IEICE TRANSACTIONS

on Fundamentals of Electronics, Communications and Computer Sciences

DOI:10.1587/transfun.2024DMP0004

Publicized:2025/02/03

This advance publication article will be replaced by the finalized version after proofreading.

A PUBLICATION OF THE ENGINEERING SCIENCES SOCIETY

The Institute of Electronics, Information and Communication Engineers

Kikai-Shinko-Kaikan Bldg., 5-8, Shibakoen 3chome, Minato-ku, TOKYO, 105-0011 JAPAN

PAPER

Computational Complexity of One-Dimensional Origami with Constraints on Thickness at Creases*

Junnosuke HOSHIDO†, **Tonan KAMATA**††a), *Nonmembers*, **Tsutomu ANSAI**†††b), *and* **Ryuhei UEHARA**††c), *Members*

SUMMARY We investigate the computational complexity of a simple one-dimensional origami problem. We are given a paper strip P of length $n+1$ and fold it into unit length by creasing at unit intervals. Consequently, we have several paper layers at each crease in general. The number of paper layers at each crease is called the crease width at the crease. For a given mountain-valley assignment of P, in general, there are exponentially many ways of folding the paper into unit length consistent with the assignment. It is known that the problem of finding a way of folding P to minimize the maximum crease width of the folded state is NP-complete. In this study, we investigate a related paper-folding problem. For any given folded state of P, each crease has its mountain–valley assignment and crease-width assignment. Then, can we retrieve the folded state uniquely when only partial information about these assignments is given? We introduce this natural problem as the crease-retrieve problem, for which there are a number of variants depending on the information given about the assignments. In this paper, we show that some cases are polynomial-time solvable and that some cases are strongly NP-complete.
key words: *computational origami, crease-retrieve problem, crease width, NP-complete, one-dimensional origami.*

1. Introduction

Recently, computational origami has attracted the interest of theoretical computer scientists. In this paper, we focus on one of the simplest origami models: one-dimensional origami. This origami model involves a long rectangular strip of paper, which can be abstracted by a line segment and is uniformly subdivided by creases. At each crease, we fold the paper strip by degree π in either one of two choices for the direction of folding: a mountain fold, or a valley fold. Finding the number of feasible (i.e., without self-crossing) ways of folding a paper strip is known as a stamp-folding problem, for which the exact value remains open [1]: Experimentally, a paper strip of length $n+1$ has a total of $\Omega(3.06^n)$ feasible ways of folding and, on average, $\Omega(1.53^n)$ ways of folding for a given random mountain–valley assignment ("MV assignment," for short) of length n.

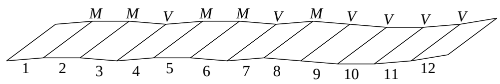


Fig. 1 Example of MV assignment $MMVMMVMVVVV$ for paper strip of length 12.

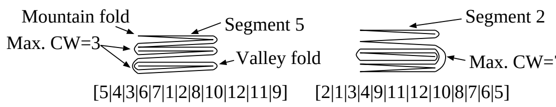


Fig. 2 Side views of two folded states for MV assignment $MMVMMVMVVVV$. [5|4|3|6|7|1|2|8|10|12|11|9] and [2|1|3|4|9|11|12|10|8|7|6|5] describe the orders of paper segments from the top. The first folded state has the maximum crease width of 3, whereas the second has the maximum crease width of 7.

It is known that a paper strip has a unique folded state if and only if it is a pleat folding with MV assignment $MVMVMVMVM\cdots$ or $VMVMVMV\cdots$ [1]. Except the pleat folding, little is known about the relationship between an MV assignment and the number of feasible folded states for the assignment. For example, a paper strip of length 12 with the MV assignment $MMVMMVMVVVV$, shown in Fig. 1, has 100 different feasible folded states (as verified by a computer program), among which some are easy, while some are difficult, to fold flat. The main reason behind the differences in difficulty is the number of paper layers between two paper segments at each crease. For example, in the first folded state shown in Fig. 2, the maximum number of layers at a crease is 3, whereas in the second folded state, the maximum number of layers is 7. From this viewpoint, an optimization problem was proposed and investigated in [2]. That paper introduced a new concept known as the "crease width" of a crease, which is defined by the number of paper layers at a crease in a folded state. Therein, it was proved that the minimization problem for the maximum crease width of a given MV assignment is NP-complete. (In fact, among the 100 feasible folded states for the MV assignment $MMVMMVMVVVV$ shown in Fig. 1, the first folded state is the only one with a maximum crease width of 3,

†FLECT Co.,Ltd.
††Computing Science Research Area, Japan Advanced Institute of Science and Technology
†††National Institute of Technology (KOSEN), Ibaraki College
*A part of this paper was presented at the 34th Canadian Conference on Computational Geometry (CCCG 2022) in August at Toronto, Canada. A part of this research is supported by JSPS KAKENHI Grant Numbers JP20H05961, JP20H05964, JP23K22694, and JP24H00690.
a) E-mail: kamata@jaist.ac.jp
b) E-mail: ansai@gm.ibaraki-ct.ac.jp
c) E-mail: uehara@jaist.ac.jp

which is optimal.)

Now, we consider the information necessary for specifying a folded state. We will observe that given both an MV assignment and a crease-width assignment for every crease ("CW assignment," for short), the folded state is uniquely determined if it is feasible. Then, what happens if we are given partial information about these assignments? This natural question leads us to our new computational origami problem, which is named the crease-retrieve problem. In this paper, we first show that the crease-retrieve problem is strongly NP-complete in general. More specifically, when we are given part of the MV assignment and CW assignment, the decision problem that asks whether there exists a feasible folded state is strongly NP-complete. Even if the entire MV assignment is given, the crease-retrieve problem is still strongly NP-complete when only a part of the CW assignment is given. On the other hand, we also investigate the cases that the crease-retrieve problem is tractable. When the maximum crease width is 0, it is easy to solve the crease-retrieve problem since only the pleat folding satisfies this condition. We extend this idea and show that the crease-retrieve problem can be solved in linear time when the maximum crease width is restricted to 1, 2, or 3.

2. Preliminaries

Herein, a *paper strip* refers to a one-dimensional line segment with creases at every integer position. (In other words, we ignore the thickness and width of the paper.) The paper strip is rigid except at the creases; that is, we are allowed to fold only along these creases at integer positions. We are given a paper strip of length $n+1$ placed in the interval $[0, n+1]$. (We will refer to this state as an *initial state*.) We call each paper segment between i and $i+1$ at the initial state the *segment* $i+1$. We assume that the top and bottom sides of the 1st segment are fixed. The paper strip is in a *folded state* if each crease is folded by a degree π or $-\pi$, and the folded strip is placed in the interval $[0, 1]$. The paper strip is *mountain (valley)-folded* at a crease i when the ith segment and the $(i+1)$st segment are folded in the direction such that their bottom sides (top sides, respectively) are close to touching (although they may not necessarily touch if they have some other paper layers between them). For a given paper strip, an *MV assignment* at crease i is either M or V, where M refers to a "mountain fold," and V refers to a "valley fold." We will use the standard notation x^k for string repetition. For example, $(MV)^3MM(VM)^3 = MVMVMVMMVMVMVM$. A folded state is *feasible* if the paper strip does not penetrate itself in the given state.

We then provide formal definitions of feasibility and MV assignment for the sake of precision. When we obtain a folded state of P placed in the interval $[0, 1]$, the segments $1, 2, \dots, n, n+1$ are positioned in this interval in some proper order. We define an ordering function f such that $f(i) = j$ denotes that the segment i is the jth layer in the folded state with $1 \le i, j \le n+1$. (That is, for the first folded state [5|4|3|6|7|1|2|8|10|12|11|9] shown in Fig. 2, we have $f(1) = 6$, $f(2) = 7$, $f(3) = 3$, $f(4) = 2$, $f(5) = 1$, and so on.) Then, for each i with $1 \le i \le n$, the crease i (between segment i and $i+1$) is mountain-folded in the folded state if and only if (1) i is odd, and $f(i) < f(i+1)$, or (2) i is even, and $f(i) > f(i+1)$. Inversely, the crease i is valley-folded if and only if (3) i is odd, and $f(i) > f(i+1)$, or (4) i is even, and $f(i) < f(i+1)$. When the paper strip does not penetrate itself, the creases form a nest structure. Precisely, a folded state is *feasible* if and only if for any pair of integers i and j ($i \ne j$) with the same parity,[†] we have either

- $\max\{f(i), f(i+1)\} < \min\{f(j), f(j+1)\}$ (crease i is over j),
- $\max\{f(j), f(j+1)\} < \min\{f(i), f(i+1)\}$ (crease j is over i),
- $f(i) < f(j) < f(j+1) < f(i+1)$, $f(i) < f(j+1) < f(j) < f(i+1)$, $f(i+1) < f(j) < f(j+1) < f(i)$, $f(i+1) < f(j) < f(j+1) < f(i)$ (crease i pinches j), or
- $f(j) < f(i) < f(i+1) < f(j+1)$, $f(j) < f(i+1) < f(i) < f(j+1)$, $f(j+1) < f(i) < f(i+1) < f(j)$, or $f(j+1) < f(i) < f(i+1) < f(j)$ (crease j pinches i).

(Consequently, the ith and jth creases should cross when we have $f(i) < f(j) < f(i+1) < f(j+1)$ or its symmetric cases, which means that the paper strip penetrates itself.)

For a given paper strip P of length $n+1$, we consider a feasible folded state. Then, the *crease width* at crease i in the state is defined by $|f(i) - f(i+1)| - 1$, which gives the number of paper layers between the ith segment and the $(i+1)$st segment joined at the crease i in the state. For a feasible folded state, the *CW assignment* at a crease i is defined by the crease width at the crease i.

In this study, we introduce the following *crease-retrieve problem*. We are given partial information on the MV and CW assignments of the creases of a folded state of P. Then, the solution to the problem is a folded state of P that satisfies these assignments. Precisely, the input of the crease-retrieve problem is composed of two functions $\mathrm{As} : [1, n] \to \{M, V, *_A\}$ and $\mathrm{Cw} : [1, n] \to \{0, 1, \dots, n-1, *_C\}$. Intuitively, the symbols $*_A$ and $*_C$ are so-called "wild cards", which mean they can take any value. That is, $*_A$ means "M or V", and $*_C$ means "any integer in $[0, n-1]$". (Note that $\mathrm{Cw}(i)$ takes an integer in $[0, n-1]$ for any $1 \le i \le n$ in a folded state.) The problem asks if there exists a feasible folded state of P consistent with these two functions. Precisely, a folded state satisfies these two functions if and only if for each crease i with $1 \le i \le n$, (1) it is mountain-folded if $\mathrm{As}(i) = M$ or $\mathrm{As}(i) = *_A$, (2) it is valley-folded if $\mathrm{As}(i) = V$ or $\mathrm{As}(i) = *_A$, and (3) the crease width at i is equal to $\mathrm{Cw}(i)$ or $\mathrm{Cw}(i) = *_C$.

3. Computational Complexity of Crease-Retrieve Problem

In this section, we investigate the computational complexities

[†]They satisfy the parenthesis theorem.

for some natural cases of the crease-retrieve problem. We first consider a few trivial cases:

Observation 1: ([1, Proposition 1]) All instances of the crease-retrieve problem are yes instances when $\mathrm{As}(i) \in \{M, V\}$ and $\mathrm{Cw}(i) = *_C$ for every i in $\{1, 2, \ldots, n\}$.

Proof. Intuitively, we can repeat "end folding" for each $i = 1, 2, \ldots, n$ following $\mathrm{As}(i)$. See [3] for the definition of the end folding. In our context, we just repeat folding along the leftmost crease line. It is easy to observe that it is feasible for any function As. □

Observation 2: We can solve the crease-retrieve problem in linear time when $\mathrm{Cw}(i) \in \{0, 1, \ldots, n-1\}$ and $\mathrm{As}(i) \in \{M, V\}$ for every i in $\{1, 2, \ldots, n\}$.

Proof. We first fix segment 1 of *height* 0, where the height $h()$ indicates the order of each paper segment in $[0, 1]$ in the final folded state. (We denote the height of segment 1 by $h(1) = 0$. For example, we have $h(5) = -5$ and $h(8) = 2$ for the left figure in Fig. 2.) Then, for each $i = 1, \ldots, n$, we can compute the height of the segment $i + 1$ from the height of the segment i by adding or subtracting $\mathrm{Cw}(i)$. The addition or subtraction is determined by the parity of i and $\mathrm{As}(i)$. Precisely,

(1) $h(i) = h(i-1) + \mathrm{Cw}(i) + 1$ if i is odd and $\mathrm{As}(i) = V$,
(2) $h(i) = h(i-1) + \mathrm{Cw}(i) + 1$ if i is even and $\mathrm{As}(i) = M$,
(3) $h(i) = h(i-1) - (\mathrm{Cw}(i) + 1)$ if i is odd and $\mathrm{As}(i) = M$, or
(4) $h(i) = h(i-1) - (\mathrm{Cw}(i) + 1)$ if i is even and $\mathrm{As}(i) = V$.

After computation of the heights, we check if the folded state is feasible, and if the heights have no gaps. The folded state has no gap if and only if there is an integer j with $j \leq 0$ such that there exists exactly one paper segment of height j' for every $j' = j, j+1, \ldots, j+n$. This consecutiveness check of heights can be done in linear time in the same technique as in bucket sort. The feasibility can be confirmed through checks of the nest structure. It is discussed in [4, Sect. 3.2.3] in the context of recognition of valid linear orderings in 2D map folding. Using the technique in [4, Sect. 3.2.3], it can be confirmed in linear time. □

Now, we turn to the main theorem in this section.

Theorem 3: The crease-retrieve problem is strongly NP-complete when $\mathrm{Cw}(i) \in \{0, 1, \ldots, n-1, *_C\}$ and $\mathrm{As}(i) \in \{M, V\}$ for every i in $\{1, 2, \ldots, n\}$.

Proof. It is easy to see that the problem is in NP. We prove the hardness via a reduction from the following problem 3-Partition, which is known to be strongly NP-complete even if B is bounded from above by some polynomial in m [5].

<u>3-Partition</u>
Input: Positive integers $a_1, a_2, a_3, \ldots, a_{3m}$ such that $\sum_{j=1}^{3m} a_j = mB$ for some positive integer B and $B/4 < a_j < B/2$ for $1 \leq j \leq 3m$.
Question: Is there a partition of $\{1, 2, \ldots, 3m\}$ into m subsets $A_1, A_2, \ldots, A_m$ such that $\sum_{j \in A_k} a_j = B$ for $1 \leq k \leq m$?

To begin with, we describe a construction of a paper strip P for a given instance $a_1, \ldots, a_{3m}$ and B of 3-Partition. The basic idea is slightly similar to the one in [2].

The strip P consists of a *folder part* and $3m$ *gadget parts* (Fig. 3). The folder part consists of creases in $[1, 2m+3]$, and each of the $3m$ gadget parts corresponds to a_j $(1 \leq j \leq 3m)$, which contains $4m + 28m^2 a_j$ consecutive points on the strip. That is, the total length of P is $2m+3+\sum_{j=1}^{3m}(4m+28m^2 a_j) = 3 + 2m + 12m^2 + 28m^3 B$. In the folder part, creases i with $1 \leq i \leq 2m+3$ form a zig-zag pattern via the MV assignment $VMVM \cdots MV$, as shown in Fig. 3. Precisely, $\mathrm{As}(i) = V$ for odd i, and $\mathrm{As}(i) = M$ for even i. For even i, we let $\mathrm{Cw}(i) = 0$; that is, we cannot have any paper layers in the folded state at this crease (assigned M). For $i = 1$ and $i = 2m + 3$, we set $\mathrm{Cw}(i) = *_C$; that is, we can have any number of paper layers in the folded state at these creases. These two creases 1 and $2m + 3$ are called *trash folders*, where we will put useless paper layers. For each i with $i = 3, 5, 7, \ldots, 2m + 1$, we set $\mathrm{Cw}(i) = 14m^2 B + 6m$. We call these m creases "unit folders."

Now, we move to the gadget part (Fig. 4). For each integer a_j, we let $b_j = 14m^2 a_j$. We first consider the case that j is an odd number. Then, the jth gadget part consists of a zig-zag pattern of length $2m+b_j$ (which can be represented by $(VM)^{b_j+2m}$ in a standard representation of string). Let s_j be the first crease of the jth gadget part (which depends on $a_{j'}$ with all $j' < j$). Then, $\mathrm{As}(i) = V$ for even $i = s_j+2k$, and $\mathrm{As}(i) = M$ for odd $i = s_j + 2k + 1$, with $0 \leq k \leq m + b_j/2$ (we note b_j is even). This zig-zag pattern contains three parts. We set their crease widths as follows: (1) $\mathrm{Cw}(i) = *_C$ for $i = s_j + k$ for $0 \leq k \leq 2m$, (2) $\mathrm{Cw}(i) = 0$ for $i = s_j + k$ for $2m < k < 2m + b_j$, and (3) $\mathrm{Cw}(i) = *_C$ for $i = s_j + k$ for $2m + b_j \leq k < 2m + b_j + 2m$. We call the first and third parts *spring parts* and the second part b_j *part*. Based on the requirement in (2), we cannot put any paper layers at the creases in the b_j part. Intuitively, this part can be considered as "glued," and this thickness of b_j should be put into some folder. On the other hand, each of the spring parts can be split in any way, and they can be put into any folders, including trash folders.

We next consider the case that j is an even number. The zig-zag pattern $(MV)^{b_j+2m}$ is obtained via flipping of the M and V used in the odd case. The crease widths are identical: (1) $\mathrm{Cw}(i) = *_C$ for $i = s_j + k$ for $0 \leq k \leq 2m$, (2) $\mathrm{Cw}(i) = 0$ for $i = s_j + k$ for $2m < k < 2m + 2b_j$, and (3) $\mathrm{Cw}(i) = *_C$ for $i = s_j + k$ for $2m + 2b_j \leq k < 2m + b_j + 2m$.

The construction of the paper strip P can be done in polynomial time. Therefore, it is sufficient to show that P can be folded into a unit length without penetration such that each crease i satisfies the condition for the crease width $\mathrm{Cw}(i)$ if and only if the instance of 3-Partition is a yes instance.

We first observe that most parts of P are in *pleat folding* $MVMV\cdots$ or $VMVM\cdots$. As shown in Fig. 5, the folder part consists of m unit folders of crease width $14m^2B + 6m$

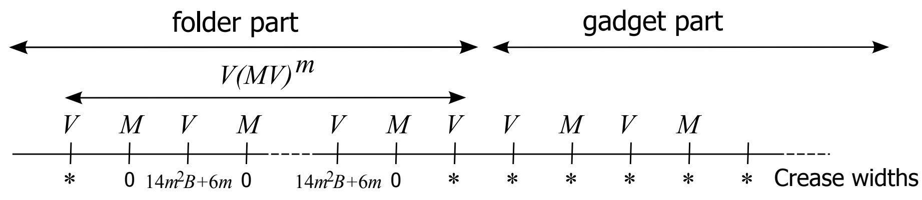


Fig. 3 Construction of paper strip.

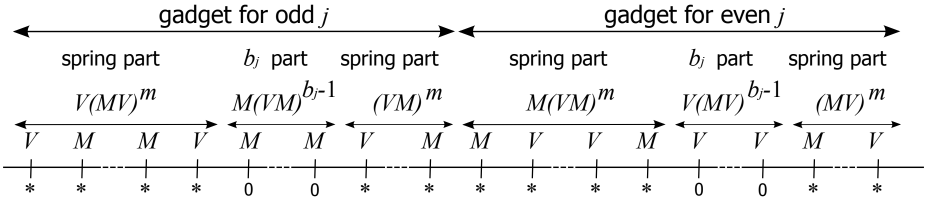


Fig. 4 Construction of gadget part.

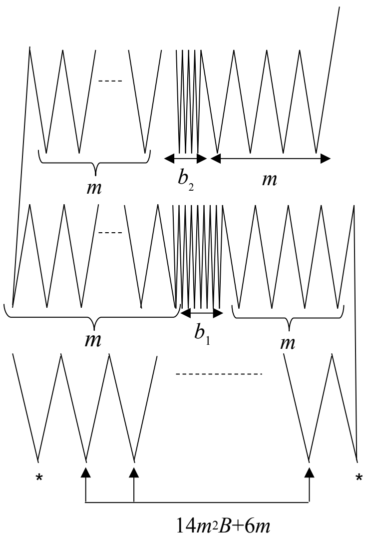


Fig. 5 Overview of folding.

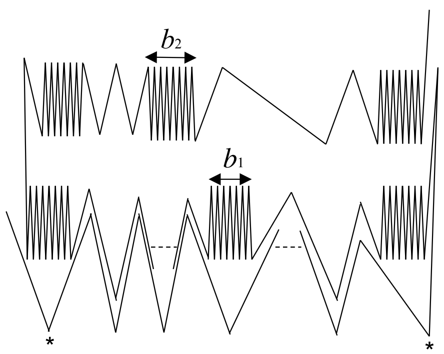


Fig. 6 One feasible way of folding.

between two trash folders, and each gadget corresponding to a_j consists of a "glued" part of width $2b_j$ between two springs of width $2m$.† Therefore, we consider putting gadget parts into unit folders to fill up each folder by exactly $14m^2B+6m$ layers of paper.

We first assume that the instance of 3-Partition is a yes instance and show that P can be folded into unit length. Because the instance is a yes instance, the positive integers $a_1, a_2, a_3, \ldots, a_{3m}$ can be partitioned into m subsets $A_1, A_2, \ldots, A_m$ such that $\sum_{j \in A_k} a_j = B$ for $1 \le k \le m$. Then, we fill the unit folders as follows (Fig. 6). We assume that a_1 is put into a subset $A_{k'}$ for some k'. Then, we put the b_1 gadget into the k'th unit folder, and two paper layers for each unit folder, as shown in Fig. 6. The other remaining segments in the two springs are put into trash folders on both sides. We can observe that these springs also act as unit folders after putting the b_1 gadget into $A_{k'}$. Therefore, we can repeat the same process for each $a_2, a_3, \ldots, a_{3m}$. Then, by the assumption with $b_j = 14m^2a_j$, each unit folder $A_{k'}$ has $14m^2B + 6m$ paper layers at its corresponding crease. Thus, we obtain the required folded state of P.

Next, we assume that the paper strip P is folded, and we construct a solution for 3-Partition from it. We first observe that the total number of paper layers in the spring parts is $3m \cdot 4m = 12m^2$, which is much less than $14m^2$. Therefore, because each $b_j = 14m^2a_j$ and $B/4 < a_j < B/2$, if a unit folder contains $14m^2B + 6m$ paper layers, it is easy to see that each unit folder contains exactly three b_j parts for some b_j, $b_{j'}$ and $b_{j''}$. Then, these parts together make $14m^2B$ paper layers because $6m$ is excessively small compared to each of b_j, $b_{j'}$, and $b_{j''}$. Therefore, we have $a_j + a_{j'} + a_{j''} = B$ for this unit. We can use the same argument for each unit folder, and we can construct a solution for 3-Partition, which completes the proof. □

Indeed, if the proof of Theorem 3 is considered carefully, it can be inferred that the MV assignment in the proof is not necessary. This fact leads us to the following corollary.

Corollary 4: The crease-retrieve problem is strongly NP-complete when $\mathrm{Cw}(i) \in \{0, 1, \ldots, n-1, *_C\}$ and $\mathrm{As}(i) = *_A$ for every i in $\{1, 2, \ldots, n\}$.

Proof. The reduction is identical to one given in the proof of Theorem 3, but we provide no MV assignment to P. When the instance of 3-Partition is a yes instance, we can use the same method as that used in the proof, and thus P can be folded into unit length in a way that satisfies the two functions. Therefore, we assume that the paper strip P is folded, and we construct a solution for 3-Partition from it.

We first focus on the folder part. We have $\mathrm{Cw}(i) = 0$ for each even i, and $\mathrm{Cw}(i)$ has the same value for each $i = 3, 5, 7, \ldots, 2m + 1$. If we valley-fold at some even i, two consecutive unit folders have to have the same crease width, which is impossible. On the other hand, if we mountain-fold at some odd i, we cannot have $\mathrm{Cw}(i-1) = \mathrm{Cw}(i+1) = 0$.

†The *width* here refers to the number of layers.

Therefore, the folder part should make a pleat folding.

Next, we focus on the gadget part for a_j. In this part, we have consecutive $b_j + 1$ creases i with $\mathrm{Cw}(i) = 0$. For the same reason as for the folder part, we can observe that this part should make a pleat folding to satisfy the condition. Then, to satisfy the crease-width conditions in all unit folders, this part has to be put into some unit folder to contribute to its crease width by b_j.

Therefore, we can use the same argument as that applied in the proof of Theorem 3, and obtain the claim. □

4. Polynomial Time Algorithms for Crease-Retrieve Problem

In this section, we investigate the case that the maximum crease width is given as a part of input to the crease-retrieve problem. That is, we are given two functions As and Cw with the maximum crease width k, and the problem asks if there exists a feasible folded state of P consistent with these two functions with the maximum crease width at most k.

As mentioned in Introduction, it is known that a paper strip has a unique folded state for a given MV assignment if and only if it is a pleat folding with MV assignment $(MV)^i$, $(MV)^i M$, $(VM)^i$, or $(VM)^i V$ for some i [1]. This fact leads us to the following observation:

Observation 5: The maximum crease width is 0 if and only if it is a pleat folding. Therefore, the crease-retrieve problem can be solved in linear time when the maximum crease width $k = 0$.

Proof. We first show that the maximum crease width is 0 if and only if it is a pleat folding. It is easy to see that a pleat folding has the maximum crease width 0. Thus we assume that a folded state has the maximum crease width 0, that is, all CW assignments are zero. In this case, once we fold at a crease i by, say, mountain fold, we can glue the ith segment and the $(i+1)$st segment since the crease width at the crease i is zero. It is then easy to see that we have to fold at the $(i-1)$st and the $(i+1)$st creases by valley fold to achieve the maximum crease width 0. Repeating this, we eventually obtain a pleat folding.

When the maximum crease width is constrained to zero, the crease-retrieve problem can be solved in linear time by checking the consistency of the function As. □

We show three simple observations which are useful in this section:

Observation 6: In a folded state, a crease i has an odd crease width if and only if either the segment 1 or the segment n is between the segments i and $i+1$ in the folded state.

Proof. Trivial. □

Observation 7: Let $S = (s_{i_1}, s_{i_2}, s_{i_3}, \ldots, s_{i_k})$ be a sequence of paper segments and let $S' = (s'_{i_1}, s'_{i_2}, s'_{i_3}, \ldots, s'_{i_k})$ be the sequence of right neighbor segments of S (precisely, $s'_{i_j} = s_{i_j+1}$ if i_j is odd, and $s'_{i_j} = s_{i_j-1}$ if i_j is even). The maximum crease width of the final folded state is at least $2(k-1)$ if the following conditions hold (Fig. 7). (1) In the final folded state, s_{i_j} is adjacent to $s_{i_{j+1}}$ and s'_{i_j} is adjacent to $s'_{i_{j+1}}$ for each $j = 1, \ldots, k$. (2) $S \cap S' = \emptyset$.

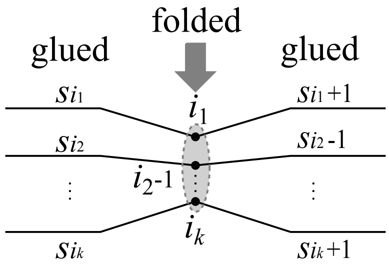


Fig. 7 Paper segments $s_{i_1}, s_{i_2}, s_{i_3}, \ldots, s_{i_k}$ are folded on their right endpoints at once and they are adjacent in the final folded state. We here note that the segment s_{i_j} is flipped if and only if i_j is even regardless of the ordering of the segments.

Proof. Regardless of mountain fold or valley fold, the outermost paper catches the other $k-1$ paper layers within it at the crease, which means that the crease width of the outermost crease is $2(k-1)$. □

We here consider a *shuffle pattern* of length i which is defined by "$(MV)^{i-1}MM(VM)^{i-1}$".[†] Although it seems to be similar to a pleat folding, it is shown that a shuffle pattern of length i has $\binom{2i}{i}$ distinct ways of folding into a unit length since two pleats can be combined in any way like riffle shuffling the cards [6]. On the other hand, a shuffle pattern of length i has only 2 ways of folding if its crease width is restricted to 2.

Observation 8: The shuffle pattern $(MV)^{i-1}MM(VM)^{i-1}$ of length i has 2 folded states of the maximum crease width 2.

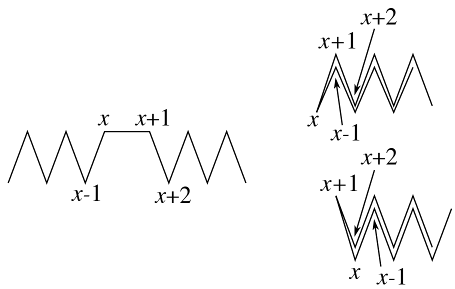


Fig. 8 Two ways of folding of a shuffle pattern $MVMVMMVMVM$.

Proof. For a shuffle pattern of length i, we let $x = 2i-1$. Then the xth and the $(x+1)$st creases are mountains and the $(x-1)$st and the $(x+2)$nd creases are valleys (Fig. 8). To achieve the maximum crease width at most 2, we have 2 possible ways of folding around the xth crease; glue the $(x-1)$st crease to the $(x+1)$st crease or glue the $(x+2)$nd crease to the xth crease. In each way, we have to glue the other line segments as shown in (Fig. 8) to achieve the maximum crease width at

[†] The notion of the shuffle pattern is introduced in [6].

most 2. Thus we have the claim. □

Now we turn to the main theorem in this section. We first suppose that MV assignments are given for all creases.

Theorem 9: The crease-retrieve problem can be solved in linear time when the maximum crease width $k \leq 3$.

In order to show Theorem 9, we show three algorithms that deal with three cases $k = 1$, $k = 2$, and $k = 3$, respectively.

Lemma 10: The crease-retrieve problem can be solved in linear time when the maximum crease width $k = 1$.

Proof. By Observation 7, we cannot stack two layers in the final folded state when $k = 1$. Therefore, clearly, $k = 1$ can be achieved only by the following ordering function f when $n > 3$: (1) $f(1) = 2$, $f(2) = 1$, $f(i) = i$ for each $2 < i < n$, $f(n) = n$, and $f(n + 1) = n + 1$, (2) $f(1) = 1$, $f(2) = 2$, $f(i) = i$ for each $2 < i < n$, $f(n) = n + 1$, and $f(n + 1) = n$, or (3) $f(1) = 2$, $f(2) = 1$, $f(i) = i$ for each $2 < i < n$, $f(n) = n + 1$, and $f(n + 1) = n$. In the context of MV assignments, the case (1) can be represented by $MMVMVMVMVM\cdots$ or $VVMVMVMV\cdots$. In the case (2), the last two MV assignments are the same, and both hold in the case (3).

When $n = 3$, $k = 1$ can be achieved when (1) $f(1) = 1$, $f(2) = 3$, and $f(3) = 2$, or (2) $f(1) = 2$, $f(2) = 1$, and $f(3) = 3$. In the context of MV assignments, they can be represented by MM or VV. The maximum crease width k cannot be 1 when $n = 1, 2$.

Therefore, the crease-retrieve problem can be solved in linear time. Precisely speaking, for given two functions $\mathrm{As} : [1, n] \to \{M, V\}$ and $\mathrm{Cw} : [1, n] \to \{0, 1, \ldots, n-1, *_C\}$, it is not difficult to check if they are consistent to the patterns above in linear time. □

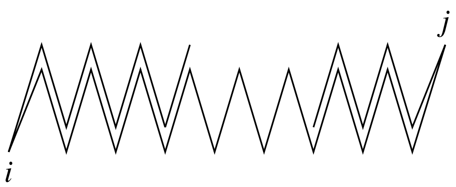


Fig. 9 Representative way of folding for $k = 2$.

Lemma 11: The crease-retrieve problem can be solved in linear time when the maximum crease width $k = 2$.

Proof. By Observation 7, once we fold a long paper strip, the crease width at the crease points on the folded part already achieves $k = 2$. Then we have to make a pleat folding in the folded part by Observation 5; otherwise, the maximum crease width becomes greater than $k = 2$. Therefore, intuitively speaking, $k = 2$ can be achieved by the following way of folding (Fig. 9): (1) we first fold along the crease i and glue the paper strip, (2) we then fold along the crease j with $\frac{n+1}{2} \leq j - i$ and glue the paper strip, (3) we then fold the glued paper strip in pleat folding. We can observe that after the steps (1) and (2), the thickness of the paper strip at each crease is at most 2. Precisely, at the creases on both sides, it has thickness 2, and it may have some creases of thickness 1 in the central part of the paper strip. When $\frac{n+1}{2} > j - i$, the thickness of the paper strip at each crease is 2 or 3 since both endpoints of the paper strip are piled up on some creases between i and j.

We here note that around the creases i and j, we have the same pattern of the shuffle pattern. By Observation 8, we have two ways of folding at the creases i and j. To avoid the overlapping of the endpoints of the paper strip, we may choose the further points.

By combining the arguments in the proofs of Observations 5 and 7, we can observe that this is the only way to achieve $k = 2$.

In the context of MV assignments, the assignment of the maximum crease width $k = 2$ consists of three pleat folding. For example, we assume that $s_1 = MVM$, $s_2 = (MV)^4M$, and $s_3 = (MV)^2M$. Then $s_1s_2s_3 = MVMMVMVMVMVMMVMVM$ has a feasible folded state of the maximum crease width $k = 2$. We call these two line segments between two consecutive same (MM in this case) MV assignments *turning points*. When we solve the crease-retrieve problem, we first find possible turning points from As, and then check the consistency with Cw. Among possible turning points, we choose the farthest pairs of creases and check if $\frac{n+1}{2} \leq j - i$ for the corresponding creases. The correctness of the algorithm is easy to follow, and they can be done in linear time. □

Lemma 12: The crease-retrieve problem can be solved in linear time when the maximum crease width $k = 3$.

Proof. By Observation 6, at least one of the segments 1 and n should be used to achieve $k = 3$. By the proof of Lemma 10, we can observe that the segment 1 or n can increase the crease width of its neighbor in the folded state. That is, the folded state of the maximum crease width $k = 3$ is the same as the folded state of the maximum crease width $k = 2$ except the segments 1 or n, and these segments can be put into their neighbor creases.

We first consider the case that both of the segments 1 and n are used to achieve $k = 3$. Then, an assignment of the maximum crease width $k = 3$ consists of three pleat foldings s_1, s_2, s_3 with two MV assignments t_1 and t_2 with $t_1, t_2 \in \{M, V\}$ such that (1) the assignment is given by $t_1s_1s_2s_3t_2$, (2) t_1 is the same as the first MV assignment of s_1, (3) the last MV assignment of s_1 is the same as the first MV assignment of s_2, (4) the last MV assignment of s_2 is the same as the first MV assignment of s_3, and (5) t_2 is the same as the last MV assignment of s_3. The length constraint is the same as Lemma 11. That is, $|s_1| + |s_3| \leq |s_2|$, where $|s|$ denotes the total number of M and V in s. The other cases are similar and omitted.

By the proofs of Lemmas 10 and 11, it is not difficult to confirm the correctness of the argument above, and the crease-retrieve problem can be solved in linear time when $k = 3$. □

Now we show the proof of Theorem 9:

Proof. By Lemmas 10, 11, and 12, we can solve the crease-retrieve problem when the maximum crease width is either $k = 1$, $k = 2$, or $k = 3$ in linear time. Thus we can solve the problem for each of the cases $k = 1$, $k = 2$, and $k = 3$ in this order in linear time. □

Intuitively, if every crease width is given and the maximum crease width is at most 3, we start at some crease point to glue some consecutive line segments sharing the crease, expand the gluing, and make pleat folding. Therefore, once we choose turning points in the case $k = 2$, the way of folding is almost uniquely determined.

In the arguments above, we suppse that $\mathrm{As} : [1, n] \to \{M, V\}$ is given. That is, all MV assignments are given. With careful analysis, we can observe that Lemmas 10, 11, and 12 hold even if $\mathrm{As}(i) = *_A$. That is, since the final folded state is essentially unique when $k \leq 3$, the assignment for each i with $\mathrm{As}(i) = *_A$ is determined by the unique final folded state. (Precisely, when $\mathrm{As}(i) = *_A$ for all i, we have two feasible assignments determined by $\mathrm{As}(1) = M$ or $\mathrm{As}(1) = V$.) Thus we have Theorem 9 for any $\mathrm{As} : [1, n] \to \{M, V, *_A\}$.

Interestingly, this strategy does not work in the case $k = 4$. When $\mathrm{Cw} = [2, 0, 0, 2, 0, 4, 0, 2, 0, 0, 2, 0]$, we have two completely different feasible folded states as shown in Fig. 10.† That is, we cannot extend Theorem 9 to the case of $k = 4$ in a straightforward way.

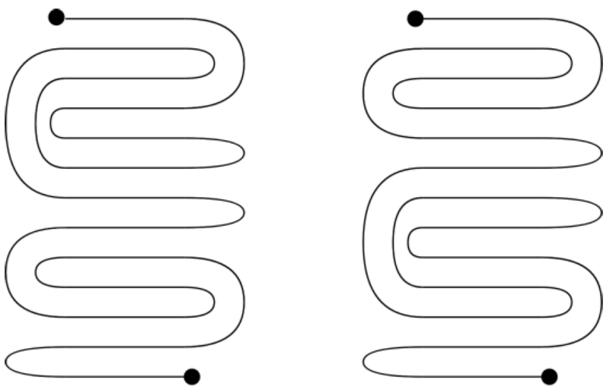


Fig. 10 Two feasible folded states for $\mathrm{Cw} = [2, 0, 0, 2, 0, 4, 0, 2, 0, 0, 2, 0]$.

5. Concluding Remarks

In this study, we introduce the crease-retrieve problem and investigate its computational complexity. As investigated in [1], an MV assignment is not sufficient for determining the folded state of a strip of a paper. On the other hand, an MV assignment and a CW assignment are sufficient for determining the folded state. When we provide partial information on the CW assignment, the decision problem is NP-complete, whether we provide a full MV assignment or provide no MV assignment.

An interesting case is the case when only a full CW assignment is given. As shown in the example in Fig. 10, feasible folded states are not unique in general, and it is completely different to the case that the maximum crease width is up to 3. A natural open problem is whether we can determine in polynomial time if a feasible folded state exists when only a full CW assignment is given.

In the context of the stamp-folding problem, the characterization of the number of folded states for a given MV assignment remains open. As mentioned in Introduction, it is known that an MV assignment is a pleat folding if and only if it has only one folded state. On the other hand, the shuffle pattern $(MV)^i MM(VM)^i$ has exponentially many folded states. It is also known that a random MV assignment has exponentially many folded states. The characterization of MV assignments in which each assignment has polynomial number of feasible folded states is still open.

†This example was found by Giovanni Viglietta in October, 2022.

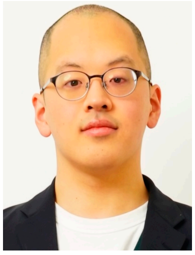
Acknowledgement

The authors are grateful to Giovanni Viglietta for pointing out the example in Fig. 10.

References

[1] R. Uehara, Origami 5, ch. Stamp foldings with a given mountain-valley assignment, pp.585–597, CRC Press, 2011.
[2] T. Umesato, T. Saitoh, R. Uehara, H. Ito, and Y. Okamoto, "The complexity of the stamp folding problem," Theoretical Computer Science, vol.497, pp.13–19, 2013.
[3] E.D. Demaine and J. O'Rourke, Geometric Folding Algorithms: Linkages, Origami, Polyhedra, Cambridge University Press, 2007.
[4] R.I. Nishat, Map Folding, master thesis, University of Victoria, Department of Computer Science, 2013.
[5] M.R. Garey and D.S. Johnson, Computers and Intractability — A Guide to the Theory of NP-Completeness, Freeman, 1979.
[6] T. Asano, E.D. Demaine, M.L. Demaine, and R. Uehara, "NP-completeness of generalized Kaboozle," Journal of Information Processing, vol.20, no.3, pp.713–718, 2012.

Junnosuke Hoshido is a researcher in computational origami. He received master's degree from Japan Advanced Institute of Science and Technology, Japan (JAIST), in 2024. His research interests include computational origami.



Tonan Kamata is an assistant professor at the School of Information Science, Japan Advanced Institute of Science and Technology (JAIST). He was a postdoctoral Fellow (PD) of a Japan Society for the Promotion of Science (JSPS) at JAIST. He received a B. S. from Tokyo University of Science, Japan, in 2019, and M. S. and Ph. D. degrees in Information Science from JAIST in 2021 and 2023, respectively.

Tsutomu Ansai is a professor in National Institute of Technology, Ibaraki College. He received B. Eng., M. Eng., and Ph. D. degrees from Yamagata University, Japan, in 1994, 1996, and 2002, respectively. His research interests include information security and computational complexity. He is a member of IEICE, IPSJ, and JSIAM.

Ryuhei Uehara is a vice president, an executive dean, a director of JAIST Gallery, and 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 in 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. Especially, he is engrossed in computational origami, games and puzzles from the viewpoints of theoretical computer science. He is a member of EATCS, IEICE, IPSJ, and the chair of EATCS Japan Chapter.