PAPER Special Section on Discrete Mathematics and Its Applications

# Research on Map Folding with Boundary Order on Simple Fold

**Yiyang JIA**[†a], **Jun MITANI**[†b], ***Nonmembers*, *and*** **Ryuhei UEHARA**[††c], ***Member***

**SUMMARY** Folding an $m \times n$ square grid pattern along the edges of a grid is called map folding. We consider a decision problem in terms of whether a partial overlapping order of the squares aligning on the boundary of an $m \times n$ map is valid in a particular fold model called simple fold. This is a variation of the decision problem of valid total orders of the map in a simple fold model. We provide a linear-time algorithm to solve this problem, by defining an equivalence relation and computing the folding sequence sequentially, either uniquely or representatively.
***key words:*** *map folding, simple fold, boundary overlapping order*

## 1. Introduction

We investigate the computational complexity of the validity of the checking problem for a special kind of overlapping order in $m \times n$ maps in the simple fold model. These overlapping orders, called *boundary overlapping orders*, are orders given on only the squares aligning on the boundary. As illustrated in Fig. 1, the input is a total order of these squares indexed from bottom to top when the map is folded to a size of $1 \times 1$. The output is whether the input is a valid boundary overlapping order that corresponds to a flat-folded state of size $1 \times 1$ and is foldable via simple folds or not. We conclude that we can determine the validity of a given boundary overlapping order and find a feasible way to fold it by simple folds for a given valid overlapping order in $O(m + n)$ time. The folding procedure is called a *whole simple folding sequence*. We also provide a method to enumerate all the other feasible folding sequences for the input.

In earlier research [1], we provided an $O(mn)$ time algorithm to solve the same decision problem but with the orderings given on all the squares. When inputs are constituted by partial orders, the relevant decisions are more intricate. With the objective to identify tractable results with such inputs, we consider the boundary overlapping orders.

The base model of this problem, the map folding problem, is a fundamental topic in flat-folding. Variations of this have been investigated for many years. Bern and Hayes proved the NP-hardness of the decision problem on the flat-

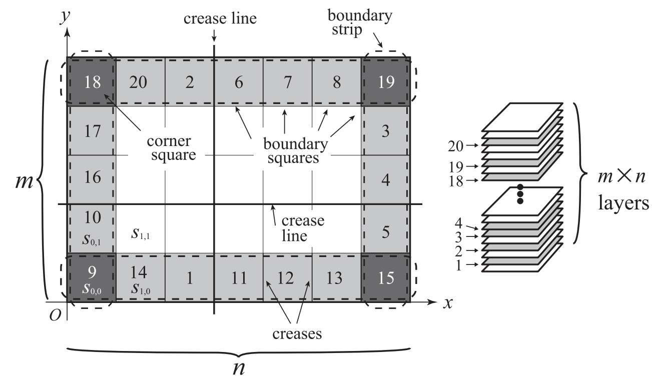


**Fig. 1** A map is folded into a square of $m \times n$ layers. The overlapping order of the boundary squares (colored gray) in the folded state is given.

foldability of a general crease pattern [2]. Map folding problems tend to be recognized as a simplification from general patterns to grid patterns. Further, determining the flat-foldability restricted to box-pleated crease patterns is NP-hard [3]. The original map folding problem, which was proposed by Edmonds in 1997 and remains unsolved, explores the complexity to decide the flat-foldability of a rectangular grid pattern of size $m \times n$ [4]. Every edge shared by two squares is assigned as either a *Mountain* or a *Valley*. The entire assignment is called *a Mountain-Valley assignment* (*an MV assignment*). Arkin et al. [5] provided a method to determine the flat-foldability of one-dimensional maps. Morgan [6] gave an $O(n^9)$ time algorithm to determine the flat-foldability of maps of size $2 \times n$. There is still no result for $m \times n$ maps by general folds when $m \geq 3$.

In opposition to general folds, Arkin et al. [5] proposed a particular type of fold called the *simple fold*. A simple fold applies on a certain number of layers along a single line so that the state after a simple fold is also flat. In [5], the simple fold concept is divided into three types. Among them, some-layers simple fold (Fig. 2), in which, at every step, some layers are folded simultaneously, is the most popular model and also used in this research. Hereafter, we use the term *simple fold* to refer to it. Results concerning the computational complexity of simple folding general patterns are given in [7].

Some extant research has focused on the reachability of a given final overlapping order of layers. A reachable overlapping order is called *valid*. In the general fold model, Bern and Hayes showed that determining a suitable overlapping order is NP-hard [2]. Nishat [8] provided an $O(mn)$

Manuscript received September 18, 2020.
Manuscript revised January 18, 2021.
Manuscript publicized March 8, 2021.
[†]The authors are with University of Tsukuba, Tsukuba-shi, 305-8577 Japan.
[††]The author is with School of Information Science, JAIST, Nomi-shi, 923-1292 Japan.
a) E-mail: yiyangjia@cgg.cs.tsukuba.ac.jp
b) E-mail: mitani@cs.tsukuba.ac.jp
c) E-mail: uehara@jaist.ac.jp
DOI: 10.1587/transfun.2020DMP0017

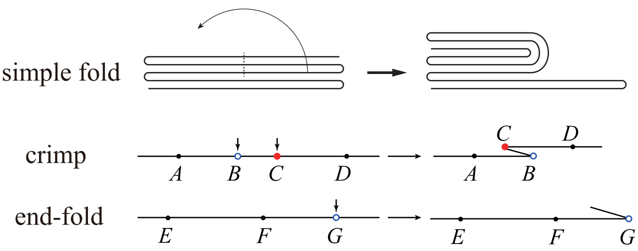


**Fig. 2** An example of some-layers simple fold and the two basic operations.

time algorithm to decide the validity of a given order of map folding. Enumerating all the valid orders costs exponential time even for 1D maps [9], [10]. Uehara [10] proved that for maps of size $1 \times n$, all the reachable flat states by general folds are also reachable by simple folds and the validity can be checked in $O(n)$ time. Research about the existence of valid total orders with partial orders as inputs shows intractability in general cases. For example, the general decision problem on the plan viability is NP-complete [11]. Despite the wide range of relevant research, no study has hitherto considered both partial orders and simple folds in map folding problems. Considering this new combination, we investigate the validity of boundary overlapping orders in the simple fold model.

## 2. Preliminaries

### 2.1 Map Notations

A map $M_{m,n}$ is a rectangular sheet of paper with $m \times n$ congruent squares arrayed in $m$ rows and $n$ columns. Its two sides are differentiated as the front and the back. The basic notation of $M_{m,n}$ is illustrated in Fig. 1. The *crease pattern* is specified as a grid pattern consisting of all the edges of the squares other than the ones on the boundary of the map. These edges are called *creases* and their non-boundary endpoints are called *vertices* (following [5]). An *MV assignment* is a mapping from the set of creases to the set $\{M, V\}$. "$M$"s are denoted by red solid-line segments, and "$V$"s are denoted by blue dashed line segments. A *crease line* comprises a set of creases aligning on the same line and passing through the map. We use points to denote the creases in a 1D map. Each vertex in $M_{m,n}$ has degree four. A vertex is locally flat-foldable if and only if exactly three of its creases are assigned the same [12]–[14].

We use $s_{i,j}$ $(0 \leq i < m, 0 \leq j < n)$ to refer to the square with its lower-left vertex located at $(i, j)$ before any fold. Specifically, our research concerns the squares aligning on the boundary of $M_{m,n}$ (shadowed in figures), i.e., the set $\{s_{i,j} \mid i \in \{0, m-1\} \text{ or } j \in \{0, n-1\}\}$. We call these squares *boundary squares*. A *boundary strip* is a set of boundary squares aligning on the same side. *Corner squares* refer to $s_{0,0}$, $s_{m-1,0}$, $s_{0,n-1}$, $s_{m-1,n-1}$, which are located at the corner in the initial state. Without loss of generality, $s_{0,0}$ is fixed to (0, 0) and it always faces the front up during the folding.

### 2.2 Fold Definitions

Arkin et al. defined two kinds of folds, *end-folds* and *crimps*, for a given MV assignment [5]. In a 1D map (see Fig. 2), an end-fold is a fold at either the first or the last crease point. The interval between the last crease point and its corresponding end of the map is not longer than its neighbor interval. A crimp is a fold along a pair of adjacent crease points labeled "$MV$" or "$VM$", where the length of the interval between the two creases is a local minimum value. We use the same operations to indicate our foldings. Instead of handling a map with all creases, we put forward a method to compute the current map at each step by only regarding the creases to be folded at the step and neglecting the creases remaining unfolded (details will be given in Sect. 4.1). By mathematical induction, it can be clarified that any valid partly folded state in a simple fold model is reachable by our crimps and end-folds.

Both crimps and end-folds can be considered as conditional simple folds. On the other hand, we call the unconditional simple folds along single lines *general simple folds*. Hereafter, *a simple fold* refers to either a crimp or an end-fold unless expressly stated otherwise. For a valid input order, we seek a folding process $F$ composed by crimps and end-folds. Of note, we require $F$ to satisfy the following: once the surfaces of two squares touch each other, the two squares can never be separated by subsequent folding operations. This *non-separating property* is crucial in our approach. Furthermore, it forces the crimps and end-folds in $F$ to be applied to all the layers where the corresponding crease lines exist. We show that other possible folding processes which do not satisfy this non-separating property can be produced from $F$.

We define a *simple folding sequence* as a sequence of crimps and end-folds. A simple folding sequence from the map to a flat state of size $1 \times 1$ is called a *whole simple folding sequence*. Every partly flat-folded state can be viewed as a smaller map. The sequence of partly flat-folded states of boundary squares is indicated by $R = (R_0, R_1, R_2, \ldots, R_t)$, where $R_t$ represents the final state. We define all the possible partly flat-folded states by the set $\mathscr{R}$. Then, any end-fold or crimp can be described as a mapping $f : \mathscr{R} \to \mathscr{R}$. Specifically, a mapping from the state $R_{i-1}$ to $R_i$ is denoted by $f_i$ where $f_i(R_{i-1}) = R_i$. We say that a pair of consecutive folds $f_i$ and $f_{i+1}$ are *interchangeable* if $f_i \circ f_{i+1}(R_{i-1}) = f_{i+1} \circ f_i(R_{i-1}) = R_{i+1}$, where $\circ$ is the composition of mappings. A folding sequence from $R_i$ to $R_j$ with $i < j$ is denoted by $(f_{i+1}, f_{i+2}, \ldots, f_j)$ and corresponds to $f_j \circ f_{j-1} \circ \ldots \circ f_{i+1}$ mapping $R_i$ to $R_j$. Then, a whole simple folding sequence $F$ is abstracted as the composition of all the $f_i$s where $1 \leq i \leq t$. Let $\mathscr{F}$ be the set of feasible whole simple folding sequences; $\mathscr{F}$ is the solution space of the decision problem on the validity of the input order.

From another perspective, $F$ can be partitioned into some sub-sequences of parallel folds. We denote these sub-sequences by $P = (p_1, p_2, \ldots, p_u)$ with $1 \leq u \leq t$. This satisfies that (1) each $p_k$ is a maximal set of some parallel $f_i$s which are perpendicular to those comprising $p_{k+1}$ and (2) the crease lines in $p_k$ are folded directly after $p_{k-1}$ and before $p_{k+1}$. The sub-sequences in $P$ are uniquely decided by $F$ and vice versa. Then, we have Observation 1.

**Observation 1.** *The elements in $p_0$, $p_2$, $p_4, \ldots, p_{2i}$ are parallel folds; The elements in $p_1$, $p_3$, $p_5, \ldots, p_{2i+1}$ are parallel folds, which are perpendicular to the elements in $p_0$, $p_2$, $p_4, \ldots, p_{2i}$.*

Without loss of generality, we assume that all elements $p_{2i}$ are parallel to the $x$-axis, and all $p_{2i+1}$ are parallel to the $y$-axis.

The term *neighbor squares* refers to a pair $s_{a,b}$, $s_{a+1,b}$ or a pair $s_{a,b}$, $s_{a,b+1}$ in the initial state. When a partly flat-folded state $R_i$ is viewed as a new map, two squares sharing a crease on the new map are called *neighbor squares in $R_i$*. In any partly or totally flat-folded state, we say that a pair of squares $s_i$ and $s_j$ whose faces touch each other are *adjacent* and denoted by $s_i \leftrightarrow s_j$. Furthermore, we denote the closure of $\leftrightarrow$ by $\leftrightarrow^*$, which can be constructed as follows: (1) $s_i \leftrightarrow^* s_i$; (2) $s_i \leftrightarrow^* s_k$ if $s_i \leftrightarrow^* s_j$ and $s_j \leftrightarrow s_k$.

For a square $s_j$ in a folded state $R_i$, we can consider the closure of squares adjacent to $s_j$ as the set of squares $\{s_l | s_j \leftrightarrow^* s_l\}$. In our objective $F$, adjacent squares after any $f_i$ would never become non-adjacent. Thus, the closure of $s_j$ in $R_i$ would be a subset of the closure of $s_j$ in $R_k$ for any $k > i$. Let $O_i(a, b)$ denote the set of squares $\{s_l | s_{a,b} \leftrightarrow^* s_l\}$ in the state $R_i$. For a given $R_i$, when the ordering of the squares in the set $O_i(a, b) = \{s_1, \ldots, s_l\}$ is fixed, we sometimes use an ordered set $(s_1, s_2, \ldots, s_l)$ such that $s_j \leftrightarrow s_{j+1}$(for any $1 \leq j < l$) to describe the order. The ordering of any two neighbor squares in $R_t$ is decided by the MV assignment [8].

### 2.3 Problem Definitions

Herein, a *valid overlapping order* refers to a feasible overlapping order $O_t(0, 0)$ of the map in the simple fold model. A *valid boundary overlapping order* is a total valid overlapping order given on the set of boundary squares.

Our focus is the validity of the boundary overlapping order of $m \times n$ maps. The problem is described as follows. Given an order $O$ of the boundary squares, is $O$ a valid boundary overlapping order in the simple fold model?

We pose Theorem 2 as our conclusion.

**Theorem 2.** *There exists an $O(m + n)$ time algorithm that determines the validity of the boundary overlapping order $O$ in the simple fold model and gives a feasible whole simple folding sequence $F$ when $O$ is determined to be valid.*

## 3. Outline

We study the decision problem on the validity of a given order of the boundary squares in the simple fold model. The method is summarized as follows. First, the MV assignment on the boundary strips is computed by traversing $O$. Then, $F$ is obtained by computing the sub-sequences of parallel folds $(p_1, p_2, \ldots, p_u)$. Every $p_k$ folds two parallel boundary strips perpendicular to $p_k$ in the same way. Therefore, we decide the simple folds in each $p_k$ by finding the same folds on two parallel boundary strips.

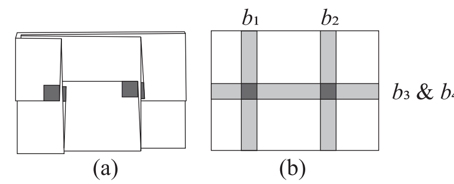


**Fig. 3** An instance with two parallel boundary strips overlapping each other. (a) shows the state of the map. (b) is the top view of the flat-folded state, i.e., the view from the top of $s_{0,0}$, where locations of the boundary strips and boundary squares in the folded state are shadowed. $b_1$, $b_2$, $b_3$ and $b_4$ correspond to boundary strips.

Since there may exist other whole simple folding sequences inducing the given boundary overlapping order, we have to prove that if no $F$ can be obtained with our algorithm, then there exists no valid whole simple folding sequence leading to the given order. The strategy is to consider the interchangeable folds in $F$ so that any other valid whole simple folding sequences can be obtained by interchanging these folds in $F$. By Cayley's theorem [15], it is sufficient to consider only the interchangeability between consecutive simple folds.

We first analyze the condition that renders consecutive folds interchangeable. Then, we assign an equivalence relation on $\mathscr{F}$ such that equivalent whole simple folding sequences lead to the same boundary overlapping order. $F$ is computed as the representative of its equivalence class. Then, deciding the validity of $O$ resolves to deciding the existence of $F$.

Note that we do not discuss the interchangeable folds when two parallel boundary strips totally overlap each other (as illustrated in Fig. 3) because that can be handled with the same operation for a single $p_k$ by considering the overlapping formed on $\{b_1, b_2\}$ and $b_3$ together.

## 4. Interchangeable Consecutive Folds

The interchangeable conditions of consecutive parallel folds and consecutive perpendicular folds are provided in this section. Crimps and end-folds might be updated to new ones by the condition in Sect. 4.2 so that other feasible foldings $F$ can also be obtained by the interchange. By this exhaustion, we concern only crimps and end-folds as the interchangeable perpendicular folds.

### 4.1 Recognizing Consecutive Parallel Folds

Because any partly flat-folded state can be viewed as a map, here we only discuss $p_k$ with an odd $k$. The creases on the two boundary strips simultaneously folded by $p_k$ should have the same assignment, which can be confirmed by a parity check of the coordinates [8].

To simplify the exposition, we discuss the folds on a single boundary strip. The two states after a crimp and after an end-fold are given in Fig. 2. Note that instead of recognizing the crimps and end-folds on the entire crease pattern, once a step, we only consider the creases which are supposed to be currently folded and assume the other creases do not exist. These currently folded creases can be clarified during the traverse of $O$. The size of the map is assumed to be $1\times n'$ with its left end located on $x = 0$. We denote the coordinates of points $A$ to $G$ as $x_A$ to $x_G$, which are all integers. Then, we have $x_B - x_A > x_C - x_B < x_D - x_C$ for the crimp and $n' - x_G \le x_G - x_F$ for the end-fold. The equals symbol is permitted only for end-folds. The two states are respectively described by the following equations, where $o(x)$ indicates the squares folded to the coordinate $(x, 0)$, and tuples indicate the order of the squares from bottom to top.

For the crimp, we have

$$o(x) = \begin{cases} (s_x), & x < 2x_B - x_C \\ (s_x, s_{2x_B-x-1}, s_{x+2x_C-2x_B}), & \\ & 2x_B - x_C \le x < x_B \\ (s_{x+2x_C-2x_B}), & \\ & x_B \le x < 2x_B + n' - 2x_C \end{cases}$$

and for the end-fold, we have

$$o(x) = \begin{cases} (s_x), & x < 2x_G - n' \\ (s_x, s_{2x_G-x-1}), & 2x_G - n' \le x < x_G \end{cases}$$

By repeating the following three steps, it can be determined whether $R_j$ can be reached from $R_i$ by parallel folds. If reachable, the elements and the inner order of $p_k$ are also decided.

Step 1. Update the 1D map representing a boundary strip of $R_i$ only with the creases whose neighbor squares are in the same $O_j(a, b)$ in $R_j$.

Step 2. Find crimps and end-folds by referring to the equations for $o(x, 0)$. If no fold exists, $R_j$ is not reachable.

Step 3. Reduce the map to a new (smaller) map by applying the folding operations found in Step 2.

This process can be completed in linear time of the 1D map by using a standard graph traverse algorithm, e.g., breadth-first search or depth-first search algorithm. An example of this process for a $1 \times 10$ map is shown in Appendix A.

### 4.2 Interchangeable Parallel Folds

Next, we give the interchangeable condition of the parallel folds. This condition can be used to produce other feasible foldings leading to the same boundary overlapping order. Here we consider interchangeable general simple folds (each along one crease line) to exhaust all the possibilities of valid whole simple folding sequences.

The following lemma provides the necessary and sufficient condition for a pair of parallel folds in a $p_k$ to be interchangeable.

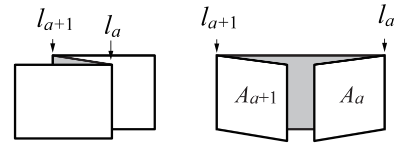


**Fig. 4** Two possible cases for interchangeable parallel folds.

**Lemma 3.** *A pair of consecutive parallel folds $l_a$ and $l_{a+1}$ is interchangeable if and only if (1) or (2) hold.*
*(1) $l_a$ and $l_{a+1}$ are labelled "MV" or "VM".*
*(2) The labels of $l_a$ and $l_{a+1}$ are the same, $A_a \cap A_{a+1} = \emptyset$ where $A_a = \{(i, j)|O_a(i, j) \neq O_{a-1}(i, j)\}$ and $A_{a+1} = \{(i, j)|O_{a+1}(i, j) \neq O_a(i, j)\}$.*

Case (2) means that the overlapping part formed by folding along $l_a$ does not overlap with the overlapping part formed by folding along $l_{a+1}$. Note that $l_a$ and $l_{a+1}$ may only affect the overlapping on some layers. Both cases are illustrated in Fig. 4. The proof is omitted since the lemma is reasonably straight-forward.

The interchange may form new crimps and end-folds in $p_k$. The computation also costs time linear in the size of the 1D map. Every available order inside $p_k$ can be obtained by interchanging these pairs finite times.

### 4.3 Interchangeable Perpendicular Folds

The interchangeable condition of consecutive perpendicular folds is discussed in this section. For convenience, we say that a simple fold $f_a$ *involves* a square $s_{i,j}$ if $O_a(i, j) \neq O_{a-1}(i, j)$. By the condition that $F$ would never let adjacent squares become non-adjacent again, the overlapping of corner squares indicates the order of horizontal and vertical folds. The most crucial factor for the interchangeability of perpendicular folds is whether the folds involve corner squares. Based on this factor and the MV assignment around the corner squares, we classify the interchangeable cases into eight classes as illustrated in Fig. 7 and Fig. 8. Detailed instructions are given in the following paragraphs.

These classes concern partly folded states of the map, where the corner squares may not lay on the corners, and may not maintain their initial relative positions. Three possible relative positions of corner squares are illustrated in the upper row in Fig. 5, where the back of the map is colored gray.

For these eight classes, we have Lemma 4. The proof is given through the following sections.

**Lemma 4.** *(1) Every case satisfying the condition of one of these eight classes is an interchangeable case;*
*(2) Every interchangeable case can be classified into one of these eight classes.*

Before the proof, we introduce a notion called *merge* to explain that our interchangeable condition maintains the boundary overlapping order. A pair $(S_a, O_a)$ is used to describe the folded state of a boundary strip, where $S_a$ is the

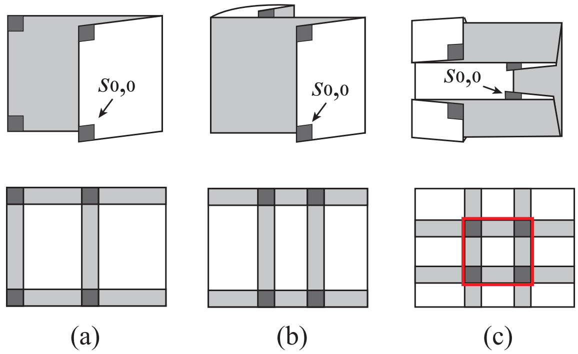


**Fig. 5** The corner squares in boundary areas.

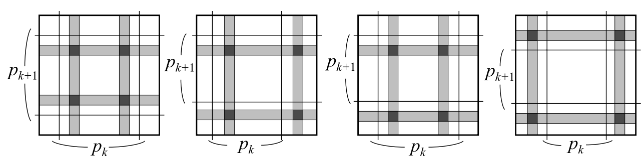


**Fig. 6** Possible relative positions of the corner squares and four end-folds (two are in $p_k$ and the other two are in $p_{k+1}$.)

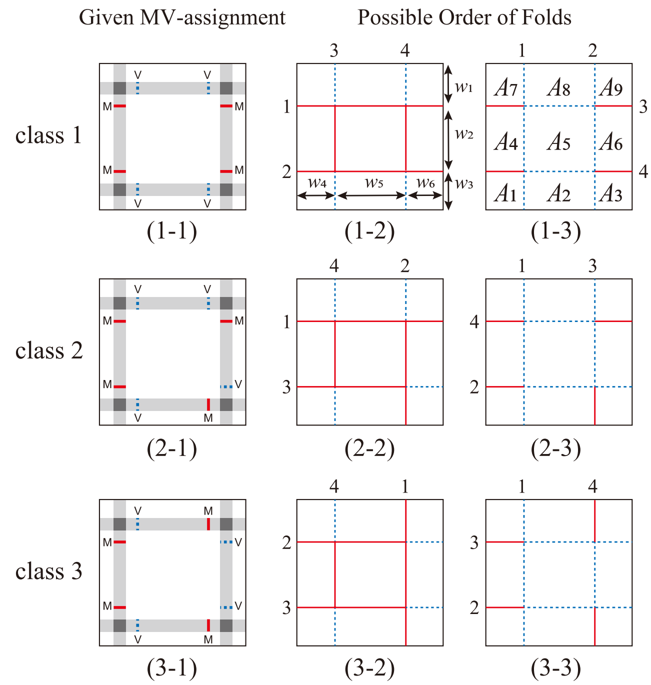


**Fig. 7** The first three cases for four consecutive end-folds.

set containing all the squares on this boundary strip and $O_a$ is the total order on $S_a$, which represents the overlapping of the squares. For two pairs $(S_a, O_a)$ and $(S_b, O_b)$, if a folding sequence induces a new tuple $(S_a \cup S_b, O_c)$ where $O_c$ is an order on $S_a \cup S_b$ (two boundary strips) and satisfies both $O_a$ and $O_b$, then we say that this folding sequence *merges* the two boundary strips.

#### 4.3.1 Classes 1-3

In this section, we discuss the first three classes. In these classes, four possible relative positions of the corner squares and four consecutive end-folds are illustrated in Fig. 6. In the following illustrations, the positions of creases are taken only for convenience. With symmetric cases omitted, there are only three interchangeable MV assignments (illustrated in Fig. 7 (1-1), (2-1) and (3-1)). This is because every two creases around a corner square must be labeled differently so that whether the horizontal crease precedes the vertical crease or not, they always form the same overlapping order. The order can be described as the triple (a square from the same vertical boundary strip as the corner square, the corner square, a square from the same horizontal boundary strip as the corner square) from bottom to top, or in reverse, and that there should exist at least one matching MV assignment to some crease line, to ensure the interchangeability.

If we consider each two parallel end-folds as a pair, then for each class, there exist two possible ways to order the four end-folds given as possible orders in Fig. 7 (all the possible ways to order two parallel folds in a pair are counted as one). The order of folds is indicated by 1, 2, 3, 4. Their interchangeable conditions are presented in Lemma 5. Let $w_1$ to $w_6$ be the widths illustrated in Fig. 7 (1-2).

**Lemma 5.** *(1) Interchangeability: in Classes 1–3 whose MV assignments are illustrated in Fig. 7, when the interchangeable conditions:*
*Class 1. Either $w1 + w3 < w2$ or $w4 + w6 < w5$.*
*Class 2. $w1 + w3 < w2$ and $w4 + w6 < w5$.*
*Class 3. $w1 + w3 < w2$*
*are satisfied, the orders of the four end-folds illustrated as Possible Order of Folds in Fig. 7 lead to the same boundary overlapping order. (2) For the partly folded states as illustrated in Fig. 7, the four end-folds cannot be interchanged when the interchangeable conditions are violated.*

*Proof.* (1) The interchange of folds is feasible if it (a) does not change the adjacent relation $\leftrightarrow$ on respective boundary strips, and (b) involves no merging. The certain MV assignment ensures (a). For (b), we use $A_1$ to $A_9$ to indicate different areas, as shown in Fig. 7 (1-3). We consider the case that the four corner squares are located in $A_1$, $A_3$, $A_7$ and $A_9$ as an example.

The interchangeable conditions avoid the boundary overlapping orders on respective boundary strips being merged in Classes 1 to 3. Correspondingly, in each class, the two possible orders of the four end-folds with the same given MV assignment lead to the same boundary overlapping order on respective boundary strips. Taking Class 1 for example, assume that $A_1$ always faces the front up. The two folding sequences shown in Fig. 7 (1-2) and (1-3) lead to the same overlap order on respective boundary strips, which can be described by the 3-tuples: $(\{A_1, A_7\}, A_4)$, $(A_2, \{A_1, A_3\})$, $(\{A_3, A_9\}, A_6)$ and $(A_8, \{A_7, A_9\})$. Cases of the other two classes are similar and straightforward. Since $A_5$ contains no boundary strip, it does not affect the order of the boundary squares.

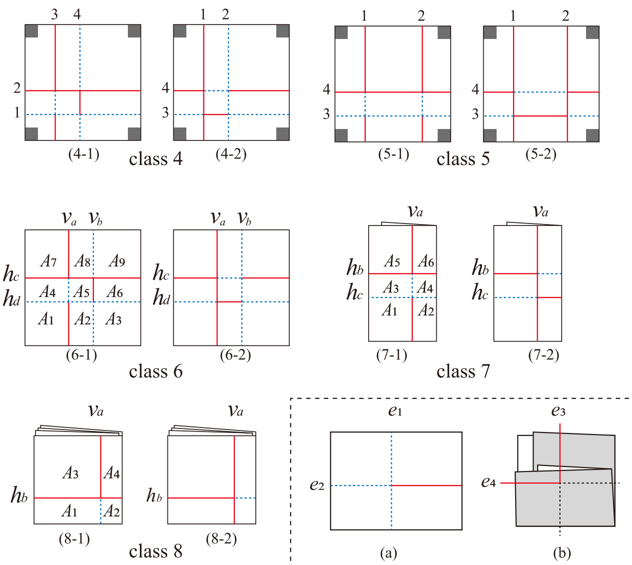


**Fig. 8** Classes 4–8: interchangeable perpendicular folds. For Classes 6–8, the interchangeable condition is the same, that at least one of $f_{i-1}$ and $f_i$ does not involve corner squares.

(2) When the interchangeable conditions are violated, each folding sequence merges the 3-tuples, i.e., the overlapping orders on four boundary strips, to an 8-tuple. For example, Fig. 7 (1-2) would lead to ($A_4$,$A_7$,$A_1$,$A_6$,$A_9$,$A_3$,$A_2$,$A_8$), while Fig. 7 (1-3) would lead to ($A_6$,$A_4$,$A_7$,$A_9$,$A_8$,$A_1$,$A_2$,$A_3$). By this fact, the necessity of the interchangeable conditions in these cases is clear. □

#### 4.3.2 Classes 4 and 5

Classes 4 and 5 are illustrated in Fig. 8. Their description and conditions are detailed as follows.
Class 4. (1, 2) and (3, 4) form pairs of differently labeled parallel end-folds or crimps. Each tuple involves the same two corner squares. 1 and 3 are labeled differently.
Class 5. (1, 2) is a pair of parallel interchangeable end-folds with the same labels. (3, 4) is a pair of differently labeled parallel end-folds or a crimp. 3 and 4 involve the same two corner squares. 1 and 3 are labeled differently.

**Lemma 6.** *We have the conclusion for the interchangeability of Classes 4 and 5: The interchange of the two tuples (1, 2) and (3, 4) in Classes 4 and 5 would not affect the boundary overlapping.*

The proof is omitted since the analysis is similar to the proof for Classes 1 to 3.

#### 4.3.3 Classes 6-8

The last three classes concern two consecutive perpendicular folds $f_{i-1}$ and $f_i$. Note that in these classes, we do not consider the cases already discussed. For these three classes, we have Lemma 7. Figure 8 illustrates these interchangeable cases.

We first give the definition of Classes 6 to 8 as follows.
Class 6. Both $f_{i-1}$ and $f_i$ are crimps.
Class 7. $f_{i-1}$ is an end-fold and $f_i$ is a crimp.
Class 8. Both $f_{i-1}$ and $f_i$ are end-folds

**Lemma 7.** *For Classes 6 to 8, when the following interchangeable condition: at least one of $f_{i-1}$ and $f_i$ does not involve corner squares is satisfied, we have the following conclusions.*
*(1) Interchangeability: in Classes 6 to 8, when the interchangeable condition is satisfied, the interchange of $f_{i-1}$ and $f_i$ in Classes 6 to 8 would not affect the boundary overlapping.*
*(2) Necessity of the interchangeable condition: in Classes 6 to 8, $f_{i-1}$ and $f_i$ cannot be interchanged when the interchangeable condition is violated.*

*Proof.* (1) We first offer a general explanation before providing specific details. Viewing $R_{i-2}$ as a reduced map, it is clear that in each class, the MV assignment induces the same overlapping order on every respective boundary strip. Folds $f_{i-1}$ and $f_i$ are applied on perpendicular boundary strips. At least one of them involves no corner square. Therefore, no overlapping order of a corner square would be simultaneously affected by $f_{i-1}$ and $f_i$. Then, since only one intersection of two boundary strips is a corner square, the overlapping orders on different boundary strips would not be merged by $f_{i-1}$ and $f_i$. Then, we provide a detailed proof for Class 6. $A_1$ to $A_9$ in Fig. 8 (6-1) indicate the areas separated by $v_a$, $v_b$, $h_c$, and $h_d$. Without loss of generality, we assume that $f_{i-1} = \{v_a, v_b\}$, $f_i = \{h_c, h_d\}$, and $s_{0,0}$ is located at $A_1$. The certain MV assignment limits the locations of boundary strips. They must be located at all or some of $A_1$, $A_3$, $A_7$ and $A_9$.

The overlapping order on every boundary strip is uniquely determined, because the MV assignment uniquely determines the overlap on respective boundary strips as ($A_1$, $A_4$, $A_7$), ($A_9$, $A_6$, $A_3$), ($A_3$, $A_2$, $A_1$) and ($A_9$, $A_8$, $A_7$). These tuples are not merged by $f_{i-1}$ and $f_i$. With no boundary square simultaneously involved in two tuples, the interchangeability is proved.

Omitting the symmetric cases, the MV assignments of two possible folding sequences in Classes 7 and 8 are also illustrated in Fig. 8.

(2) We take an instance of Class 8 violating the interchangeable condition to explain the necessity of the condition. As illustrated in the last figure in Fig. 8, state (b) is obtained by folding a single piece of paper along $e_1$ and $e_2$ as illustrated in (a). For the flat-folded state (b), the two orders of end-folds, $(e_3, e_4)$ and $(e_4, e_3)$, induce different $O_i(0, n-1)$s. This means that the two end-folds are not interchangeable. Similarly, the other instances of these three classes violating the interchangeable condition also indicate the uninterchangeable property of the two folds. □

#### 4.3.4 Completion of Proof

In the above three sections, we have concluded the proof of (1) in Lemma 4. We have also proved that the interchangeable conditions we provided for Classes 1 to 8 are necessary. To complete the proof of Lemma 4, we have to prove that every interchangeable case in a partly flat-folded state must belong to one of Classes 1 to 8.

For a pattern with only a pair of perpendicular crease lines, the four creases are definitely assigned either three "M"s and a "V" or three "V"s and an "M". The interchange of the two crease lines would change the labels of two creases, which can also be considered as a prohibited change of assignment on two perpendicular boundary strips. By mathematical induction, the interchange of an odd number of pairs of perpendicular crease lines would definitely change the MV assignment on the boundary strips, while the interchange of an even number of pairs of perpendicular crease lines would maintain the original MV assignment on the boundary strips. It forces every interchange of horizontal folds and vertical folds to be implemented on both an even number of horizontal crease lines and an even number of vertical crease lines. Thus, we consider the interchangeable perpendicular creases of the minimal even number such that any feasible interchange between $p_k$ and $p_{k+1}$ can be obtained by a combination of these minimal interchanges.

The crimps always affect an even number of layers at the initial state, so we consider a pair of perpendicular crimps in Class 6. In a partly flat-folded state, it is possible for an end-fold along a single line to fold an odd number or an even number of layers and to involve an odd number or an even number of crease lines. For the end-folds involving an odd number of crease lines, we consider two consecutive end-folds together to maintain the parity. In this case, all the possible interchangeable cases can be classified to (a) two pairs of end-folds (Classes 1 to 3) and (b) a pair of end-folds and a crimp (Classes 4 and 5). In an intermediate state where the end-fold corresponds to an even number of layers, its relevant possible interchangeable cases can be classified to (a) two end-folds (Class 8) and (b) an end-fold and a crimp (Class 7).

Because all the possible interchangeable cases are considered, Lemma 4(2) is proved. Lemmas 5, 6 and 7 conclude the interchangeable conditions of consecutive perpendicular folds on the eight classes. In the next section, we will put forward a folding process based on these classes.

## 5. Algorithm for the Decision Problem

We first define the equivalence relation on $\mathscr{F}$ and the method to find the representative $F$. Based on the interchangeable conditions, an equivalence relation can be defined as follows: For any two elements $F_1$ and $F_2$ in $\mathscr{F}$, if $F_2$ can be induced from $F_1$ by interchanging the interchangeable folds, then we write $F_1 \sim F_2$. It is straightforward to check that $\sim$ is an equivalence relation.

Note that the number of elements in an equivalent class can be exponential to $m + n$ according to the number of possible interchanges.

We provide a description of the algorithm in what follows and show how to enumerate all the whole valid simple folding sequences. The algorithm itself is shown in Appendix B.

### 5.1 Description of the Algorithm

The algorithm for the decision problem consists of three steps.

(1) First Step: Crimps

We now give the method to find $F$. Referencing Classes 6 and 7, crimps are interchangeable with any end-fold in $R_0$. We use the method introduced in Sect. 4.1 to find the crimps and fold them if they exist. The map can then be reduced. We repeat this process until no more crimp can be found. After folding these crimps, corner squares should still locate at the four corners.

(2) Second Step: End-Folds

The first feasible fold of the reduced map must be an end-fold and must involve corner squares. It is sufficient to determine whether the end-fold is along a horizontal or vertical crease line because based on the results we have for $p_k$, we can decide the feasible fold when its direction is known.

The possible cases of the first end-fold are divided into (i) and (ii). In (i), at least one of the four corners has its two adjacent squares on the same boundary strip. When a corner square is the first or the last element in the overlapping order, we also consider the case as (i). In (ii), the two adjacent squares of any corner square are on perpendicular boundary strips.

For (i), we first consider $s_{0,0}$. Without loss of generality, we assume that $s_{0,0}$ is in $(s_{i,0},\ s_{0,0},\ s_{j,0})$, which indicates that the first end-fold involving $s_{0,0}$ must be vertical. It is necessary to consider the squares adjacent to other corner squares, and here we take $s_{0,n-1}$ first.

We define the *consistency* between two adjacent pairs of squares on parallel boundary strips. If (a) $(s_{i,n-1},\ s_{0,n-1})$ (or $(s_{0,n-1}, s_{j,n-1})$) holds when $n$ is odd; (b) $(s_{0,n-1},\ s_{i,n-1})$ (or $(s_{j,n-1}, s_{0,n-1})$) holds when $n$ is an even, we say that the overlapping of $\{s_{0,n-1},\ s_{i,n-1}\}$(or $\{s_{0,n-1}, s_{j,n-1}\}$) is consistent with $\{s_{0,0},\ s_{i,0}\}$($\{s_{0,0},\ s_{j,0}\}$); if the reverse holds, we say the overlappings of the corresponding pairs are *inconsistent*. When both pairs satisfy the consistency, we say that $\{s_{i,n-1},\ s_{0,n-1},\ s_{j,n-1}\}$ is consistent with $\{s_{i,0},\ s_{0,0},\ s_{j,0}\}$. In this manner, the definition of consistency can further be extended to sets of arbitrary numbers of squares.

Corresponding to different cases of the closure of $s_{0,n-1}$, three valid cases omitting the symmetries are summarized in the first column of Fig. 9. In some cases, a further check on the closures of other corner squares is necessary. The details are as follows:

Case (1). In the overlapping, $\{s_{i,n-1},\ s_{0,n-1},\ s_{j,n-1}\}$ is ordered

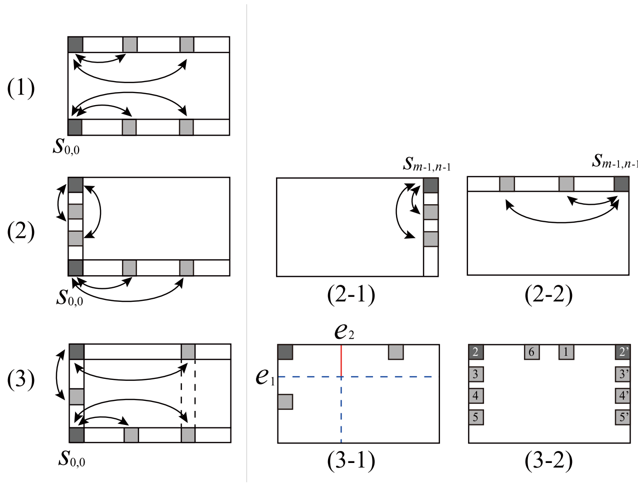


**Fig. 9** Three sub-cases in the analysis of case (i). The arrows indicate the adjacent squares.

consistently with $\{s_{i,0},\ s_{0,0},\ s_{j,0}\}$. In this case, the first end-fold must be vertical. Otherwise, at least one of $s_{0,0}$ and $s_{0,n-1}$ should be adjacent to a boundary square $s_{0,y}$ (for some $y$).

Case (2). The overlapping includes $(s_{0,k},\ s_{0,n-1},\ s_{0,l})$. Then, at least one horizontal fold involving $s_{0,n-1}$ precedes the end-fold involving $s_{0,0}$. Otherwise, $s_{0,n-1}$ should be adjacent to a boundary square $s_{i,n-1}$ or $s_{j,n-1}$ in the overlapping. However, there may exist end-folds involving only $s_{m-1,0}$ and $s_{m-1,n-1}$ that precede this horizontal fold. A further consideration on the closure of $s_{m-1,n-1}$ is necessary. Two of the three possible cases are illustrated in Fig. 9:

(2-1). The adjacent squares of $s_{m-1,n-1}$ are $s_{m-1,k}$ and $s_{m-1,l}$ (in either order). This indicates that horizontal folds precede vertical folds. Otherwise, at least either $s_{0,n-1}$ or $s_{m+1,n-1}$ should be adjacent to a boundary square $s_{x,n-1}$ (for some $x$).

(2-2). If the closure of $s_{m-1,n-1}$ is $(s_{i',n-1},\ s_{m-1,n-1},\ s_{j',n-1})$, then the vertical folds involving $s_{m-1,n-1}$ precede the horizontal folds involving $s_{0,n-1}$. The first end-fold should be vertical. Otherwise, either $s_{0,0}$ should be adjacent to a boundary square $s_{0,y}$ or $s_{m-1,n-1}$ should be adjacent to a boundary square $s_{m-1,y\prime}$ (for some $y$ and $y\prime$).

To avoid repetition, another case (2-3) where adjacent squares of $s_{m-1,n-1}$ belong to different boundary strips, is not discussed here. It can be classified into Case (3) by rotating.

Case (3). The overlapping includes $(s_{a,n-1},\ s_{0,n-1},\ s_{0,b})$ or $(s_{0,b},\ s_{0,n-1},\ s_{a,n-1})$ where $a = i$ or $a = j$. The first one of all the folds involving $s_{0,0}$ and the folds involving $s_{m-1,0}$ must be vertical, otherwise $s_{0,0}$ should be adjacent to a boundary square $s_{0,y}$. However, we also have to consider the folds do not involve $s_{0,0}$ and $s_{m-1,0}$. There are only two cases where the first fold is horizontal.

(3-1). The first fold is horizontal if there exists an adjacent relation on $\{s_{u,n-1},\ s_{v,n-1}\}$ with $u$= 0 or $m-1$, and is inconsistent with $\{s_{u,0},\ s_{v,0}\}$. This means that the folds on the two horizontal parallel boundary strips are not the same. As illustrated in Fig. 9, $e_2$ induces the adjacent relation of $\{s_{0,n-1},\ s_{a,n-1}\}$ and $\{s_{0,0},\ s_{a,0}\}$, and it should thus be folded after $e_1$. The inconsistency corresponds to an odd number of horizontal folds preceding the vertical folds.

(3-2). In the overlapping, if there exists $S=(s_{u,n-1},\ s_{0,a_1},\ s_{0,a_2}, \ldots, s_{0,a_k},\ s_{v,n-1})$ $(k > 2)$ including $s_{0,n-1}$ as the second or penultimate element in this tuple (as the sequence of squares (1, 2, 3, 4, 5, 6) in 3-2 of Fig. 9), and there also exists a closure on $\{s_{m-1,a_1},\ s_{m-1,a_2}, \ldots, s_{m-1,a_k}\}$ such that the ordered set is consistent with this sub-sequence (as (2', 3', 4', 5')), then there exist an even number of horizontal folds preceding the vertical folds. $k > 2$ ensures that the number of horizontal folds is at least two. These horizontal folds are either folded before the vertical folds, or would form a zigzag for each corner square with the vertical folds in the folded state. The zigzag case corresponds to Classes 4 and 5 and thus the horizontal fold can be applied first by interchangeability. This case corresponds to an even number of horizontal folds preceding the vertical folds.

The above analysis exhausts both cases where the proceeding horizontal folds are an odd number and an even number. In all other cases, it is determined that the first end-fold is vertical.

(ii) is the case that each corner square has its two adjacent squares on perpendicular boundary strips. The closures of the four corner squares (we consider each corner square with its adjacent two squares as the closure of the corner square here) are formed by at least four end-folds. Fig. 7 shows all the instances where the closures of the four corner squares are formed by four consecutive end-folds. Each end-fold assigns new adjacent squares to the pair of corner squares it involves.

At the beginning of Sect. 5.1(2), we explained that it is sufficient to identify the direction of the first end-fold because we can leave the task of finding legal end-folds to the procedure introduced in Sect. 4.1. When more than four end-folds form the closures of the four corner squares, there must exist some end-folds which only assign one or even no new adjacent square to the pair of corner squares it involves. Because the first end-fold must assign new adjacent squares to two corner squares, it is not possible for these end-folds to be the first fold. Correspondingly, by checking the consistency of adjacent relations of pairs of parallel corner squares, at most four end-folds can become candidates for the first fold. Here, we only discuss the case where the first four end-folds form all the closures of the four corner squares. To form all the closures, every two creases around a corner square must be labeled differently. Thus Fig. 7 (1-1), (2-1), (3-1) exhaust all the possible cases of the MV assignments.

The first check concerns whether the two parallel pairs of folds respectively have unique orders. If the case matches with the given MV assignments in Fig. 7, then there exist two choices to order these end-folds. We choose an arbitrary one. Otherwise, the order of the four folds can be uniquely decided by the overlapping order of boundary strips. Their MV assignments are the same as Classes 1 to 3 in Fig. 7, whereas the interchangeable conditions are not satisfied. As given in the proof of Lemma 5, the known closures on the boundary strips uniquely decide the order of these four end-folds.

When the direction of the current fold is decided, the first end-fold can be determined using the method introduced in Sect. 4.1. After the first end-fold, we have to find the crimps before applying other end-folds because some boundary squares may change the upward side and form new crimps.

(3) Third Step: Folds on Partly Folded States

After the steps introduced above, there may exist some crimps involving corner squares and some end-folds involving no corner squares. According to Classes 6 to 8, if there are some folds only influencing the overlapping outside the rectangle formed by the four corner squares (as illustrated by the red line segments in the last figure in Fig. 9 (3)), we first apply these folds.

Next is a repeating process of the check as for $R_0$ until the map is reduced to a state where either pair of parallel boundary strips overlap each other. When deciding the order of perpendicular folds, if they form the cases in Classes 1 to 8, we can order the folds as any of the orders introduced in Sect. 4.3. Otherwise, the order of the folds is always uniquely determined by the overlapping of the boundary strips. This decision is similar to the analysis in the proof of Lemma 5, just with crimps included.

The above checking process ends when a pair of parallel boundary strips totally overlap each other. This state can be viewed as a 1D map and then handled with the method in Sect. 4.1. If $O$ is a valid boundary overlapping order in the simple fold model, we can finally reduce the map to a $1 \times 1$ size. A corresponding representative folding $F$ is composed of the uniquely decided folds and the arbitrary choices for interchangeable folds.

### 5.2 Analysis of Computational Complexity

Referencing our algorithm in Appendix B, making the decision costs $O(m + n)$ time. The computation of the MV assignment on the boundary strips costs $O(m+n)$ time [8]. By a traverse of $O$, the input order of all the boundary squares can be saved as a directed graph $G$. Each node represents a square and each edge represents an adjacent relation. The initial weight of each edge is one. During the folding process, once an adjacent relation is induced by the folds we find, we reassign the corresponding edge weight to zero. The reduction of the map is indicated by the decrease in the values of the edges. A valid final state $R_t$ corresponds to $G$ with a total weight of zero on all the edges. At each step, the new neighbor squares can be found by checking the value of the path connecting two nodes.

The search for the same crimps and end-folds on the two parallel boundary strips needs $O(m)$ ($O(n)$) time for horizontal (vertical) crimps. During the construction of $G$, by assigning zeros to edges, newly developed adjacency between squares is uniquely recorded. Because the closure

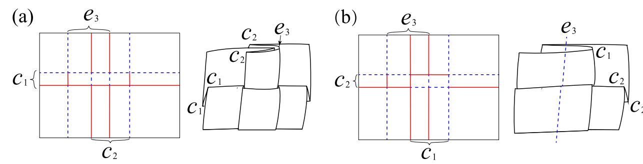


**Fig. 10** A possible lock induced by Step 3. $e_3$ is an end-fold after $c_1$ and $c_2$. The interchange of $c_1$ and $c_2$ makes a later fold $e_3$ become unavailable.

$\leftrightarrow^*$ finally forms a Hamiltonian path in $G$, which involves $O(m+n)$ edges, the total cost pf this search is $O(m+n)$ time.

Each time we decide the direction of the end-fold, checking for closures of the four corner squares costs constant time. During the reduction of the map based on the determined folds, every adjacent relation is considered in the computation only once. Thus, the total cost of ordering the end-folds and crimps is $O(m + n)$ time.

### 5.3 Extension to Enumeration Algorithm

In this section, we give an extension to enumerate valid whole simple folding sequences induced from $F$, which requires the following steps.

S1. Compute $P = (p_1, p_2, \ldots, p_l)$ by grouping the parallel folds in $F$;
S2. Find all the interchangeable pairs of folds in each $p_i$;
S3. Record all the different folding sequences obtained by interchanging the pairs found in Step 2;
S4. For each sequence obtained in S3, output the sequence as a valid whole simple folding sequence; then, find the perpendicular interchangeable folds in the sequence.
S5. Record all the different simple folding sequences by interchanging the folds checked in S4 and identify their feasibility. Some folds would form a lock (see Fig. 10 for a lock). If there exists no such lock, output the sequence and then reapply Steps 1 to 4.

## 6. Conclusion and Future Work

In this work, the decision problem of the validity of the boundary overlapping orders in $m \times n$ maps in the simple fold model has been solved. We showed how to recognize the order of parallel and perpendicular folds, with the interchangeable condition for consecutive folds as an equivalence relation. The validity of the boundary overlapping order is determined by the existence of a representative folding via a reduction of the map. We then provide an extension to enumerate other valid whole simple folding sequences. The computational complexity of the enumeration and other cases of partial orders are interesting open problems.

### References

[1] Y. Jia, J. Mitani, and R. Uehara, "Valid orderings of layers when simple-folding a map," J. Information Processing, vol.28, pp.816–824, 2020.

[2] M. Bern and B. Hayes, "The complexity of flat origami," Ann. ACM-SIAM Symposium on Discrete Algorithms, pp.175–183,

ACM, 1996.
[3] H.A. Akitaya, K.C. Cheung, E.D. Demaine, T. Horiyama, T.C. Hull, J.S. Ku, T. Tachi, and R. Uehara, "Box pleating is hard," Japanese Conference on Discrete and Computational Geometry and Graphs, pp.167–179, Springer, 2015.
[4] E.D. Demaine and J. O'Rourke, Geometric Folding Algorithms, Cambridge University Press, Cambridge, 2007.
[5] E.M. Arkin, M.A. Bender, E.D. Demaine, M.L. Demaine, J.S.B. Mitchell, S. Sethia, and S.S. Skiena, "When can you fold a map?," Computational Geometry, vol.29, no.1, pp.23–46, 2002.
[6] T. Morgan, "Map folding," master thesis, Massachusetts Institute of Technology, 2012.
[7] H.A. Akitaya, E.D. Demaine, and J.S. Ku, "Simple folding is really hard," J. Information Processing, vol.25, pp.580–589, 2017.
[8] R.I. Nishat, "Map folding," master thesis, University of Victoria, 2013.
[9] J.E. Koehler, "Folding a strip of stamps," J. Combinatorial Theory, vol.5, no.2, pp.135–152, 1968.
[10] R. Uehara, "Stamp foldings with a given mountain-valley assignment," Origami 5, pp.599–612, AK Peters/CRC Press, 2016.
[11] X. Tan and M. Gruninger, "The complexity of partial-order plan viability problems," Twenty-Fourth International Conference on Automated Planning and Scheduling, 2014.
[12] J. Justin, "Towards a mathematical theory of origami," International Meeting of Origami Science and Scientiffic Origami, pp.15–29, K. Miura, ed., 1996.
[13] K. Kasahara and T. Takahama, Origami for the Connoisseur, Japan Publications, 1987.
[14] T. Kawasaki, "On the relation between mountain-creases and valley-creases on a flat origami," International Meeting of Origami Science and Technology, pp.229–237, H. Huzita, 1989.
[15] B. Baumslag and B. Chandler, Schaum's Outline of Theory and Problems of Group Theory McGraw-Hill, New York, 1968.

## Appendix A: An Instance of the Computation of Crimps and End-Folds in a 1 × 10 Map

We use numbers 0 to 9 to label the squares of a $1 \times 10$ map $M$ from its left end to its right end. The input order $O$ is given as (9, 2, 3, 4, 1, 0, 5, 8, 7, 6). The objective is to determine whether $O$ is a valid overlapping order of $M$ in the simple fold model, and if it is, a feasible whole simple folding sequence is also desired.

The process of computation is as follows: First, the MV assignment of $M$ is computed using the method introduced in [8], as illustrated in Fig. A· 1(a). Then, by a traverse of the map, all the pairs of neighbor squares are recorded. In this instance, they are {2, 3}, {3, 4}, {1, 0}, {8, 7}, {7, 6}. Except for the creases between every pair, all other creases are removed from the map, which forms a new map as illustrated in Fig. A· 1(b).

Next, by locally considering every single crease and every pair of nearest creases, the candidates of crimps and end-folds are the crimp forming (2, 3, 4), the crimp forming (8, 7, 6), and the end-fold forming (1, 0). For the crimps, referencing the equation provided in Sect. 4.1, the existence of the adjacent relation ($s_x$, $s_{2x_B-x-1}$, $s_{x+2x_C-2x_B}$) or its reverse should be checked. For (2, 3, 4), $x$ =2, $x_B$ =3 and $x_C$ =4, satisfying the equation and thus the determination of the first fold is made. The check for (8, 7, 6) proceeds in the same way. Similarly, the existence of ($s_x$, $s_{2x_G-x-1}$) or its reverse

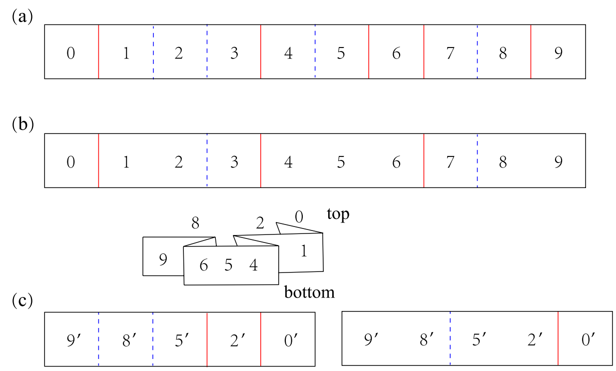


**Fig. A· 1** Computing the orders of folds an instance 1D map.

should be checked for the end-fold. For (1, 0), $x$=0 and $x_G$=1, the reverse ($s_{2x_G-x-1}$, $s_x$) satisfies the equation and indicates that this end-fold is firstly folded. The folded state is shown in Fig. A· 1(b). These three folds are disordered. After folding them, the map is reduced to a new map, as illustrated in Fig. A· 1(c). For convenience, we use the labels of the top layers to indicate the squares in the new map. $O$ is correspondingly updated to (9', 2', 0', 5', 8'). The MV assignment is changed along with the change of the side that these squares face up.

Two candidate end-folds in the new map are decided as that corresponding to (2', 0') and that corresponding to (5', 8'). (2', 0') satisfies the equation and is thus folded first in this step while (5', 8') does not satisfy the equation and is checked in the new map after folding the end-fold of (2', 0'). (5', 8') satisfies the equation in the new map and is then folded. As per Sect. 5.2, using the graph structure in the realization can reduce the check for every fold only once.

Following a similar process, we can finally determine the validity of $O$ and obtain the folding process as follows: (the creases are indicated by neighbor squares) ({$c_1$=(2, 3, 4), $c_2$=(6, 7, 8), $e_1$=(0, 1)}, $e_2$=(1, 2), $e_3$=(5, 6), $e_4$={(4, 5), (8, 9)}). The folds in the same brace can be arbitrarily ordered.

## Appendix B: Algorithm Description

```
Input  : A total order O on the set of boundary
         squares//indicating 2m + 2n − 4 numbered squares
Output: A Boolean value // the validity of O in the simple fold
         model
Begin
    initialization
    G ← A directed graph with 2m + 2n − 4 nodes // the folded
      state of the map
    F ← ∅ // the folds applied to the map in order
    Compute the MV assignment on the boundary strips
      according to O
    if |F| = 0 then
        Find the first horizontal and vertical folds (Sect. 4.1)
        if no feasible folds can be found then
            return false // O is invalid
        end
        while crimp exists do
            Append the crimps to F, update G by assigning
              zero to the edges incident to the new adjacent
              pairs of nodes induced by these crimps
        end
        Find End-Folds(F, G)
    end
    while no parallel boundary strips totally overlap each
      other according to G do
        Find the crimps or end-folds involving no corner
          squares in the current state of the map indicated by
          G, append them to F, update G by assigning zero to
          the edges incident to the new adjacent pairs
        Find End-Folds(F, G)
    end
    while not all the edges in G are assigned zero do
        Find out the folds involving no corner squares, append
          them to F, update G
        Find the next crimps or end-folds according to the
          adjacent squares of the corner squares in the current
          map
        if no feasible folds can be found then
            return false // O is invalid
        else
            Decide the unique order of folds, append them to
              F, update G
        end
        return true
    end
End
Function Find End-Folds(F, G):
    Check the closure of the adjacent relation of corner squares
    if Classes 1–3 exists then
        Decide the first four consecutive end-folds. Assign
          either of the corresponding feasible order to them
          and append them to F, update G
    else
        if the first end-fold can be uniquely decided or is
          arbitrarily decided by the interchangeability of
          Classes 4 and 5 then
            Append the first end-fold to F, update G and
              check for crimps
            if crimps exist in the current state then
                Append the crimps to F, update G
            else
                Append the next end-folds to F, update G
            end
        else
            return false // O is invalid
        end
    end
End Function
```

**Algorithm 1:** Algorithm of the decision problem of the boundary overlapping order

**Yiyang Jia** is a Ph.D. student at the Graduate School of Systems and Information Engineering, University of Tsukuba. Her research mainly focuses on computational origami, especially map folding problems.

**Jun Mitani** received his Ph.D. in engineering from the University of Tokyo in 2004. He has been a professor of the University of Tsukuba since April 2015. His research interests center on computer graphics, especially geometric modeling techniques. He studies the geometry of curved origami as well as interactive design interfaces.

**Ryuhei Uehara** is a professor in School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He received B.E., M.E., and Ph.D. degrees from University of Electro-Communications, Japan, in 1989, 1991, and 1998, respectively. He was a researcher at CANON Inc. during 1991–1993. In 1993, he joined Tokyo Woman's Christian University as an assistant professor. He was a lecturer during 1998–2001, and an associate professor during 2001–2004 at Komazawa University. He moved to JAIST in 2004. His research interests include computational complexity, algorithms and data structures, and graph algorithms. He is especially engrossed with computational origami, games and puzzles from the viewpoints of theoretical computer science. He is a member of EATCS and the chair of EATCS Japan Chapter.