, u_{n-1}$ from left to right. Similarly, the lower ribs are denoted by $l_1, l_2, \ldots, l_{n-1}$ from left to right, and the spinal creases are denoted by $s_1, s_2, \ldots, s_n$ from left to right. A pair of upper and lower ribs $\{u_i, l_i\}$ incident to the same vertex is called a *pre-spine fold* if they both have mountain or valley label.

We now define the subclass S_n of unfoldable $2 \times n$, $n \geq 5$, mountain-valley patterns. Let $\mathbb{C}$ be a pattern for S_n that satisfies the following (a)–(d).

(a) $\mathbb{C}$ is locally flat foldable.

(b) There are exactly two pre-spine folds $\{u_2, l_2\}$ and $\{u_{n-2}, l_{n-2}\}$.

(c) All the upper ribs receive the same label.

(d) s_3 receives the opposite label of the upper ribs.

The upper ribs of $\mathbb{C}$ can be labeled either mountain or valley. Without loss of generality we assume that the upper ribs of $\mathbb{C}$ receive the label mountain as shown in Figure 5. Consequently, all the lower ribs of $\mathbb{C}$ except l_2 and l_{n-2} must be labeled valley. Since $\{u_2, l_2\}$ and $\{u_{n-2}, l_{n-2}\}$ are the pre-spine folds, l_2 and l_{n-2} receive the same label mountain as u_2 and u_{n-2}. The spinal creases $s_4, \ldots s_{n-2}$ must all be labeled the same as s_3, which according to requirement (s) must be labeled valley. To preserve local flat foldability, the other spinal creases s_1, s_2, s_{n-1}, s_n must be labeled mountain (opposite to the label of s_3).

The following lemma shows that $\mathbb{C}$ is unfoldable.

Lemma 5 *Let $\mathbb{C}$ be a $2 \times n$ mountain-valley pattern in S_n, $n \geq 5$. Then $\mathbb{C}$ is unfoldable.*

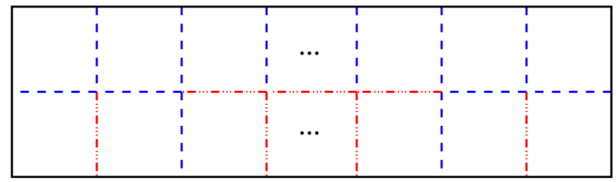

Figure 5: An unfoldable $2 \times n$ mountain-valley pattern.

Proof. *(Sketch of proof)* We show the case when n is odd. The case when n is even is similar. Let L_i be a candidate for a valid linear ordering of the faces in the columns $c_0, \ldots, c_i$ of $\mathbb{C}$, $0 \leq i \leq n-1$. From the directed network of $\mathbb{C}$, $L_3 = f_{0,0} \prec f_{1,0} \prec f_{1,1} \prec f_{1,2} \prec f_{0,2} \prec f_{0,3} \prec f_{1,3} \prec f_{0,1}$ is the unique candidate for the case $i = 3$. We show that for each $4 \leq i \leq n-3$, when $n > 5$, the following (a)–(c) hold.

(a) L_i is the only candidate,

(b) the order of the faces $f_{0,i}, f_{1,i}, f_{0,i-2}$ and $f_{0,i-1}$ in L_i is $f_{0,i-2} \prec f_{1,i} \prec f_{0,i} \prec f_{0,i-1}$, when i is even, and $f_{0,i-1} \prec f_{0,i} \prec f_{1,i} \prec f_{0,i-2}$, when i is odd, and

(c) the faces $f_{0,i}, f_{1,i}, f_{0,i-2}$ and $f_{0,i-1}$ are consecutive as a set (i.e., they appear together, with no other faces lying between the extremal faces in this set).

We first construct the unique candidate L_{n-3} (for $n = 5$, L_{n-3} is L_3). We then show that we cannot avoid self-intersection of the paper when constructing L_{n-2} from L_{n-3}.

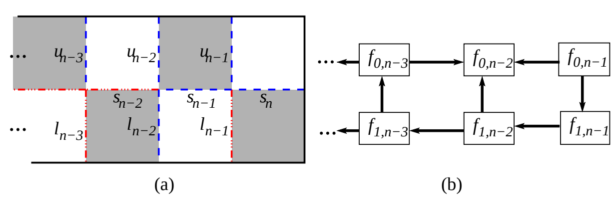

Figure 6: (a) The checkerboard pattern of the last three columns $c_{n-3}, \ldots, c_{n-1}$ of $\mathbb{C}$, when n is odd and (b) the directed network of $\mathbb{C}$.

When n is odd, $f_{0,n-3} \prec f_{0,n-2}$ from Figure 6(b). By the conditions (a)–(c) above, $f_{0,n-5} \prec f_{1,n-3} \prec f_{0,n-3} \prec f_{0,n-4}$ in L_{n-3} (since $n-3$ is even) and these four faces are consecutive. If $f_{0,n-4} \prec f_{0,n-2}$, then the linear ordering $f_{0,n-5} \prec f_{0,n-3} \prec f_{0,n-4} \prec f_{0,n-2}$ causes intersection between the east butterflies u_{n-4} and u_{n-2} (since n is odd, $n-4$ and $n-2$ are odd and hence the butterflies u_{n-4}, u_{n-2} are east butterflies). Therefore, $f_{0,n-2} \prec f_{0,n-4}$ and the linear ordering of the faces $f_{0,n-3}, f_{0,n-2}, f_{0,n-4}$ in L_{n-2} is $f_{0,n-3} \prec f_{0,n-2} \prec f_{0,n-4}$. Since there is a directed path from $f_{0,n-1}$ to $f_{0,n-3}$ in the directed network, $f_{0,n-1} \prec f_{0,n-3}$. We then have to place $f_{0,n-1}$ somewhere above $f_{0,n-3}$. But any such placement will have the linear ordering $f_{0,n-1} \prec f_{0,n-3} \prec f_{0,n-2} \prec f_{0,n-4}$, and thus cause intersection between the west butterflies u_{n-3} and u_{n-1} (the indices of the ribs are even since n is odd and hence the butterflies are west butterflies). Therefore, there is no valid linear ordering of $\mathbb{C}$. □

We claim that any mountain-valley pattern that has an unfoldable fragment is also unfoldable.

Lemma 6 *Let $\mathbb{C}$ be an $m \times n$ mountain-valley pattern. Let $\mathbb{C}'$ be a fragment of $\mathbb{C}$. If $\mathbb{C}'$ is not flat foldable, then $\mathbb{C}$ is not flat foldable.*

We now define a class χ_n of unfoldable $2 \times n$ mountain-valley patterns, where $n \geq 5$. We say a pattern belongs to χ_n if and only if $\mathbb{C}$ satisfies the following (a)–(d).

(a) $\mathbb{C}$ is locally flat foldable.

(b) There are exactly two pre-spine folds $\{u_i, l_i\}$ and $\{u_j, l_j\}$, where $2 \leq i < j \leq n-2$.

(c) All the upper ribs receive the same label and all the lower ribs except u_i, u_j receive the label opposite to the upper ribs.

(d) s_{i+1} receives the opposite label of the upper ribs.

We now show that every member of χ_n is unfoldable.

Theorem 7 *Let $\mathbb{C}$ be a $2 \times n$ mountain-valley pattern in χ_n, where $n \geq 5$. Then $\mathbb{C}$ is unfoldable. Furthermore, membership in χ_n can be tested in linear time.*

Proof. Let $i = 2$ and $j = n - 2$. Then $\mathbb{C} \in S_n$, and hence the pattern is unfoldable by Lemma 5. Therefore, we assume that $\mathbb{C} \notin S_n$. Let $\mathbb{C}'$ be the fragment of $\mathbb{C}$ with the faces in the columns $c_{i-2}, \ldots, c_{j+1}$. Then $\mathbb{C}' \in S_x$, where $x = j-i+4$. Therefore, $\mathbb{C}'$ is unfoldable by Lemma 5. Since a fragment of $\mathbb{C}$ is unfoldable, $\mathbb{C}$ is unfoldable by Lemma 6.

We can check in $O(n)$ time whether all the upper ribs receive the same label (i.e., Condition (c) is satisfied) by scanning from left to right. We can check in $O(n)$ time whether $\mathbb{C}$ is locally flat foldable (i.e., Condition (a) is satisfied) by checking the creases incident to each of the $n-1$ vertices of $\mathbb{C}$. If Conditions (a) and (c) are satisfied, we check in $O(n)$ time whether there are exactly two pre-spine folds (Condition (b)) and get the index i for the leftmost pre-spine fold $\{u_i, l_i\}$. If Conditions (a)–(c) are satisfied, then it takes $O(1)$ time to check whether the label of s_{i+1} is opposite to the label of the upper ribs (Condition (d)). Therefore, it takes $O(n) + O(n) + O(n) + O(1) = O(n)$ time to check whether $\mathbb{C}$ is a member of χ_n. □

6 Conclusion

In this paper, we introduced the concepts of *butterflies*, *checkerboard patterns* and *directed networks*. Using these tools, we gave a linear time algorithm to recognize a valid linear ordering and an exponential time algorithm to decide flat foldability of an $m \times n$ mountain-valley pattern. We also have identified a class of unfoldable $2 \times n$ mountain-valley patterns. It remains open to characterize all the unfoldable $2 \times n$ mountain-valley patterns. It is also open to solve Edmonds' open problem for $m \times n$ maps.

References

[1] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S. B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? *Computational Geometry : Theory and Applications*, 29:23–46, September 2004.

[2] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the 7th annual ACM-SIAM symposium on Discrete algorithms (SODA 1996)*, SODA 1996, pages 175–183. Society for Industrial and Applied Mathematics, 1996.

[3] Erik D. Demaine and Joseph O'Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, New York, NY, USA, 2007.

[4] Thomas Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatorica*, 67, 2003.

[5] Jacques Justin. Aspects mathematiques du pliage de papier (mathematical aspects of paper fold. In H. Huzita, editor, 1*st International Meeting of Origami Science and Scientific Origami*, pages 263–277, 1989.

[6] Jeff Kahn and Jeong Han Kim. Entropy and sorting. In *Proceedings of the* 24*th annual ACM symposium on Theory of computing (STOC 1992)*, pages 178–187. ACM, 1992.

[7] Kunihiko Kasahara and Toshie Takahama. *Origami for the Connoisseur*. Japan Publications Inc., 1987.

[8] Tom Morgan. Map folding. Master's thesis, Massachusetts Institute of Technology, June 2012.

[9] Rahnuma Islam Nishat. Map folding. Master's thesis, University of Victoria, BC, Canada, April 2013.

[10] Gara Pruesse and Frank Ruskey. Generating linear extensions fast. *SIAM Journal on Computing*, 23(2):373–386, 1994.

[11] Ryuhei Uehara. Stamp foldings with a given mountain-valley assignment. In *ORIGAMI 5*, pages 585–597. CRC Press, 2011.