Chapter 1

The Complexity of Flat Origami

Marshall Bern *

Barry Hayes †

Abstract

We study a basic problem in mathematical origami: determine if a given crease pattern can be folded to a flat origami. We show that assigning mountain and valley folds is NP-hard. We also show that determining a suitable overlap order for flaps is NP-hard, even assuming a valid mountain and valley assignment.

1 Introduction

Origami, the centuries-old art of folding paper into sculpture, is currently enjoying a renaissance. Contemporary origami artists invent new models of great beauty and intricacy. To achieve these stunning results, artists such as Engel, Fuse, Lang, and Maekawa have taken a geometric approach to origami design. One useful technique, incorporated into Lang's *TreeMaker* program, uses the centers of non-overlapping disks to determine the tips of "flaps" [10, 11]. Another technique [1, 12] builds complicated crease patterns out of repeating blocks called "molecules".

Alongside this "technical" approach to origami design, some mathematicians have started to study origami. Huffman [5] gives relations between face angles in polyhedral models, using an approach related to network flow. A number of authors [4, 6, 7, 9] have discovered angle conditions for an origami to fold flat in the neighborhood of a single vertex. In this paper, we extend this study to "global flat foldability". Although this study is entirely theoretical, it is not as far removed from practice as it may first appear. Most models fold flat up until the finishing steps, and some, including the traditional crane, fold flat even after completion.

We consider three versions of the problem of flat foldability. In each case, the input is a planar straightline graph drawn on a square; edges represent the locations of folds. If we only require that the origami fold flat within a neighborhood of each vertex, the problem turns out to be easy. For this (rather artificial) problem, we give a linear-time algorithm that assigns "mountain" and "valley" orientations and determines an overlap order for flaps. If we require that the origami fold flat everywhere, it turns out to be NP-hard [3] to find appropriate orientations and overlap order. Finally, if orientations are given, just finding an overlap order is NP-hard.

Together our results show that the real difficulty of the problem does not lie in simultaneously handling all vertices, but rather in avoiding edge-edge collisions.

2 Definitions

With some slight deviations, our definitions follow those of other authors [4, 9].

DEFINITION 1. A crease pattern is a finite planar straight-line graph drawn on a convex planar region (the paper). A crease is an edge of the planar graph.

Unless we state otherwise, we shall assume that the paper is square. The operative part of the next definition is the requirement that embeddings be oneto-one, modeling the fact that (physical) paper cannot penetrate itself.

DEFINITION 2. An **embedding** is a continuous, one-to-one mapping of a crease pattern to \mathbb{R}^3 . The mapping must be smooth (differentiable) everywhere except along creases.

An embedding need not be differentiable at creases, but along each crease the dihedral angle (defined locally) must be a smooth function. It is convenient to measure dihedral angles by deviation from flatness, so that sharp folds are close to π or $-\pi$ radians.

DEFINITION 3. A flat origami is an infinite sequence of embeddings of the same crease pattern, such that the images of each crease converge to a line segment and the images of each face converge to a planar polygonal region, congruent to the face. (Convergence is not just pointwise, but sufficiently strong that metric properties converge as well.) Moreover, the dihedral along each crease must converge uniformly to either π or $-\pi$.

We shall intuitively talk about flat origamis as single embeddings rather than sequences of embeddings,

^{*}Xerox PARC, 3333 Coyote Hill Rd., Palo Alto, CA 94304. bern@parc.xerox.com

[†]ParcPlace-Digitalk, Inc., 999 E. Arques Ave., Sunnyvale, CA 94086. bhayes@cs.stanford.edu

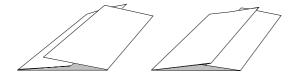


Figure 1: Two origamis with the same MV-assignment but different overlap maps.

with the understanding that the statement holds in the limit. We shall call a crease pattern flat foldable if it is the crease pattern of some flat origami. A crease of a flat origami is called a mountain if its limiting dihedral is $-\pi$ and a valley if its limiting dihedral is π . Intuitively, a mountain points up and a valley points down if the paper is unfolded back to a square.

A flat origami can be described by its MV-assignment, a mapping from the set of creases to the set $\{-\pi, \pi\}$. This description, however, is not complete, as origamis with the same MV-assignment may overlap differently, as shown in Figure 1. To give a complete description, we orient the flat origami so that all its faces are parallel to the xy-plane and then project all creases, along with the boundary of the paper, down onto the plane. This forms a cell complex with $O(n^2)$ cells, where n is the number of creases. The overlap map gives the vertical order of origami faces above each cell of this arrangement.

The overlap map can be encoded with $O(n^2)$ bits by recording for each pair of origami faces, which face lies on top; the vertical relationship between pairs remains constant throughout the origami. However, we know of no crease pattern with more than $2^{O(n\log n)}$ different flat origamis. Figure 2 gives a $2^{O(n\log n)}$ example. A few initial folds make the square paper into a rectangle, which is then folded into a U-shape (shown opened up a bit for clarity). The lower arm is pleated vertically and the upper horizontally, so that each of the $\Omega(n)$ tabs on the lower arm can be tucked into any of the $\Omega(n)$ slots on the upper. Thus the crease pattern of this folding has $n^{\Omega(n)} = 2^{\Omega(n\log n)}$ different flat origamis.

3 Single Vertex Flat Folding

We first review some well-known necessary conditions for crease patterns to fold flat in the neighborhood of a single vertex. Let v be an interior vertex of crease pattern \mathcal{C} , and assume that there is a flat origami with crease pattern \mathcal{C} . Center a small sphere at v, such that the intersection of the flat origami and the sphere is a flat spherical polygon P_v , as in Figure 3. Convex and reflex vertices of P_v correspond to mountains and valleys (shown as dotted and dashed lines, respectively), and the arc lengths of P_v are proportional to the angles

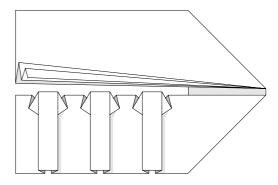


Figure 2: There are $2^{\Omega(n \log n)}$ ways to tuck the tabs into the accordion.

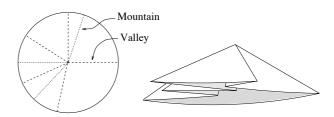


Figure 3: A circle around an interior vertex in a crease pattern folds to a flat spherical polygon.

between creases meeting at v. The requirement that the flat polygon closes up has the following consequences. See [4, 6, 8, 9].

(K1) The sum of alternate angles around v is π .

(M) The number of mountain folds minus the number of valley folds meeting at v is either 2 or -2.

Condition (K1)—"Kawasaki's theorem"—implies that a flat-foldable crease pattern is a convex subdivision. Condition (M)—"Maekawa's theorem"—implies that v must be incident to an even number of creases. Now let the creases around v be e_1, e_2, \ldots, e_k , with $e_1 = e_{k+1}$. Let the angle between e_i and e_{i+1} measure α_i . The requirement that polygon P_v not cross itself has the following consequence [7, 9].

(K2) If $\alpha_i < \alpha_{i-1}$ and $\alpha_i < \alpha_{i+1}$, then e_i and e_{i+1} must have opposite assignments.

If v is a boundary vertex of the crease pattern, lying along an edge or at a corner of the paper, P_v will be an open polygonal chain, and hence only condition (K2) applies. For boundary vertices we define α_1 and α_k to be the angles between the first and last creases and the edge of the paper; we do not consider these angles to be adjacent.

We now consider the converse: sufficient conditions. Kawasaki [9] and Justin [7] independently proved that (K1) alone is a sufficient condition for a single-vertex crease pattern to fold flat. We give a proof similar to Justin's [7], an algorithm that will find use in Section 4.

THEOREM 3.1. (KAWASAKI, JUSTIN) Let \mathcal{D} be a crease pattern drawn on a disk, consisting of a single vertex v at the center of the disk, along with some number of creases, each a radius of the disk. Then \mathcal{D} is flat foldable if the sum of alternate angles around v is π .

Proof. Again let $e_1, e_2, \ldots, e_k, e_{k+1} = e_1$ denote the creases in order around v, and let α_i denote the angle between e_i and e_{i+1} .

The algorithm uses condition (K2) recursively. The algorithm finds a locally minimal angle α_m , that is, an m such that $\alpha_m \leq \alpha_{m+1}$ and $\alpha_m \leq \alpha_{m-1}$. It then subtracts α_m from α_{m+1} , removes e_m and e_{m+1} , and merges α_{m-1} and the remainder of α_{m+1} into one angle. These steps cut a pie-shaped wedge from \mathcal{D} and produce a new crease pattern \mathcal{D}' . The algorithm then calls itself recursively to compute a flat origami for \mathcal{D}' . (Here we have slightly generalized the notion of crease pattern, since the angles in \mathcal{D}' do not sum to 2π .) The removed wedge can be attached to the flat origami for \mathcal{D}' in either one of two ways, corresponding to the two possible assignments for e_m and e_{m+1} : M and V or V and M. See Figure 4. In either case, the wedge should be next to the side to which it is attached in the vertical order, or else the flat polygon may self-intersect.

The recursion bottoms out when there are only two creases left; (K1) implies that the last two angles must be equal. The algorithm assigns the creases to be both M or both V; this free choice determines which side of the spherical polygon P_v is the interior. \Box

The algorithm also solves the problem of flat folding an arbitrary boundary vertex. In this case, the recursion can bottom out with zero, one, or two creases left. The last assignment is arbitrary, unless there are two creases left and the minimum angle is between the two creases.

Notice that in this algorithm each crease e_i is paired with a partner e_j , such that e_i and e_j have opposite assignments or—in the case of the last pair—the same assignments. Other choices—which of e_i and e_j is M, and which locally minimal angle to process next—are unconstrained.

It is not hard to implement this algorithm so that it runs in linear time. An initial linear-time pass finds all locally minimal angles and places them on a queue. After removing the first angle and merging the two neighboring angles, we update the queue by removing angles that are no longer locally minimal and adding angles that have become locally minimal—O(1) operations in all.

4 All Vertices Simultaneously

We now have a condition—that alternate angles sum to π —which determines whether there exists an origami that is flat in the neighborhood of a single vertex. For the remainder of this paper, we shall assume that all crease patterns satisfy this angle condition at each interior vertex. In this section, we take a halfway step towards the problem of deciding the global flat foldability of a crease pattern.

DEFINITION 4. A vertex-flat assignment is an MV-assignment to a crease pattern such that for each vertex this assignment is the assignment of a flat single-vertex origami.

A vertex-flat assignment is necessary but not sufficient for the existence of a flat origami for the entire crease pattern, because folded edges may collide at points far away from vertices. A slight modification of the "letter" in Figure 1 gives an example of a crease pattern with a vertex-flat assignment but no flat origami: simply make the middle face of the letter narrower than either of the other faces.

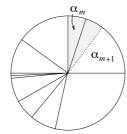
Definition 5. Vertex Flat Foldability is the problem of determining whether or not a given crease pattern has a vertex-flat assignment.

We now give a polynomial-time algorithm for Vertex Flat Foldability. For "yes" instances of the problem, this algorithm finds an MV-assignment that satisfies conditions (M) and (K2) (and also (K2) applied iteratively as in the single-vertex algorithm). Incidentally, (M) alone gives the following interesting matching problem: assign M and V to edges such that at each vertex the number of M edges minus the number of V edges is plus or minus two. This problem is polynomially solvable even for general graphs (see [13], page 389).

Theorem 4.1. There is a linear-time algorithm for Vertex Flat Foldability.

Proof. We start by imposing a general position condition that we shall later remove: at each vertex v in crease pattern \mathcal{C} , at each step of the single-vertex algorithm given in the proof of Theorem 3.1, no angle has the same measure as one adjacent to it.

We use the pairing of creases in the single-vertex algorithm to group creases of C, that is, partition them into disjoint sets. If e_i is the partner of e_j at vertex v, then we place e_i and e_j into the same group. (This grouping is similar to the "origami line graph" of Hull [4] and Justin [7].) Since a crease has exactly one partner at an interior vertex, and at most one partner at a



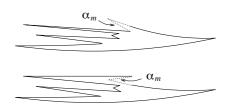
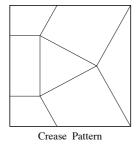
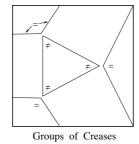


Figure 4: A recursive algorithm for folding a single-vertex flat origami.





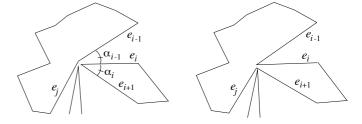


Figure 5: A crease pattern with an inconsistent cycle.

Figure 6: Groups formed by tie-breaking choices can be merged.

boundary vertex, each group forms either a cycle or a path. Assigning M or V to any single crease within a group sets all the creases. A cycle—but never a path—may be self-contradictory, or *inconsistent*, for example, an odd cycle in which each pair of partners must be oppositely assigned. Figure 5 shows an example due to Hull [4] of a crease pattern with an inconsistent cycle.

We now remove the general position condition. Assume that at each vertex, we run the single-vertex algorithm as before and obtain some grouping of creases into cycles and paths. Our plan is to merge two groups that meet at a tie. Assume the following: at some step of the single-vertex algorithm, crease e_i forms the same angle with e_{i-1} as it does with e_{i+1} ; initially e_i was paired with e_{i+1} and e_{i-1} with e_j ; and e_i and e_{i-1} belong to two different groups. If e_i had instead been paired with e_{i-1} , e_{i+1} would have been paired with e_j , because e_{i+1} forms the same angle with e_j that e_{i-1} did after subtracting α_i from α_{i+1} . By changing the pairing, we can merge e_i and e_{i-1} 's groups, as shown in Figure 6.

This merging procedure can be applied repeatedly in the case of more than one tie. In fact, it is not hard to see that any valid initial grouping, followed by a maximal sequence of mergers, yields the same final grouping.

It is never disadvantageous to merge an inconsistent cycle with another group. If the other group is a path, the result of the merger is a larger path, hence consistent. If the other group is a consistent cycle, the result is a larger inconsistent cycle with at least as many opportunities for mergers. And if the other group is another inconsistent cycle, the inconsistencies eliminate each other and the result is a consistent cycle.

We can now fill in the complete algorithm. Run the linear-time single-vertex algorithm, breaking ties arbitrarily. Form groups of edges with a linear-time traversal of the graph. Now try to resolve inconsistent cycles with mergers. Each merger causes a "union" operation in a union-find data structure [14] representing the groups and changes O(1) pointers in a planar graph data structure representing the crease pattern. Because unions occur only between edges adjacent around a vertex, the overall running time is linear [2]. Pattern $\mathcal C$ has a vertex-flat assignment if and only if all inconsistent cycles can be eliminated. \square

5 Flat Foldability

This section and the next give NP-hardness reductions. In each case, we reduce the following NP-complete problem [3] to the origami problem.

DEFINITION 6. NOT-ALL-EQUAL 3-SAT is given by a collection of clauses, each containing exactly three literals. The problem is to determine whether or not there exists a truth assignment such that each clause has either one or two true literals.

Our first target problem is FLAT FOLDABILITY, the canonical flat-origami problem. It is not clear whether FLAT FOLDABILITY is in the class NP, because it may take very high precision arithmetic to check the validity of an overlap map.

DEFINITION 7. FLAT FOLDABILITY is the problem of determining whether or not a given crease pattern is the crease pattern of a flat origami.

As usual in reductions to versions of SAT, we construct "gadgets" for boolean variables and clauses, which we interconnect by "wires". For us, a wire will be two closely-spaced parallel creases. The spacing is close enough that in any flat folding the two creases in a wire must have opposite assignments, forming a "pleat". In order to distinguish left from right, we shall label the wires in our gadgets with directions. These directions serve only as expository devices; they are not part of the crease pattern. We shall call a wire in an MV-assignment true (respectively, false) if the valley crease lies to the right (left) of the mountain crease when facing along the wire's direction.

Figure 7(a) shows a clause gadget. This crease pattern consists of three wires that meet at a central equilateral triangle. We think of all three wires as directed into the triangle. There are eight MV-assignments that satisfy conditions (M) and (K2), the four shown in Figure 7(b) and the four obtained from these by reversing M and V. However, the rightmost assignment is not flat foldable, because its three mountain edges collide at a point above—out of the plane of—the triangle. Of course, the same is true of its reversal. We omit proofs of the claims in our NP-hardness reductions; they are all straightforward.

LEMMA 5.1. The clause crease pattern is flat foldable if and only if one or two of the incoming wires are true.

We construct the truth-setting part of our reduction out of gadgets called *reflectors*. A reflector crease pattern consists of three wires that meet at an isosceles triangle with largest angle in the range [90°, 180°), as shown in Figure 8. By varying the angle of the isosceles triangle, we obtain a one-parameter family of reflectors.

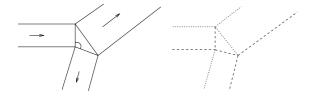


Figure 8: Reflector crease pattern.

LEMMA 5.2. The reflector crease pattern is flat foldable if and only if the incoming wire agrees with the outgoing broad wire and disagrees with the outgoing narrow wire.

The last gadget, called a *crossover* and shown in Figure 9, lets wires cross while preserving their truth settings. The angles at which the wires meet the central parallelogram are chosen so that each wire folds over or under the parallelogram's center, forcing continuity of truth settings. Our reduction uses crossovers of 90° and 135° , as shown.

LEMMA 5.3. The crossover crease patterns are flat foldable if and only if each opposite pair of incoming and outgoing wires agree.

We now put these gadgets together. Figure 10 shows a schematic of the entire reduction. Gadgets are spaced quite far apart relative to the width of wires, so that any flat origami with this crease pattern would look like a slightly wrinkled square sheet of paper. At the top of the paper, we place one clause gadget for each clause in the boolean formula. Each wire entering a clause gadget has width $\lambda>0$.

At the bottom of the paper, we place one truthsetting construction for each boolean variable. truth-setting construction includes a right-angled zigzag formed by a wire of width $\lambda/\sqrt{2}$ with reflectors at each turn. We consider zig-zags to be directed from left to right. The orientation of the folds—zigs true and zags false or the other way around—determines whether the boolean variable is true or false. For each use of boolean variable x in a clause, we send a wire of width λ upwards from a reflector along the zig-zag for x. To obtain uncomplemented x, we can either take the signal from an upper reflector and use no additional reflectors on the path to the clause gadget or take the signal from a lower reflector and use three additional reflectors on the path to the clause. Similarly, there are two ways to obtain \bar{x} . Additional reflectors send out extraneous "noise" wires.

It is not hard to confirm that the crease pattern just described will be flat foldable exactly when the original boolean formula has an assignment that makes one or two literals true in each clause. A final point concerns the encoding of our construction. As we have explained our gadgets, some vertices must have irrational coordinates. However, it is should be intuitively clear that the constructed crease pattern can be described by rational distances along a grid containing horizontal, vertical, and diagonal lines with O(1) different slopes. Overall the total number of bits will be only polynomial in the size of the Not-All-Equal 3-SAT instance.

THEOREM 5.1. FLAT FOLDABILITY is NP-hard.

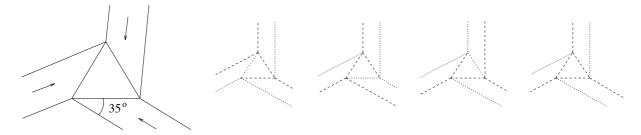


Figure 7: (a) The crease pattern of the clause gadget. (b) Possible assignments.

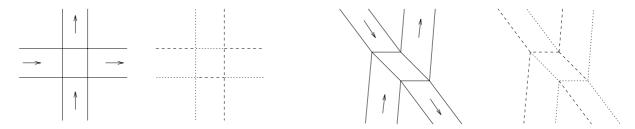


Figure 9: Crossovers of 90° and 135° .

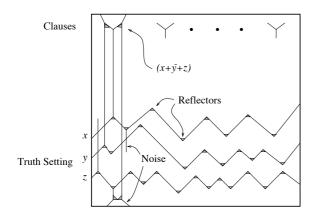


Figure 10: Schematic of the entire reduction.

6 Assigned Flat Foldability

In this section we prove that the origami problem remains hard even if we are given an assigned crease pattern, that is, a crease pattern along with a valid MV-assignment. We again use NOT-ALL-EQUAL 3-SAT as our starting problem. This reduction is quite intricate, so we shall omit some details.

Definition 8. Assigned Flat Foldability is the problem of determining whether or not there exists a flat origami with a given assigned crease pattern.

The overall strategy is similar to the previous reduction. A wire will now consist of four parallel creases, with M, V, V, and M orientations. A wire is *true* if the left pleat is folded over the right pleat and

false if the right is folded over the left, again defining left and right relative to nominal directions. We make the distance between the two pleats—the middle channel of three parallel channels—slightly larger than the width of the pleats themselves.

This time our gadgets use a subgadget called a *tab*, shown in Figure 11. The crease pattern of a tab has only one valid folding; this folding includes a rectangular flap. Such a flap will only fit into a "pocket" deep enough to accommodate it; we shall use this fact to restrict the set of possible overlap maps for our gadgets.

Tabs send out "noise" pleats that cross over signal wires. Fortunately, we can allow noise pleats to cross wires arbitrarily without any danger of signals inverting. A change in truth assignment would cause the two M creases to interpenetrate. Thus we shall think of tabs as built into the paper, and we do not show the noise pleats in subsequent figures.

Lemma 6.1. The truth assignment of a wire cannot change when it crosses over a pleat or another wire.

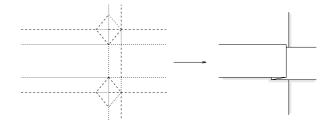


Figure 11: A tab is a rectangular flap.

We can make an arbitrarily long and thin tab—think of a diving board—by folding tabs "on top of" tabs. That is, the *base* of one tab—the attached end of the diving board—serves as the *tip*—the free end—of the next tab. The assigned crease pattern of this construction has only one valid folding; the resulting tab has the same width but twice the length of a single tab, and is twice as thick at its base.

The assigned crease pattern for the new clause gadget is shown in Figure 12(a). The points marked with dots fold to be very nearly coincident. They would be exactly coincident if the three widths within a wire were all the same; as is, they leave a small window at the center of the folded gadget, as shown in Figure 12(b). If we add tabs, represented by open boxes in Figure 12(a), to the sides of this window, we can eliminate the two "spiral" foldings, which correspond to the all-true and all-false settings. How does this work? The lowest tab in Figure 12(a) folds to the position shown in Figure 12(b). If the wire coming in from the left has its lower (i.e., lower-on-the-page) flap over its upper, then this tab must lie above the opposing side e of the windowframe; if the incoming wire has the opposite truth assignment then the tab must lie one level lower, in a pocket below the windowframe and above the gadget's equilateral and nearly-equilateral triangles. If two tabs both lie at the same level, the more-clockwise tab must lie over the less-clockwise. Hence the two overlap maps that place all three tabs at the same level are impossible. We summarize:

LEMMA 6.2. The assigned crease pattern of a clause gadget is flat foldable if and only if one or two of the incoming wires are true.

We next design a signal *splitter*, the gadget that takes the place of the reflector. The splitter starts with the same crease pattern and MV-assignment as the clause gadget, but uses tabs to eliminate all foldings *except* the two spirals, forcing the two output wires to disagree with the one input wire. This construction is even more intricate than the last one, so we shall start by building some intuition.

Imagine cutting a hexagon with six arms attached from a sheet of paper, as shown in Figure 13(a). For this cut-out to fold flat with each side of the hexagon forming a valley fold, shaded and unshaded sets of arms must each form spirals. Figure 13(b) shows the shaded arms folded in a counterclockwise spiral, c' above b' above a' above c'. If all unshaded arms were folded, rather than just arm b, they would form a clockwise spiral lying overtop the shaded spiral. There are exactly two overlap maps for the hexagon: shaded spiral below unshaded spiral, and vice versa.

The hexagon we need is not exactly a regular hexagon. Angles alternate between a little less than 120° and a little greater than 120° , so that arms a' and b (and similarly, b' and c and c' and a) overlap when folded, with the overlap starting just beyond the center of the hexagon. Thus even the pairs that fold to be nearly parallel indicate which spiral is on top; for example, the fact that b lies above a' in Figure 13(b) tells us that c and a must fold to lie above b' and c'. Finally, pairs of nearly-parallel arms are long enough that they eventually cross, as shown in the figure.

The key idea in our splitter construction is to add tabs inside the window of the clause gadget in the form of the hexagon arms described above. The truth setting of the wire coming in from the left—the input wire—determines the overlap of the tabs marked a' and b in Figure 14(a). The other four tabs then force the settings of the output wires. The two possible foldings of the splitter are shown in Figures 14(b) and (c). In these two pictures, the bold curved lines indicate the positions of tabs a' and b.

We now explain how this construction works. Tabs a' and b are sufficiently long that their tips—the ends of the "diving boards"—must be topmost faces in the overlap order. The tabs, however, do not pass directly out of the window; each has one mountain and one valley fold so that it forms a Z shape as seen from the side. The tabs' mountain and valley folds coincide, so that either tab may be folded inside the other. Each tab's initial section (from base to mountain fold) is sufficiently long that it cannot fit just above the equilateral-triangle floor of the window and must find some other pocket.

The widths and placements of tabs a' and b are also critical. Tab a' is too wide to fit out of the window directly. In other words, the initial section of a' lies underneath the corner of the windowframe marked 1. Tab b is wide enough that both its initial and final sections overlap vertex 2; tab b runs essentially parallel to the long skinny triangle incident to this windowframe vertex. Tabs a' and b eventually cross so that out near their tips a' but not b overlaps vertex 3.

First assume that the initial section of a' lies over the initial section of b. See Figure 14(b). Then the M folds of both tabs must lie in a pocket just under the upper flap of the incoming wire, with a' folded inside b. Since the upper flap of the incoming wire lies over the initial section of tab a', which lies over tab b, which in turn lies over vertex 2, the lower flap of the incoming wire must lie below the upper flap.

Now assume that the initial section of b lies over the initial section of a'. See Figure 14(c). Then the final section of b must also lie over the final section of a', which implies that vertex 3 must lie over the final

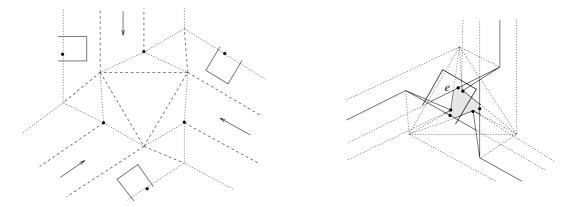


Figure 12: (a) Clause gadget for Assigned Flat Foldability. (b) Folded (only one tab shown).

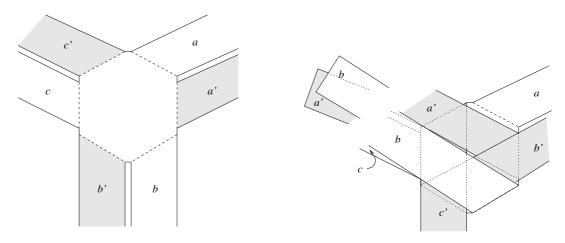


Figure 13: (a) A hexagon cutout with exactly two foldings. (b) Partially folded.

section of a'. This implies that the lower flap of the incoming wire must lie over the upper flap, since the final section of a' must eventually emerge as a topmost face

LEMMA 6.3. The assigned crease pattern of a splitter is flat foldable if and only if the two outgoing wires disagree with the one incoming wire.

Using clause gadgets and splitters, we can create all the logic elements we need. For example, we can build an inverter by splitting a signal into two wires which are then fed into a clause gadget. Extra splitters are used to turn the signal; a complete inverter contains five splitters and one clause gadget laid out in the form of a hexagon.

Finally, the overall construction is similar to the unassigned case. However, because we no longer have reflectors of arbitrary angles, we cannot take signals from the bottom of zig-zags and bounce them back upwards. Instead we run a 120° zig-zag for each of x and \bar{x} , with the two zig-zags linked by an inverter.

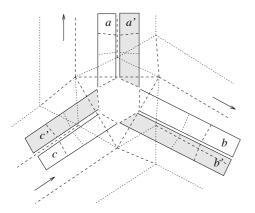
Alternatively, we could start from the problem Non-Negated Not-All-Equal 3-SAT, in which each literal is uncomplemented.

THEOREM 6.1. ASSIGNED FLAT FOLDABILITY is NP-hard.

7 Conclusion

The mathematical study of origami is fairly new, so we close by suggesting some open problems for future research.

- 1. How many different flat origamis can there be with the same crease pattern? This question is related to the problems of "stamp" and "map" folding.
- 2. Does Flat Foldability remain NP-hard for special inputs? Our reduction uses only degree-4 vertices, but perhaps some other restriction renders the problem easy.
- 3. Our use of tabs in the reduction for Assigned Flat Foldability is somewhat unesthetic. Are



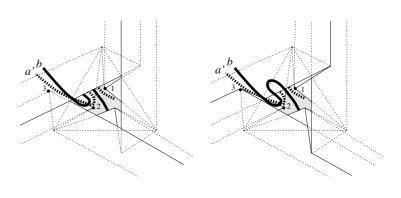


Figure 14: (a) Signal splitter. (b) With the incoming wire folded as shown, a' must lie over b. (c) The other truth assignment forces b over a'.

they necessary? If "tabs" can somehow be ruled out, does the problem become polynomially solvable?

- 4. Is every simple polygon, when scaled sufficiently small, the silhouette of a flat origami? How many creases are necessary to fold an *n*-vertex polygon? How thick (number of layers of paper) must the origami be? This last question is motivated by the fact that in practice it is very difficult to simultaneously fold a large number of layers.
- 5. Which simple polygons are silhouettes of flat origamis? In this question, the polygon has a fixed size.
- 6. K-Layer Flat Foldability is the problem of determining, for a given crease pattern, whether or not there exists a flat origami that is at most k layers thick. Our reduction above shows that this problem is NP-hard for $k \geq 7$. What about k from 2 to 6? Similar questions can be posed for Assigned Flat Foldability.

Acknowledgments

We would like to thank Brian Ewins, Tom Hull, Jacques Justin, and Robert Lang for valuable comments.

References

- [1] P. Engel. Folding the Universe: Origami from Angelfish to Zen. Vintage Books, 1989. Dover, 1994.
- [2] H. Gabow and R.E. Tarjan. A linear-time algorithm for a special case of the disjoint set union. In Proc. 15th Annual ACM Symp. on Theory of Computing, 1983, 246–251.
- [3] M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Co., 1979.
- [4] T. Hull. On the mathematics of flat origamis. Congressus Numerantium 100 (1994), 215–224.
- [5] D.A. Huffman. Curvature and creases: a primer on paper. *IEEE Trans. on Computers* Volume C-25 (1976), 1010–1019.
- [6] J. Justin. Mathematics of Origami. British Origami 110 (Feb. 1985) to 118 (June 1986).
- [7] J. Justin. Towards a mathematical theory of origami. Manuscript, 1994. Abstract in 2nd Int. Meeting of Origami Science, 1994, Otsu, Japan.
- [8] K. Kasahara and T. Takahama. Origami for the Connoisseur. Japan Publications, Tokyo, 1988.
- [9] T. Kawasaki. On the relation between mountaincreases and valley-creases of a flat origami. In *Origami* Science and Technology. H. Huzita, ed., 1989, 229–237.
- [10] R. Lang. Mathematical algorithms for origami design. Symmetry: Culture and Science 5 (1994), 115–152.
- [11] R. Lang. The tree method of origami design. Manuscript, 1995.
- [12] R. Lang. A synopsis of the universal molecule. Manuscript, 1995.
- [13] L. Lovász and M.D. Plummer. Matching Theory. Annals of Discrete Mathematics 29, North-Holland, 1986.
- [14] R.E. Tarjan. Data Structures and Network Algorithms. SIAM, 1983.