Folding a Paper Strip to Minimize Thickness☆

Erik D. Demaine[a], David Eppstein[b], Adam Hesterberg[c], Hiro Ito[d], Anna Lubiw[e], Ryuhei Uehara[f], Yushi Uno[g]

[a] *Computer Science and Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, USA*
[b] *Computer Science Department, University of California, Irvine, USA*
[c] *Department of Mathematics, Massachusetts Institute of Technology, USA*
[d] *School of Informatics and Engineering, University of Electro-Communications, Tokyo, Japan*
[e] *David R. Cheriton School of Computer Science, University of Waterloo, Ontario, Canada*
[f] *School of Information Science, Japan Advanced Institute of Science and Technology, Ishikawa, Japan*
[g] *Graduate School of Science, Osaka Prefecture University, Osaka, Japan*

Abstract

In this paper, we study how to fold a specified origami crease pattern in order to minimize the impact of paper thickness. Specifically, origami designs are often expressed by a mountain-valley pattern (plane graph of creases with relative fold orientations), but in general this specification is consistent with exponentially many possible folded states. We analyze the complexity of finding the best consistent folded state according to two metrics: minimizing the total number of layers in the folded state (so that a "flat folding" is indeed close to flat), and minimizing the total amount of paper required to execute the folding (where "thicker" creases consume more paper). We prove both problems strongly NP-complete even for 1D folding. On the other hand, we prove both problems fixed-parameter tractable in 1D with respect to the number of layers.

Keywords: linkage, NP-complete, optimization problem, rigid origami.

1. Introduction

Most results in computational origami design assume an idealized, zero-thickness piece of paper. This approach has been highly successful, revolutionizing artistic origami over the past few decades. Surprisingly complex origami designs are possible to fold with real paper thanks in part to thin and strong

☆A preliminary version of this draft was presented at the 9th International Workshop on Algorithms and Computation (WALCOM 2015).

Email addresses: edemaine@mit.edu (Erik D. Demaine), eppstein@uci.edu (David Eppstein), achester@mit.edu (Adam Hesterberg), itohiro@uec.ac.jp (Hiro Ito), alubiw@uwaterloo.ca (Anna Lubiw), uehara@jaist.ac.jp (Ryuhei Uehara), uno@mi.s.osakafu-u.ac.jp (Yushi Uno)

paper (such as made by Origamido Studio) and perhaps also to some unstated and unproved properties of existing design algorithms.

This paper is one of the few attempting to model and optimize the effect of positive paper thickness. Specifically, we consider an origami design specified by a *mountain-valley pattern* (a crease pattern plus a mountain-or-valley assignment for each crease), which in practice is a common specification for complex origami designs. Such patterns only partly specify a folded state, which also consists of an *overlap order* among regions of paper. In general, there can be exponentially many overlap orders consistent with a given mountain-valley pattern [1]. Furthermore, it is NP-hard to decide flat foldability of a mountain-valley pattern, or to find a valid flat folded state (overlap order) given the promise of flat foldability [2]. But for 1D pieces of paper, the same problems are polynomially solvable [3, 4], opening the door to optimizing the effects of paper thickness among the exponentially many possible flat folded states — the topic of this paper.

One of the first mathematical studies about paper thickness is also primarily about 1D paper. Britney Gallivan [5], as a high school student, modeled and analyzed the effect of repeatedly folding a positive-thickness piece of paper in half. Specifically, she observed that creases consume a length of paper proportional to the number of layers they must "wrap around", and thereby computed the total length of paper (relative to the paper thickness) required to fold in half n times. She then set the world record by folding a 4000-foot-long piece of (toilet) paper in half twelve times, experimentally confirming her model and analysis.

Motivated by Gallivan's model, Uehara [6] defined the *stretch* at a crease to be the number of layers of paper in the folded state that lie between the two paper segments hinged at the crease. We will follow the terminology of Umesato et al. [7] who later replaced the term "stretch" with *crease width*, which we adopt here. Both papers considered the case of a strip of paper with *equally spaced* creases but an arbitrary mountain-valley assignment. When the mountain-valley assignment is uniformly random, its expected number of consistent folded states is $\Theta(1.65^n)$ [1]. Uehara [6] asked whether it is NP-hard, for a given mountain-valley assignment, to minimize the maximum crease width or to minimize the total crease width (summed over all creases). Umesato et al. [7] showed that the first problem is indeed NP-hard, while the second problem is fixed-parameter tractable.

We consider the problem of minimizing crease width in the more general situation where the creases are not equally spaced along the strip of paper. This more general case has some significant differences with the equally spaced case. For one thing, if the creases are equally spaced, all mountain-valley patterns can be folded flat by repeatedly folding from the rightmost end; in contrast, in the general case, some mountain-valley patterns (and even some crease patterns) have no consistent flat folded state that avoids self-intersection. Flat foldability of a mountain-valley pattern can be checked in linear time [3] [4, Sec. 12.1], but it requires a nontrivial algorithm.

For creases that are not equally spaced, the notion of crease width must also

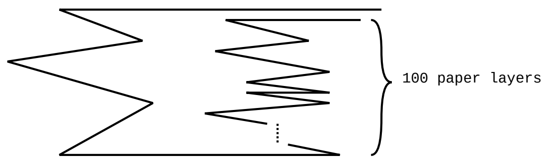

Figure 1: How can we count the paper layers?

be defined more precisely, because it is not so clear how to count the layers of paper between two segments at a crease. For example, in Figure 1, although no layers of paper come all the way to touch the three creases on the left, we want the sum of their crease widths to be 100.

We consider a folded state to be an assignment of the segments to horizontal *levels* at integer y coordinates, with the creases becoming vertical segments of variable lengths. See Figure 2 and the formal definition below. Then the *crease width* at a crease is simply the number of levels in between the levels of the two segments of paper joined by the crease. That is, it is one less than the length of the vertical segment assigned to the crease. In the case of equally spaced creases, this is the number of layers of paper between the two horizontal segments at the crease, so we have generalized the previous definition. Analogous to Uehara's open problems [6], we will study the problems of minimizing the maximum crease width and minimizing the total crease width for a given mountain-valley pattern. The total crease width corresponds to the extra length of paper needed to fold the paper strip using paper of positive thickness, naturally generalizing Gallivan's work [5]. [1]

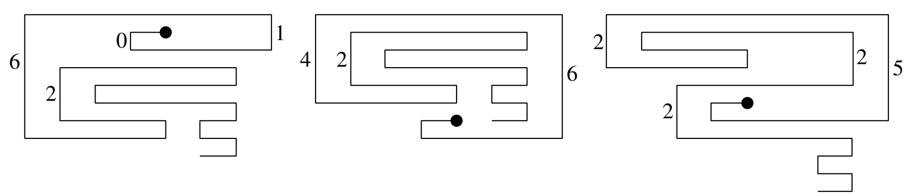

Figure 2: Three different folded states of the crease pattern $VMVMVVMMMM$ (ending at the dot). The positive crease width of each crease is given beside its corresponding vertical segment. Each folding is better than the other two in one of the three measures, where h is the height, m is the maximum crease width, and t is the total crease width: (1) $h = 11, \underline{m = 5}, t = 11$, (2) $\underline{h = 8}, m = 6, t = 12$, and (3) $h = 9, m = 6, \underline{t = 9}$.

In the setting where creases need not be equally spaced, there is another sensible measure of thickness: the *height* of the folded state is the total number of levels. The height is always $n + 1$ for n equally spaced creases, but in our

[1] In this figure, we assume orthogonal bends to make the notions clear. On the other hand, Gallivan measures turns as circular arcs, this changes the length by only a constant factor. Gallivan's model seems to correspond better to practice.

setting different folds of the same crease pattern can have different heights. Figure 2 shows how the three measures can differ. Of course, the maximum crease width is always less than the height.

Our main results (section 3) are NP-hardness of the problem of minimizing height and the problem of minimizing the total crease width. See Table 1. In addition, we show in section 4 that the problem of minimizing height is fixed-parameter tractable, by giving a dynamic programming algorithm that runs in $O(2^{O(h \log h)}n + n \log n)$ time, where h is the minimum height. This dynamic program can be adapted to minimize maximum crease width or total crease width for foldings of bounded height, with the same time complexity as measured in terms of the height bound.

thickness measure	**equally spaced creases**		**general creases**
height	trivial		NP-hard (new)
max crease width	NP-hard [7]	$\Longrightarrow$	NP-hard [7]
total crease width	open		NP-hard (new)

Table 1: Complexity of minimizing thickness, by model, for the case of equally spaced creases and for the general case.

2. Preliminaries

We model a *paper strip* as a one-dimensional line segment. It is rigid except at *creases* $p_1, p_2, \ldots, p_n$ on it; that is, we are allowed to fold only at these crease points. For notational convenience, the two ends of the paper strip are denoted by p_0 and p_{n+1}. We are additionally given a *mountain-valley string* $s = s_1 s_2 \cdots s_n$ in $\{M, V\}^n$. In the *initial state* the paper strip is placed on the x-axis, with each crease p_i at a given coordinate x_i. Without loss of generality, we assume that $x_0 = 0 < x_1 < \cdots < x_n < x_{n+1}$. Sometimes we will normalize so $x_{n+1} = 1$. We may consider the paper strip as a sequence of $n+1$ *segments* S_i of length $x_{i+1} - x_i$ delimited by the creases p_i and p_{i+1} for each $i \in \{0, 1, \ldots, n\}$. Each crease's letter determines how we can fold it: when it is M (resp. V), the two paper segments sharing the crease are folded in the direction such that their bottom sides (resp. top sides) are close to touching (although they may not necessarily touch if they have other paper layers between them).

Following Demaine and O'Rourke [4] we define a *flat folding* (or *folded state*) via the relative stacking order of collocated layers of paper. We begin with x_0 at the origin, and the first segment lying in the positive x-axis. The lengths of the segments determine where each segment lies along the x-axis (because they zig-zag). Suppose that point p_i is mapped to x-coordinate $f(p_i)$. The mountain-valley assignment determines for each segment S_i whether S_i lies above or below S_{i+1}. We extend this to specify the relative vertical order of any two segments that overlap horizontally. This defines a *folded state* so long as the vertical ordering of segments is transitive and *non-crossing*. More formally:

1. if segments S_i and S_{i+1} are joined by a crease at x-coordinate $f(p_i)$ then for any segment S that extends to the left and the right of $f(p_i)$, either $S < S_i, S_{i+1}$ or $S > S_i, S_{i+1}$,
2. if segments S_i and S_{i+1} are joined by a crease at x-coordinate $f(p_i)$, segments S_j and S_{j+1} are joined by a crease at the same x-coordinate $f(p_j) = f(p_i)$, and all four segments extend to the same side of the crease, then the two creases do not *interleave*, i.e., we do not have $A < B < A' < B'$ where A and A' are one of the pairs joined at a crease and B and B' are the other pair.

When the x_i's are not equally spaced, the paper strip cannot necessarily be folded flat with the given mountain-valley assignment. For example, segments of lengths 2, 1, 2 do not allow the assignment VV. There is a linear time algorithm to test whether an assignment has a flat folding [4].

In order to define crease width, we will use an enhanced notion of folded states: a *leveled folded state* is an assignment of the segments to *levels* from the set $\{1, 2, \ldots\}$ such that the resulting vertical ordering of segments is a valid folded state. See Figure 2.

Clearly a leveled folded state provides a folded state, but in the reverse direction, a folded state may correspond to many leveled folded states. However, for the measures we are concerned with, we can efficiently compute the best leveled folded state corresponding to any folded state.

The *height* of a leveled folded state is the number of levels used. Given a folded state, the minimum height of any corresponding leveled folded state can be computed efficiently, since it is the length of a longest chain in the partial order defined on the segments in the folded state.

The *crease width* of a crease in a leveled folded state is the number of levels in-between the two segments joined at the crease. We are interested in minimizing the maximum crease width and in minimizing the total crease width, i.e., the sum of the crease widths of all the creases. In both cases, given a folded state, we can compute the best corresponding leveled folded state using linear programming.

A mountain-valley string that alternates $MVMVMV\ldots$ is called a *pleat*. For equally-spaced creases, the legal folded state is unique (up to reversal of the paper) if and only if s is a pleat [6, 1].

In this paper, we consider three versions of minimizing thickness in a flat folding. For all three problems we have:

INSTANCE: A paper strip P, with creases $p_1, \ldots, p_n$ at positions $x_1, \ldots, x_n$ with a mountain-valley string $s \in \{M, V\}^n$, and a natural number k.

The three problems are as follows:

MinHeight. QUESTION: Is there a leveled folded state of height at most k?

MinMaxCW. QUESTION: Is there a leveled folded state with maximum crease width at most k?

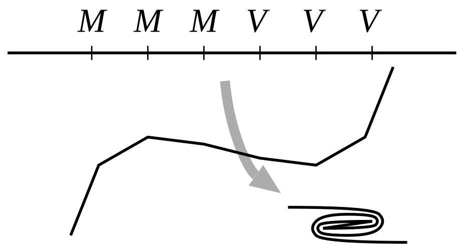

Figure 3: The unique flat folding of the string MMMVVV.

MinSumCW. QUESTION: Is there a leveled folded state with total crease width at most k?

3. NP-completeness

In this section, we show NP-completeness of the MinHeight problem and the MinSumCW problem. We remind the reader that the pleat folding has a unique folded state [6, 1]. We borrow some useful ideas from [7].

Observation 1. *Let n be a positive integer, P be a strip with creases $p_1, \ldots, p_{2n}$, and s be a mountain-valley string $M^n V^n$. We suppose that the paper segments are of equal length except a longer one at each end. Precisely, we have $|S_i| = |S_j| < |S_0| = |S_{2n}|$ for all i, j with $0 < i, j < 2n$, where $|S_i|$ denotes the length of the segment S_i, Then the legal folded state with respect to s is unique up to reversal of the paper. Precisely, the legal folded state has the segments in vertical order $S_0, S_{2n-1}, S_2, S_{2n-3}, \ldots, S_{2i}, S_{2(n-i)-1}, \ldots, S_1, S_{2n}$ or the reverse.*

A simple example is given in Figure 3. We call this unique folded state the *spiral folding* of size $2n$.

Our hardness proofs reduce from 3-PARTITION, defined as follows.

Problem: 3-PARTITION (cf. [8])

Instance: A finite multiset $A = \{a_1, a_2, \ldots, a_{3m}\}$ of $3m$ positive integers. Define $B = \sum_{j=1}^{3m} a_j/m$. We may assume each a_j satisfies $B/4 < a_j < B/2$, and $\sum_{j=1}^{3m} a_j = mB$.

Question: Can A be partitioned into m disjoint sets $A^{(1)}, A^{(2)}, \ldots, A^{(m)}$ such that $\sum A^{(i)} = B$ for every i with $1 \le i \le m$?

It is well-known that 3-PARTITION is strongly NP-complete, that is, it is NP-hard even if the input is written in unary notation [8]. Our reductions are based on a similar reduction of Umesato et al. [7].

Theorem 2. *The MinHeight problem for paper folding height is NP-complete.*

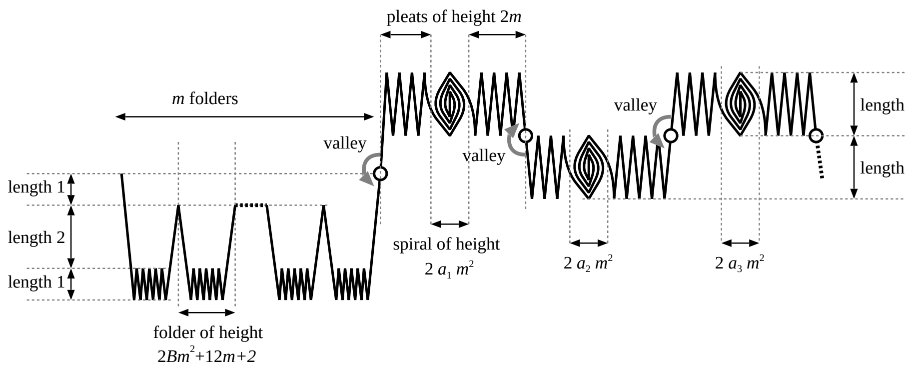

Figure 4: Outline of the reduction for Theorem 2. Note that this figure and the following ones are sideways compared to previous figures, so height is horizontal.

Proof. It is easy to see that the problem is in NP. To prove hardness, we reduce from 3-PARTITION.

Given an instance $\{a_1, a_2, \ldots, a_{3m}\}$ of 3-PARTITION, we construct a corresponding paper strip P as follows (Figure 4). The left part of P is folded into m *folders*, where each folder is a pleat consisting of $2Bm^2 + 12m$ *short* segments of length 1 between two segments of length 3, except for the very first and last long segments, which have length 4.[2] The right part of P contains $3m$ gadgets, where the ith gadget represents the integer a_i. The ith gadget consists of one spiral of height $2a_i m^2$ between two $2m$ pleats. Each line segment in the gadget has length 2 except for the one end segment which has length 3. This construction can be carried out in polynomial time.

By Observation 1, each spiral folds uniquely, and also we know that each pleat folds uniquely [6, 1]. Therefore, the folders and gadgets fold uniquely. Figure 4 shows the unique combination of these foldings before folding at the *joints*, depicted by white circles. Once the joints are valley folded, the folding will no longer be unique.

The intuition is that the pleats between spirals give us the freedom to place the spiral of each gadget in any folder. The heights of the spirals ensure that the packing of spirals into folders acts like 3-PARTITION. More precisely, we show:

Claim 3. *An instance (A, B) of 3-PARTITION has a solution if and only if the paper strip P can be folded with height at most $2Bm^3 + 12m^2 + 2m$.*

To prove the claim, first suppose that the 3-PARTITION instance $\{a_1, a_2, \ldots, a_{3m}\}$ has a solution, say, $A^{(1)}, A^{(2)}, \ldots, A^{(m)}$. Then we have $A^{(i)} \subset A$, $\left|A^{(i)}\right| = 3$, $\sum A^{(i)} = B$ for each i in $\{1, 2, \ldots, m\}$, and $A = \dot{\bigcup}_{i=1}^{m} A^{(i)}$. For the

[2] In the reduction in [7], this folder consists of just two segments.

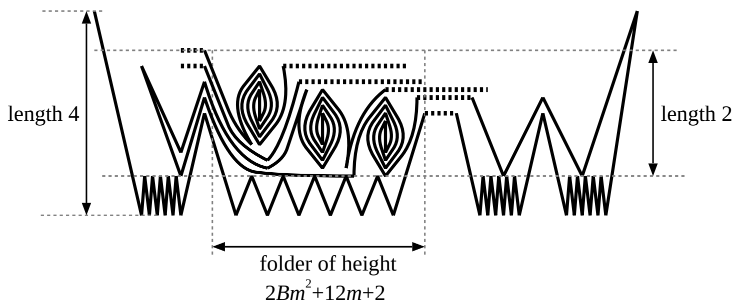

Figure 5: Putting spirals into a folder.

three items in $A^{(i)}$, we put the three corresponding spirals into the ith folder; see Figure 5. Because the items sum to B, the total height of the spirals is $2Bm^2$. Each gadget uses $2(m-1)$ of the $4m$ total pleats to position its spiral, leaving $2(m+1)$ pleats which we put in the folder of the spiral, for a total of $6(m+1)$. The $3m-3$ other gadgets also place two pleats in this spiral, just passing through, for a total of $6m-6$. Thus each folder has at most $2Bm^2+12m$ layers added and, because it already had $2Bm^2+12m$ short pleat segments, its final height is $2Bm^2+12m+2$ (including the two long segments). Therefore the total height of the folded state is $2Bm^3+12m^2+2m$ as desired.

Next suppose that the paper strip P can be folded with height at most $k = 2Bm^3+12m^2+2m$. There are m folders each with height at least $2Bm^2+12m+2$. Therefore, each folder must have height exactly $2Bm^2+12m+2$ and the number of levels inside the folder is $2Bm^2+12m$. Furthermore, the spirals must be folded into the folders. We claim that the spirals in each folder must have total height at most $2Bm^2$. If the spirals in one of the folders have total height more than $2Bm^2$, they have height at least $2(B+1)m^2 = 2Bm^2+2m^2$, which is greater than $2Bm^2+12m$ if $2m^2 > 12m$, i.e., if $m > 6$ (which we may assume without loss of generality). In particular, each folder must have at most three spirals: because each $a_j > B/4$, each spiral has height greater than $Bm^2/2$, so four spirals would have height greater than $2Bm^2$. Because the $3m$ spirals are partitioned among m folders, exactly three spirals are placed in each folder, and their total height of at most $2Bm^2$ corresponds to three elements of sum at most (and thus exactly) B. Therefore we can construct a solution to the 3-PARTITION instance. □

Theorem 4. *The* MinSumCW *problem is NP-complete.*

Proof. This reduction from 3-PARTITION is a modification to the reduction to MinMaxCW in the proof of Theorem 2; refer to Figures 6 and 7. We introduce a deep "molar" at both ends of each gadget, which must fit into deep "gums" at either end of the folders. Specifically, for $z = m^4$, each gum has $2z+4m$ pleats, and each molar in the ith gadget has $2z+4(m-i)$ pleats. In the intended folded state, the left molars nest inside each other (smaller/later inside larger/earlier) within the left gum, and similarly for the right molars into the right gum. In this case, every molar and every gum remains at its minimum possible height given by its pleats.

The heights of the molars guarantee that, in any legal folding, every molar ends up in a gum. If, in any of the m gadgets, the right molar folds into the left gum, the left molar of that gadget also folds into the left gum, so the left gum has height at least $4z$ in the folded state, $2z - 4m$ more than its minimum height. This increase in height translates into an equal increase in the total crease width (because the number of creases remains fixed). Because $z = m^4$, this increase will dominate the total crease width. Therefore every folding with a right molar in the left gum, or with a left molar in the right gum, has total crease width larger than the intended folded state.

This argument guarantees that, in any solution folding to the MinSumCW instance, each gadget has its left molar in the left gum and its right molar in the right gum. In this case, the height of each gadget is the height of its spiral plus the height of all the folders, which will be minimized precisely when the folders do not grow in height. The total crease width of a gadget differs from its height by a fixed amount (the number of creases), so we arrive at the same minimization problem. Thus the proof reduces to the MinMaxCW construction. □

4. Fixed-parameter tractability

In this section, we show that the three problems, MinHeight, MinMaxCW, and MinSumCW are fixed parameter tractable with respect to height.

Theorem 5. *Testing whether a strip with n folds has a folded state with height at most h can be done in time $O(2^{O(h \log h)} n + n \log n)$.*

Proof. We use a dynamic programming algorithm that sweeps from left to right across the line onto which the strip is folded, stopping at each of the points on the line where a strip endpoint or fold point (crease) is placed. First we note that in any folded state, each segment is placed at the same interval independent of the crease pattern. Precisely, in any folded state from the initial state of the paper strip with coordinates $x_0 = 0 < x_1 < x_2 < \cdots < x_n < x_{n+1}$, the segment S_0 is placed at $[x_0, x_1]$, the segment S_1 is placed at $[x_1 - x_2, x_1]$, S_2 is placed at $[x_1 - x_2, x_1 - x_2 + x_3]$, and so on. Therefore, as a preprocessing step, we first compute x_1, $x_1 - x_2$, $x_1 - x_2 + x_3$, $x_1 - x_2 + x_3 - x_4$, $\cdots$, in linear time, and sort them in $O(n \log n)$ time to obtain the stopping points. At each point of the line between two stopping points, there can be at most h segments of the strip, for otherwise the height would necessarily be larger than h and we could terminate the algorithm, returning that the height is not less than or equal to h. We define a *level assignment* for a point p between two stopping points to be a function a from input segments that overlap p to distinct integer levels from 1 to h. The number of possible level assignments for any point is therefore at most h^h.

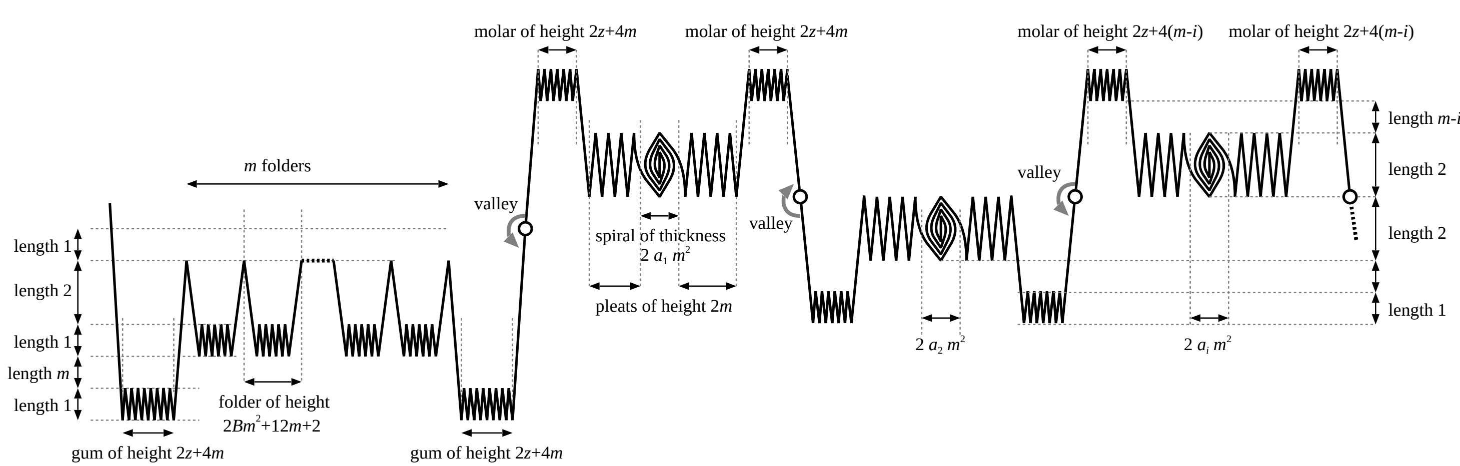

Figure 6: Outline of the reduction.

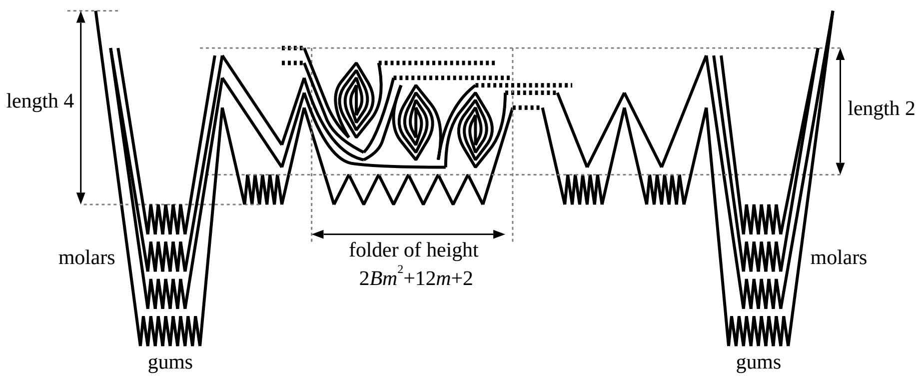

Figure 7: Putting spirals into folders and molars into gums.

Let $\varepsilon > 0$ be smaller than the distance between any two stopping points. At each stopping point p of the algorithm, we will have a set A of allowed level assignments a^- for the point $p - \varepsilon$; initially (for the leftmost point of the folded input strip) A will contain the unique level assignment for the empty set of segments. For each combination of a level assignment a^- in A for the point $p - \varepsilon$ and an arbitrary level assignment a^+ for the point $p + \varepsilon$, we check whether there is a valid folding of the part of the strip between $p - \varepsilon$ and $p + \varepsilon$ that matches this level assignment. To do so, we check the following four conditions that capture the *noncrossing* conditions defined in Section 2:

- If a segment S extends to both sides of p without being folded at p, it has the same level on both sides. That is, $a^-(S) = a^+(S)$.

- For each two input folds at p that connect pairs of segments that overlap $p - \varepsilon$, the levels of these pairs of segments are nested or disjoint. That is, if we have a fold connecting segments S_i and S_{i+1}, and another fold connecting segments S_j and S_{j+1}, then $[a^-(S_i), a^-(S_{i+1})]$ and $[a^-(S_j), a^-(S_{j+1})]$ are either disjoint intervals or one of these two intervals contains the other.

- For each two input folds at p that connect pairs of segments that overlap $p + \varepsilon$, the levels of these pairs of segments are nested or disjoint. This is a symmetric condition to the previous one, using a^+ instead of a^-.

- For each fold at p, connecting segments S_i and S_{i+1}, and for each input segment S_j that crosses p without being folded there, the interval of levels occupied by the fold should not contain the level of S_j. That is, if the two segments S_i and S_{i+1} extend to the left of p, then the interval $[a^-(S_i), a^-(S_{i+1})]$ should not contain $a^-(S_j)$. If the two segments extend to the right of p, then we have the same condition using a^+ instead of a^-.

If the pair (a^-, a^+) passes all these tests, we include a^+ in the set of valid level assignments for $p + \varepsilon$, which we will then use at the next stopping point of the algorithm.

If, at the end of this process, we reach the rightmost stopping point with a nonempty set of valid level assignments (necessarily consisting of the unique level assignment for the empty set of segments) then a folding of height h exists. The folding itself may be recovered by storing, for each level assignment a^+ considered by the algorithm, one of the level assignments a^- such that $a^- \in A$ and (a^-, a^+) passed all the tests above. Then, backtracking through these pointers, from the rightmost stopping point back to the leftmost one, will give a sequence of level assignments such that each consecutive pair is valid, which describes a consistent folding of the entire input strip.

The running time of the algorithm is the number of stopping points multiplied by the number of pairs of level assignments for each stopping point and the time to test each pair of level assignments. This is $O(2^{O(h \log h)} n)$. Adding the $O(n \log n)$ preprocessing time gives the time bound claimed in the theorem. □

The same algorithm can be enhanced to show that MinMaxCW and MinSumCW are fixed parameter tractable with respect to the height of the solution.

Theorem 6. *Given a strip with n folds, we can find in time $O(2^{O(h \log h)} n + n \log n)$ a folded state that has height at most h and minimizes: (1) the maximum crease width; (2) the total crease width.*

Proof. We use the same dynamic programming algorithm as in the proof of Theorem 5. With every level assignment we store its maximum and total crease width. At each stopping point, when we add a^+ to the set of valid level assignments for $p + \varepsilon$, we must compute its maximum and total crease width. To do this, we need the crease width of every fold at p in a^- or a^+. We look at every pair of segments S_i and S_{i+1} that are joined in a fold at p. If these segments extend to the left of p then their crease width is $|a^-[S_i] - a^-[S_{i+1}]|$, and if the two segments extend to the right of p then their crease width is $|a^+[S_i] - a^+[S_{i+1}]|$.

The maximum crease width of a^+ is the max of the maximum crease width of a^- and the crease width of each fold at p in a^- or a^+. The total crease width of a^+ is the sum of the total crease width of a^- and the crease width of each fold at p in a^- or a^+.

Note that we may arrive at a level assignment a^+ multiple times by pairing it with different assignments a^-. Whenever this occurs we should preserve the minimum values for maximum and total crease width of a^+.

Performing these computations only adds a constant factor to the running time of the algorithm. □

5. Conclusions

We have shown that even in 1D it is NP-hard to find a folding for a given crease pattern and mountain-valley assignment that minimizes the total number of layers (the height) or the amount of extra paper needed to execute the folding (the total crease width).

In the positive direction, we have shown that these 1D problems are fixed parameter tractable with respect to the height of the solution. One remaining open question is to find an FPT algorithm for minimizing maximum crease width with respect to maximum crease width as the parameter. The analogous problem for total crease width has been solved only for the special case of equally-spaced creases [7], although this special case is not known to be NP-hard.

A more challenging question is to find fixed parameter tractable algorithms for any of these problems in 2D.

Acknowledgments

This research was performed in part at the 29th Bellairs Winter Workshop on Computational Geometry.

Erik Demaine was supported in part by NSF ODISSEI grant EFRI-1240383 and NSF Expedition grant CCF-1138967. David Eppstein was supported in part by NSF grant 1228639 and ONR grant N00014-08-1-1015. Adam Hesterberg was supported in part by DoD, Air Force Office of Scientific Research, National Defense Science and Engineering Graduate (NDSEG) Fellowship, 32 CFR 168a. Hiro Ito was supported in part by JSPS KAKENHI Grant Number 24650006 and MEXT KAKENHI Grant Number 24106003. Anna Lubiw was supported in part by NSERC. Ryuhei Uehara was supported in part by JSPS KAKENHI Grant Number 23500013, 26330009 and MEXT KAKENHI Grant Number 24106004. Yushi Uno was supported in part by KAKENHI Grant numbers 23500022 and 25106508.

[1] R. Uehara, Stamp foldings with a given mountain-valley assignment, in: Origami^5: Proceedings of the 5th International Meeting of Origami Science, Mathematics, and Education (AK Peters/CRC Press, 2011), 2011, pp. 585–597.

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

[3] E. M. Arkin, M. A. Bender, E. D. Demaine, M. L. Demaine, J. S. B. Mitchell, S. Sethia, S. S. Skiena, When can you fold a map?, Computational Geometry: Theory and Applications 29 (1) (2004) 23–46.

[4] E. D. Demaine, J. O'Rourke, Geometric Folding Algorithms: Linkages, Origami, Polyhedra, Cambridge University Press, 2007.

[5] B. Gallivan, Folding paper in half 12 times: An 'impossible challenge' solved and explained, Historical Society of Pomona Valley (2002).
URL http://pomonahistorical.org/12times.htm

[6] R. Uehara, On stretch minimization problem on unit strip paper, in: 22nd Canadian Conference on Computational Geometry (CCCG), 2010, pp. 223–226.

[7] T. Umesato, T. Saitoh, R. Uehara, H. Ito, Y. Okamoto, The complexity of the stamp folding problem, Theoretical Computer Science 497 (2013) 13–19.

[8] M. R. Garey, D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman & Co., 1979.