

International Journal of Computational Geometry & Applications
© World Scientific Publishing Company

FOLDING EQUILATERAL PLANE GRAPHS*

ZACHARY ABEL

*MIT Department of Mathematics,
77 Massachusetts Ave., Cambridge, MA 02139, USA
zabel@math.mit.edu*

ERIK D. DEMAINE

MARTIN L. DEMAINE

SARAH EISENSTAT

JAYSON LYNCH

TAO B. SCHARDL

*MIT Computer Science and Artificial Intelligence Laboratory,
32 Vassar St., Cambridge, MA 02139, USA
{edemaine,mdemaine,seisenst,jaysonl,neboat}@mit.edu*

ISAAC SHAPIRO-ELLOWITZ

*University of Massachusetts Boston,
100 Morrissey Blvd., Boston, MA 02125, USA
isaac.shapiroello001@umb.edu*

Received May 26, 2012

Revised December 20, 2012

Communicated by Zachary Abel

We consider two types of folding applied to *equilateral* plane graph linkages. First, under continuous folding motions, we show how to reconfigure any *linear* equilateral tree (lying on a line) into a canonical configuration. By contrast, it is known that such reconfiguration is not always possible for linear (nonequilateral) trees and for (nonlinear) equilateral trees. Second, under instantaneous folding motions, we show that an equilateral plane graph has a noncrossing linear folded state if and only if it is bipartite. Furthermore, we show that the equilateral constraint is necessary for this result, by proving that it is strongly NP-complete to decide whether a (nonequilateral) plane graph has a linear folded state. Equivalently, we show strong NP-completeness of deciding whether an abstract metric polyhedral complex with one central vertex has a noncrossing flat folded state. By contrast, the analogous problem for a polyhedral manifold with one central vertex (*single-vertex origami*) is only weakly NP-complete.

Keywords: Folding; linkages; plane graphs.

*A preliminary version of this paper appeared in *Proceedings of the 22nd International Symposium on Algorithms and Computation (ISAAC 2011)*, LNCS 7074, 2011, pages 574–583.

2 *Abel, Demaine, Demaine, Eisenstat, Lynch, Schardl, Shapiro-Elowitz*

1. Introduction

This paper is motivated by two different types of problems related to folding: (1) linkage folding, specifically locked trees; and (2) paper folding, specifically single-vertex origami.

1.1. *Locked Trees: Not if Equilateral and Linear*

Biedl et al.⁶ introduced the notion of a “locked tree” and gave the first example thereof. Here a *tree* refers to a plane tree linkage, that is, a tree graph with specified edge lengths and a preferred planar embedding. Such a linkage can *move* or *fold continuously* subject to the constraints that the edges remain straight line segments of the specified lengths and the edges never properly cross each other.⁷ A tree is *universally foldable* if it can be folded continuously from any configuration to any other.⁶ Equivalently, a tree is universally foldable if it can be folded from any configuration into a *canonical* configuration, in which the edges lie along a horizontal line and point rightward from a single root vertex. Otherwise, the tree is *locked*. We say that a particular tree configuration is *locked* if it cannot be folded to a canonical configuration. While the complexity of testing lockedness of a tree or tree configuration is unknown, the related problem of deciding whether two given configurations of a tree are connected by a continuous motion is PSPACE-complete.³

Beidl et al.⁶ gave a specific example of a locked tree configuration in 2002. In 2006, Poon¹⁴ provided a simpler example and conjectured that any *equilateral* tree, in which all edges lengths are equal, is universally foldable. However, Ballinger et al.⁴ later constructed a locked equilateral tree configuration. Furthermore, their paper produced the first examples of *linear* locked tree configurations, where all edges initially lie along a line. The constructed examples of linear locked tree configurations and equilateral locked tree configurations seem very different; for example, the locked equilateral tree configuration has no touching bars (except at common endpoints), while the locked linear tree configurations necessarily have overlapping bars. Thus it is natural to wonder whether there are locked tree configurations that are simultaneously linear and equilateral.

Our results. We settle this question by showing that every linear equilateral tree configuration can be folded into a canonical configuration. As a consequence, any equilateral tree can be folded between all of its linear configurations. This result may be considered an analogue of *flat-state connectivity*^{1,2}, previously considered for 3D fixed-angle chains. Our proof of this result, given in Section 3, builds up a progressively more canonical configuration by repeatedly fixing any deviations. To keep track of the overall structure of the linkage, we introduce the notion of a *plane homomorphism* to enable manipulating multiple overlapping edges as one.

1.2. Single-Vertex Origami: Generalization

A classic structure in mathematical origami is the *single-vertex crease pattern*—a circular piece of paper with creases all emanating from the center. This special case is useful because it effectively models the local behavior of a general crease pattern in a small neighborhood around a vertex. Kawasaki's Theorem^{13,11,10} describes precisely when a single-vertex crease pattern can fold flat using exactly the prescribed creases, which can be tested in linear time.

As described in Ref. 10, flat folding single-vertex crease patterns can be viewed as folding a cycle linkage into a linear configuration, subject to the constraint of bending at every vertex. Each edge of the cycle linkage corresponds to a pie wedge of the crease pattern, and the edge length equals the pie angle. Although flat pieces of paper have the extra property that the lengths/angles sum to 360° , the characterization of flat foldability has been generalized to arbitrary cycles.^{10,15}

Recall that the question of locked trees deals with *continuous* motions. Continuous motions of the cycle linkage, corresponding to *single-vertex rigid origami*, are in fact always possible.¹⁵ For this section, however, we focus on *instantaneous* motions, asking whether a linear configuration exists instead of how to get there. Our motivation is that regular (nonrigid) origami allows the paper to bend anywhere during the motion, so long as it is creased only where we want at the end. For polygonal paper (not just single vertex), any configuration can be reached by such a flexible folding motion.⁹

We consider the generalized problem of instantaneous foldings of plane graphs (instead of just cycles) into linear configurations. Mapped to the context of origami, this problem is equivalent to finding flat foldings of *single-vertex complexes*, which consist of pie wedges with a common apex, sharing some edges, and bounded by great circular arcs on the surface of a sphere centered at the common apex. Figure 1 shows a simple example. This situation models the local behavior of a vertex neighborhood in a polyhedral complex (3D polygons attached at edges or vertices).

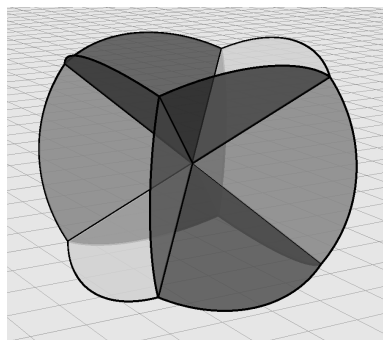


Fig. 1. A single-vertex complex with one pie wedge per edge of a cube.

Our results. For the special case of equilateral plane graphs, we prove that instantaneous folding into a linear configuration is possible if and only if the graph is bipartite. Bipartiteness is an obvious necessary condition: any linear configuration of a cycle naturally assigns each edge of the cycle to one of two directions, so the number of edges in any cycle must be even. The interesting result, shown in Section 4, is that all bipartite equilateral plane graphs have a linear configuration.

Interpreted in the context of single-vertex complex origami, this result says that

a single-vertex complex in which all pie angles are equal can be folded flat if and only if every spherical region has an even number of edges. For example, Figure 1 satisfies both conditions, so it folds flat. We can even require that the flat folding uses all of the given creases: any linear equilateral configuration can be folded down so that all angular regions are collocated.

Finally, we prove that these results can hold only for the equilateral situation, in a strong sense. Specifically, Section 5 shows that finding a linear configuration becomes strongly NP-complete for nonequilateral plane graphs, even when edge lengths are restricted to the set $\{1, 2\}$. Our reduction is from planar monotone 3-SAT, which was recently shown to be NP-complete.⁸

Interpreted in the context of single-vertex complex origami, this result says that it is strongly NP-complete to determine flat foldability of an abstract metric single-vertex complex with a specified outside region. We show moreover that this technical “outside region” condition may be removed. This result suggests that there is no complex analog of the efficient Kawasaki’s Theorem characterizing the case of a single cycle, though there are a few differences. First, the constructed complex might not be embeddable on the sphere with the given edge lengths. Second, we do not require every vertex (crease) to be folded. This change makes even the cycle problem NP-complete, though only in the weak sense, as it admits a pseudopolynomial-time algorithm. Our result shows that the problem is strongly NP-complete for general graphs.

2. Definitions

In this paper, every graph $G = (V(G), E(G))$ is assumed to be equipped with positive edge weights (lengths) $\ell : E(G) \rightarrow \mathbb{R}_{>0}$ unless otherwise specified. A **plane graph** is a (weighted) graph with a preferred combinatorial planar embedding (i.e., not necessarily respecting edge lengths).

Recall that a **linkage** is a straight-line embedding of a plane graph (known as the **underlying graph**) with compatible edge lengths. A **motion** of a linkage is a continuous deformation of the linkage that preserves the lengths of edges and does not self-intersect. Intuitively, a **self-touching linkage** is a linkage that can self-intersect, but cannot combinatorially cross itself. Connelly et al.⁷ give a more formal definition of a self-touching linkage, which we use throughout this paper.

Definition 1. A **self-touching embedding** (also known as a **configuration** or **state**) of a plane graph G is a self-touching linkage L with an isomorphism identifying L ’s underlying graph with G —in particular, the planar embedding and the edge lengths agree.

Next we define the notion of a plane homomorphism, i.e., a graph homomorphism that respects the underlying combinatorial planar embeddings. This language is pivotal for describing “partially folded” states of graphs. In this definition below, a **chain** is a path of two distinct edges.

Definition 2. For two weighted or unweighted plane graphs G and H , a **plane homomorphism** $g : G \rightarrow H$ is a graph homomorphism (preserving edge weights in the weighted case) together with, for each oriented edge $e = (w_1, w_2) \in E(H)$, a linear (counterclockwise) ordering $I_g(w_1, w_2)$ around w_1 of the edges $g^{-1}(e)$ (the set of undirected edges in G mapping to e) satisfying certain compatibility and planarity constraints. For any vertex $w \in H$, the cyclic ordering of edges around w in H and the linear orders $I_g(w, \cdot)$ around w induce a cyclic order of all edges in G whose images are incident to w ; call this ordering $I_g(w)$. The compatibility and planarity constraints may be expressed as follows:

- (1) **Edge Ordering Compatibility:** For every oriented edge $e = (w_1, w_2)$ in H , the linear orders $I_g(w_1, w_2)$ and $I_g(w_2, w_1)$ are reversed, i.e., edges $g^{-1}(e)$ are linearly ordered.
- (2) **Respect for Planar Embeddings:** For each vertex $v \in G$, the cyclic order of the edges incident to v induced by $I_g(g(v))$ matches their cyclic order around v in G .
- (3) **Noncrossing:** For any two vertices $v \neq v'$ in G with $g(v) = g(v') = w$, and any two chains (e_0, e_1) and (e'_0, e'_1) centered at v and v' respectively, the induced cyclic order of these four edges around w is not e_0, e'_0, e_1, e'_1 or e_0, e'_1, e_1, e'_0 . In other words, these chains are not made to cross at w .

A plane homomorphism is **surjective** if it is surjective on vertices and edges.

Intuitively, we interpret a plane homomorphism $g : G \rightarrow H$ as a “self touching embedding of G along graph H ,” and this connection can be made explicit using the language of “magnified views” (cf. the Connelly et al.⁷ definition of self-touching linkage.) For each vertex $w \in H$, we define the **magnified view** of g around w as a graph inside a disk specified as follows: There is a terminal node inside the disk for each vertex in $g^{-1}(w)$, and a nonterminal node for each edge in G whose image is incident to w . The nonterminals are ordered around the boundary of the disk

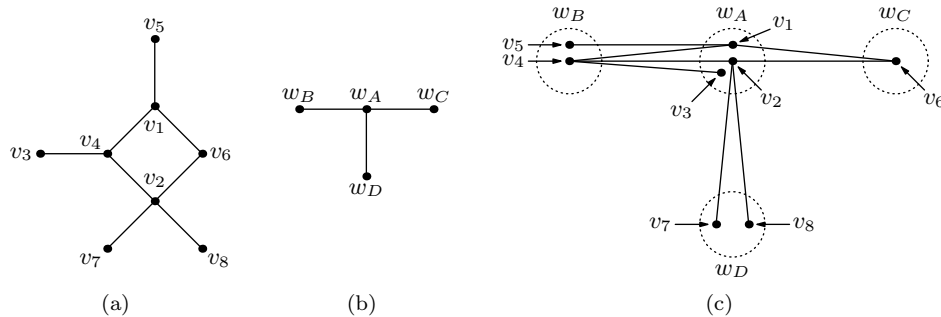


Fig. 2. An example of a plane homomorphism. Figures (a) and (b) depict the graphs G and H , respectively. Figure (c) is a schematic representation of plane homomorphism $g : G \rightarrow H$.

in the cyclic order $I_g(w)$ induced by g . Finally, this graph has an edge connecting terminal-nonterminal pairs corresponding to vertex-edge incidences in G . Then the noncrossing condition can be restated simply as follows: these magnified view graphs are planar.

Figure 2 depicts an example of a plane homomorphism. Let us discuss this figure more thoroughly to clarify the three conditions of Definition 2. As an example of condition (1), the linear order $I_g(w_A, w_B) = ((v_1, v_5), (v_1, v_4), (v_2, v_4), (v_3, v_4))$ (of unoriented edges) is the reverse of the linear order $I_g(w_B, w_A) = ((v_3, v_4), (v_2, v_4), (v_1, v_4), (v_1, v_5))$, reflecting the fact that these edges are “stacked” along edge (w_A, w_B) according to g . As an example of condition (2), observe that the cyclic order $[(v_2, v_4), (v_2, v_7), (v_2, v_8), (v_2, v_6)]$ of edges around v_2 in G is compatible with their induced order from $I_g(w_A)$, ensuring that g is consistent with the preferred embedding of G . Finally, the fact that the edges incident to v_1 and the edges incident to v_2 do not interleave in the cyclic ordering $I_g(w_A)$ illustrates condition (3) for these vertices, ensuring that these edges do not “intersect combinatorially” at w_A . The dotted circles in Figure 2(c) enclose the magnified views of g .

The following lemmas make plane homomorphisms particularly useful:

Lemma 1. *Plane homomorphisms compose: Two plane homomorphisms $g : F \rightarrow G$ and $h : G \rightarrow H$ canonically induce a plane homomorphism $h \circ g : F \rightarrow H$.*

Lemma 2. *A plane homomorphism $g : G \rightarrow H$ and a self-touching embedding M of H canonically induce a self-touching embedding of G , denoted $g^*(M)$. Furthermore, if $t \mapsto M_t$ is a valid motion of M , then $t \mapsto g^*(M_t)$ is a valid motion of $g^*(M)$.*

The proofs of these lemmas are somewhat technical and may safely be skipped upon first reading.

Proof of Lemma 1. To define plane homomorphism $h \circ g : F \rightarrow H$, we begin by setting $(h \circ g)(u) = h(g(u))$ for each vertex $u \in V(F)$ and $(h \circ g)(u, u') = h(g(u, u'))$ for each edge $(u, u') \in E(F)$. Next, we define the linear orders $I_{(h \circ g)}(w, w')$ for each oriented edge $(w, w') \in V(H)$ as follows. If $I_h(w, w') = ((v_1, v'_1), \dots, (v_k, v'_k))$, then $I_{(h \circ g)}(w, w')$ is obtained by concatenating the lists $I_g(v_1, v'_1), \dots, I_g(v_k, v'_k)$ in this order.

It remains to verify that our definition of $h \circ g$ satisfies the requirements of Definition 2. Conditions (1) and (2) for $h \circ g$ follow directly from the corresponding conditions on g and h individually. To verify the noncrossing condition, suppose we have two vertices $u \neq u'$ in F with $(h \circ g)(u) = (h \circ g)(u') = w$, and edge chains (e_0, e_1) and (e'_0, e'_1) centered at u and u' respectively. We consider two cases. If $g(u) = g(u') = v$, then the noncrossing constraint of g ensures that edges $g(e_0)$ and $g(e_1)$ are not interleaved with edges $g(e'_0)$ and $g(e'_1)$ around v in G , and condition (2) of h preserves this non-interleaved cyclic ordering around $h(v) = w$ in H . Otherwise, if $g(u) \neq g(u')$, then the result follows from the noncrossing constraint on h . \square

Proof of Lemma 2. Recall that, for any plane graph G , a path of two distinct edges is called a chain. If this path follows two consecutive edges of a face of G in clockwise order around the face, we call it a *facial* chain. Equivalently, a facial chain consists of two distinct edges that are consecutive in the counterclockwise order of edges around their central vertex.

The self-touching linkage $L = g^*(M)$ is obtained from M by modifying the magnified views of M . In particular, each nonterminal-nonterminal connection representing an edge e in a magnified view of M is replaced by multiple copies corresponding to the edges $g^{-1}(e)$. Likewise, for each vertex w in H , in the magnified view of M containing w , the component containing w is replaced by the planar magnified view graph of g around w . The resulting self-touching linkage indeed respects the planar embedding of G , which can be verified as in Lemma 1.

Now suppose $t \mapsto M_t$ defines a valid motion of M ; we wish to show $t \mapsto L_t = g^*(M_t)$ is a valid motion of L . As in Ref. 7, we must show that no active vertex-edge, edge-edge, or vertex-chain constraint is violated geometrically *or* combinatorially.

We consider vertex-edge constraints first. If at time s a vertex $v \in V(G)$ is in a neighborhood of the relative interior of edge $e \in E(G)$, then the configuration uniquely specifies which side of e vertex v lies on, whether by geometry or—if geometrically lies in the relative interior of e —by the magnified view at v of the configuration L_s . We must verify that this sidedness answer stays consistent in a neighborhood of time s . It follows from the definition of plane homomorphism that v 's side of e is the same as $g(v)$'s side of $g(e)$, and since these satisfy their vertex-edge constraint in motion $t \mapsto M_t$, it follows that v and e do not violate their constraint in L_t .

For edge-edge constraints between, say, edges e_1 and e_2 , if $g(e_1) = g(e_2)$ then the configurations induced by the plane homomorphism guarantee that these edge do not switch sides. Otherwise, the constraint holds by the same argument used for vertex-edge constraints.

Finally, consider an active vertex-chain constraint with vertex v and facial chain c that visits vertices v_0, v_1, v_2 and edges e_0, e_1 in order in G . We must show that, if v is inside chain c in a small neighborhood of v_1 at some time t , whether geometrically or—if v lies on one of these edges—combinatorially according to v 's magnified view, then this remains true in a neighborhood of time t . There are two cases. If $g(e_0) \neq g(e_1)$, then $g(c)$ is a (not necessarily facial) chain in H , and the facial chain that bounds $g(v)$ near $g(v_1)$ restricts $g(v)$'s motion to a (geometric and combinatorial) wedge contained in the corresponding wedge $g(e_0e_1)$. So v indeed does not leave wedge e_0e_1 for positive time, and the constraint is not violated. Otherwise, $g(e_0) = g(e_1)$. Let us write $I_g(v_1, v_0) = I_g(v_1, v_2) = [e'_1, \dots, e'_k]$, which contains edges e_0 and e_1 . By the definition of induced configurations, it follows that e'_1, \dots, e'_k are consecutive *without interruptions* in the magnified view around v at time t . Because v and its adjacent edges are between e_0 and e_1 at this time, it follows that, in fact, the edges adjacent to v are among the edges e'_1, \dots, e'_k and lie between e_0 and e_1 in the linear order, and furthermore, that $g(v) = g(v_1)$. So v is pinned inside wedge

8 *Abel, Demaine, Demaine, Eisenstat, Lynch, Schardl, Shapiro-Elowitz*

e_0e_1 at all times by plane homomorphism g , and the constraint is not violated. \square

3. Linear Equilateral Trees

In this section, we consider the question of whether a linear equilateral tree can be “unfolded.” Recall that a *linear* (which we also refer to in this paper as *flat*) state of a graph is a state where all edges geometrically lie on a line.⁴ For trees, a *canonical* state with root vertex v is a horizontal linear state where all simple paths in the tree starting at v proceed monotonically to the right. Note that Ballinger et al.⁴ call this a “flat configuration”; we use the term “canonical state” instead to minimize the ambiguity of the word “flat.”

It is useful to interpret canonical states of trees as “unfolded” states, because all canonical states are equivalent: for any vertex v' and any edge e incident to v' , there exists a continuous motion from any canonical state to the canonical state rooted at v' in which edge e is the topmost edge incident to v' .⁶

Not all linear trees can be deformed into a canonical state; Ballinger et al. provide multiple such examples. Likewise, not all equilateral trees are universally foldable: as shown in Ref. 4, there are configurations of equilateral trees that cannot be deformed into a canonical state. By contrast, for tree configurations that are both linear *and* equilateral, we show:

Theorem 1. *Any linear configuration of an equilateral tree can be continuously deformed (without overlap) into a canonical state.*

Our algorithm proceeds roughly as follows. The initial linear state is “partially canonical.” We search for breaks in the “boundary” of the homomorphism, and unfold G at the location of the break to make it closer to canonical. By repeating this process, we end up with a canonical state.

We will need two definitions to make this argument precise. This first definition allows us to formally discuss the “boundary” of a plane homomorphism as a set of threshold edges:

Definition 3. Say that we have a plane graph G on n vertices, and a plane homomorphism $g : G \rightarrow H$. For each oriented edge e from w_1 to w_2 in the image of g , the edges $g^{-1}(e)$ are ordered counterclockwise around w_1 , and we define the *threshold edge* $\text{thr}(e)$ to be the first edge in that ordering. Furthermore, if $\text{thr}(e)$ has endpoints v_1, v_2 with $g(v_1) = w_1$ and $g(v_2) = w_2$, then choose orientation (v_1, v_2) for this edge.

It will also be helpful to have a definition for “partially canonical” states to measure our progress during an induction:

Definition 4. A configuration of a tree G is *k-canonical* if there exists a tree H_k on k nodes in a canonical state and a surjective plane homomorphism $g_k : G \rightarrow H_k$ such that the configuration of G is the one induced by g_k . Note that an n -canonical configuration of a tree with n nodes is in fact a canonical configuration.

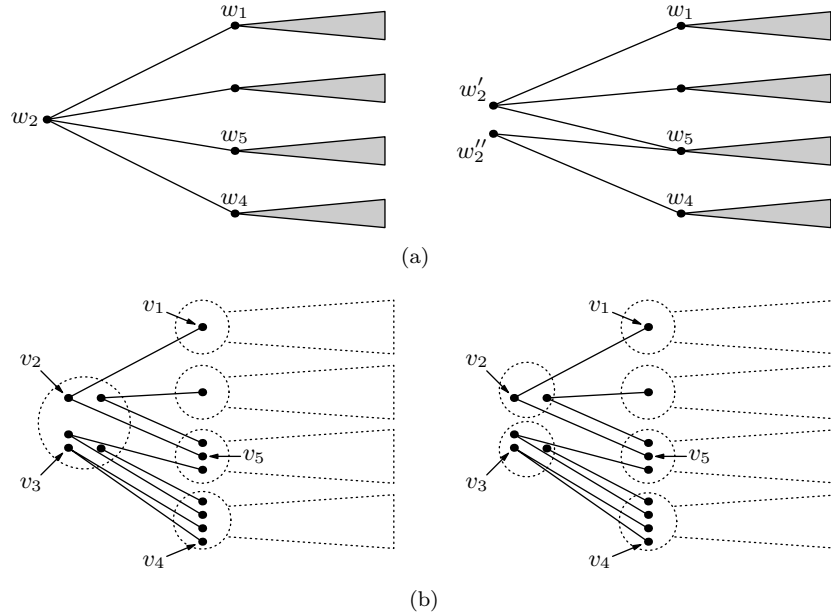
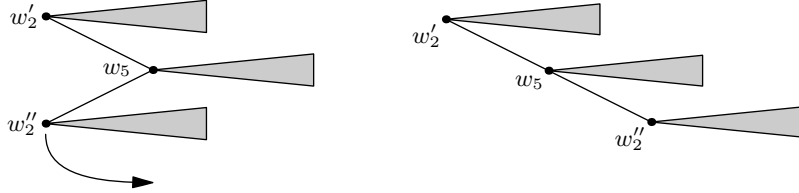


Fig. 3. An example of splitting a vertex. Figure (a) shows how the split is used to transform H_k into H_{k+1} . Figure (b) shows how the graph G is affected.

Proof (of Theorem 1). Let G be the given tree with n vertices, initially in a linear state L . This initial state of G is ℓ -canonical for some $2 \leq \ell \leq n$. We will show, by induction on $\ell \leq k \leq n$, that G can be deformed from L to a k -canonical state; the base case $k = n$ is the desired result.

Suppose that G has been deformed into the k -canonical state induced by a surjective plane homomorphism $g_k : G \rightarrow H_k$, where H_k is a tree on k vertices in a canonical state. We will show how to deform G into a $(k+1)$ -canonical configuration. Let $p_1, p_2, \dots, p_{2k-2}$ be the vertices of the outer face of H_k in clockwise order around this face, and define $d_1 = \text{thr}(p_1, p_2), d_2 = \text{thr}(p_2, p_3), \dots, d_{2k-2} = \text{thr}(p_{2k-2}, p_1)$. Intuitively, these are the oriented edges of G that lie adjacent to H_k 's outer face according to g_k . We claim that these edges cannot form a cycle in G . Assume to the contrary that they form a cycle. Because $k < n$, there must be some oriented edge $e = (w_1, w_2)$ in H_k with $|g_k^{-1}(e)| \geq 2$. Hence the edges $\text{thr}(w_1, w_2)$ and $\text{thr}(w_2, w_1)$, when taken as unoriented edges, are not equivalent, and so each appears in the cycle exactly once. The cycle is therefore nontrivial, contradicting the fact that G is a tree.

It follows that there must be some i such that the oriented edges $d_i = (v_1, v_2)$ and $d_{i+1} = (v_3, v_4)$ have $v_2 \neq v_3$. Fix some i with this property. Let $w_1 = g_k(v_1)$, $w_2 = g_k(v_2) = g_k(v_3)$, and $w_4 = g_k(v_4)$. By Ref. 6, we may move H_k to a new canonical state with w_2 at the root, and (w_2, w_1) as the topmost edge incident to w_2 (thereby making (w_2, w_4) the bottommost edge incident to w_2). This motion of

10 *Abel, Demaine, Demaine, Eisenstat, Lynch, Schardl, Shapiro-Elowitz*

 Fig. 4. Reconfiguring H_{k+1} into a canonical state.

H_k to a new state N_k induces a motion of G to a new state $M_k = g_k^*(N_k)$ lying on a horizontal line. Let (v_2, v_5) be the bottommost edge of G incident to v_2 in M_k , and let $w_5 = g_k(v_5)$. Note that w_5 may equal w_1, w_4 , both, or neither.

We next show how to “split” the vertex w_2 along edge (w_2, w_5) to construct a surjective plane graph homomorphism $g_{k+1} : G \rightarrow H_{k+1}$, where H_{k+1} is a tree with $k + 1$ vertices. We construct H_{k+1} from H_k by replacing w_2 with two vertices w'_2 and w''_2 , as depicted in Figure 3(a). Vertex w'_2 is connected to all neighbors of w_2 between w_5 and w_1 inclusive in the counterclockwise ordering of the edges around w_2 , and likewise, vertex w''_2 is connected to all neighbors of w_2 between w_4 and w_5 inclusive in the counterclockwise ordering. Edges (w_5, w'_2) and (w_5, w''_2) replace (w_5, w_2) in the counterclockwise order of edges around w_5 , with (w_5, w'_2) coming before (w_5, w''_2) . Splitting the node in this way naturally yields a surjective plane graph homomorphism $h : H_{k+1} \rightarrow H_k$ sending w'_2 and w''_2 to w_2 , which in turn defines a planar configuration $P = h^*(N_k)$ of H_{k+1} .

This construction also yields a plane homomorphism $g_{k+1} : G \rightarrow H_{k+1}$ defined as follows: In the counterclockwise order $I_{g_k}(w_5, w_2)$ on $g_k^{-1}(w_5, w_2)$, those edges before and including (v_5, v_2) (equivalently, lying above (v_5, v_2) in M_k) map to (w_5, w'_2) in H_{k+1} with the same ordering, and the rest map to (w_5, w''_2) . The rest of g_{k+1} is defined to match g_k . This can be checked to be well defined. We may also prove surjectivity: Homomorphism g_{k+1} hits every edge of H_{k+1} except possibly (w''_2, w_5) , and the connected graph G cannot surject onto the disconnected graph $H_{k+1} \setminus \{(w''_2, w_5)\}$, so this edge is in the image of g_{k+1} . We also have $g_{k+1} \circ h = g_k$, and it follows that the current configuration on G , namely M_k , is induced by g_{k+1} : indeed, $M_k = g_k^*(N_k) = g_{k+1}^*(h^*(N_k)) = g_{k+1}^*(P)$.

Finally, we use plane homomorphism g_{k+1} to reconfigure G from M_k to a $(k + 1)$ -canonical state. Consider (H_{k+1}, P) schematically as in Figure 3(b) with two edges (w'_2, w_5) and (w''_2, w_5) and a canonical subtree rooted at each of these vertices. Swinging edge (w''_2, w_5) around w_5 while holding the subtree rooted at w''_2 horizontal, as in Figure 4, reconfigures H_{k+1} into a canonical state with root w'_2 . This induces a motion on G , resulting in a $(k + 1)$ -canonical state. \square

4. Flat-Foldable Planar Graphs

We now consider the more general question of instantaneously folding a plane graph into a linear state. In this section we show:

Theorem 2. *Given a plane graph G , there exists a linear equilateral linkage configuration with the same planar embedding if and only if G is bipartite.*

Proof. Suppose graph G has a linear equilateral linkage configuration. This configuration can be accordion-folded into a configuration whose geometric graph consists of a single edge of length 1. The two nodes of this graph induce a bipartite structure on G , so G is bipartite.

Conversely, consider a bipartite graph G with a planar embedding and $n = |V(G)|$ vertices. Without loss of generality, we may assume G is connected. We proceed by induction, showing roughly that we can repeatedly fold together adjacent edges on a face until only two vertices remain. Specifically, we show by downward induction on $n \geq k \geq 2$ that there exists a plane homomorphism from G to a bipartite graph H_k on k vertices. The configuration induced by the plane homomorphism $G \rightarrow H_2$ will yield a linear state of G .

The base case $k = n$ is satisfied by the identity homomorphism $G \rightarrow G$. Now suppose we have a plane homomorphism $h_k : G \rightarrow H_k$ for some $k \geq 3$. It can be verified that there must therefore exist at least one face F in H_k with at least 4 edges because H_k is bipartite. Face F must contain at least two adjacent edges (u_1, u_2) and (u_2, u_3) such that u_1, u_2 , and u_3 are all distinct. We now “fold” these two edges together: define H_{k-1} as the plane graph obtained by first inserting edge (u_1, u_3) into H_k inside face F and then contracting this edge to a vertex w . This operation defines a plane homomorphism from H_k to H_{k-1} sending u_1 and u_3 to w , and furthermore H_{k-1} is bipartite. The composed plane homomorphism, $G \rightarrow H_k \rightarrow H_{k-1}$ proves the inductive step. Any configuration of H_2 must be linear, and therefore the configuration induced by $G \rightarrow H_2$ is also linear. \square

5. NP-Hardness of Graph Folding

Although it is possible to determine in polynomial time whether an equilateral graph has a linear state, it is hard to determine whether a weighted graph has a linear state. Consider the problem when restricted to cycles. Because the cycle need not fold at every vertex, it is possible to reduce from the integer partition problem (defined in Ref. 12) by creating a cycle whose edge lengths are the numbers to partition. Hence, it is weakly NP-hard to determine whether a weighted cycle has a linear state. In this section, we show that the problem on general weighted graphs (not necessarily cycles) is strongly NP-hard via a reduction from planar monotone 3-SAT, which is known to be NP-hard.⁸

5.1. Reduction Overview

Let $G = (U \cup (C_+ \cup C_-), E)$ be a plane graph encoding an instance of the planar monotone 3-SAT problem. Specifically, let $U = (x_1, x_2, \dots, x_n)$ denote a sequence of n variables that lie along the y -axis in order with x_1 on top. Let C_+ denote a 3-CNF formula over U containing only positive literals, and similarly let C_- denote a 3-CNF formula over U containing only negative literals. The clauses $c \in C_+$ have x -coordinate less than zero, and the clauses $c \in C_-$ have x -coordinate greater than zero. The edge set E of the graph G consists of all edges $(x, c) \in U \times (C_+ \cup C_-)$ such that clause c contains either x or \bar{x} .

We first define a new graph G' from G as follows. Each variable vertex x in G with degree k “splits” into k copies of itself in G' , thus forming a longer vertical line of variable vertices, and each clause connects to a copy of each of its literals such that each variable copy connects to at most one clause. This can be done while preserving planarity.

The idea of the reduction is as follows (see Figure 5): we encode the original graph G with a linkage where we represent each variable $x_i \in G$ with a “variable gadget” as shown in Figure 5(a), which encodes as many copies of x_i as are to be found in G' . Each clause is represented with the “clause gadget” shown in Figure 5(c). Connections between variable and clause gadgets are dictated by planar graph G' . Each variable gadget may independently “point” right or left, indicating true or false values for the variable. The clause gadgets fit onto this central chain of variables, and the “probe” of a clause gadget fits inside its face if and only if one of the variables points away from the clause. We now discuss these constructions in detail.

5.2. Variable Gadget

First we describe the variable gadget. Consider the *doubled angle-fish* plane graph illustrated in Figure 6. The triangles with side lengths 1, 1, 2 must be flat in any planar configuration of this graph, so we draw these flat triangles as straight segments with the understanding that they may not bend at their midpoint. It may be checked that this graph has exactly 12 linear configurations, precisely two of which have u_1 and u_2 exactly 4 units apart. One such configuration is given by laying the vertices along a horizontal line with x -coordinates as in Figure 6; in this configuration, its *base-points* s_1 and s_2 are “pointing left.” The other configuration is the mirror image, with base-points “pointing right.” Observe that in both configurations, vertices s_1 and s_2 are 4 units apart, as are vertices u_3 and u_4 .

The full variable gadget is illustrated in Figure 5(a), made from many doubled angle-fish subgraphs. Specifically, for each variable x_i with degree k in G , we construct a *k-instance variable gadget*, as exemplified in Figure 5(a) for $k = 4$. We refer to the vertices $\{v_{i,0}, \dots, v_{i,k}\}$ as the (*positive*) *spine points* of the variable gadget and the points $\{w_{i,1}, \dots, w_{i,k}\}$ as the (*positive*) *flex points* of the variable gadget. Similarly, we define the points $\{v'_{i,0}, \dots, v'_{i,k}\}$ and $\{w'_{i,0}, \dots, w'_{i,k}\}$ to be the

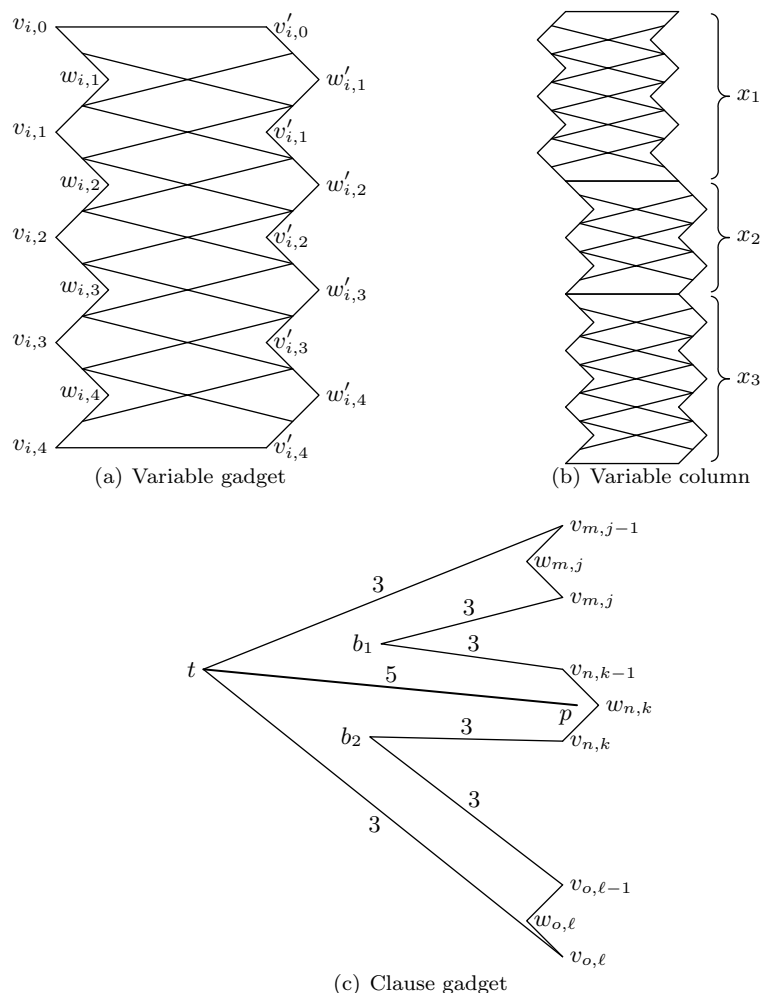


Fig. 5. Figure (a) shows a 4-entry variable gadget pointing right, i.e., set to true. The edge length between any pair of adjacent spine and flex points on one side is 2. Figure (b) shows an example variable column containing variable gadgets for three variables x_1, x_2 , and x_3 . Variables x_2 and x_3 are set to true, while x_1 is set to false. Figure (c) shows an example clause gadget. The six edges connecting t, b_1 , and b_2 to the spine points all have length 3, while the probe (edge (t, p)) has length 5.

(negative) spine points and *(negative) flex points* respectively. (We omit the “positive” and “negative” specifiers when it is clear from context.) Recall that, according to the description of the doubled angle-fish graph, all edges in the variable gadget connecting a spine point with a flex point are in fact flat 1, 1, 2 triangles.

We show that the i th variable gadget has exactly two linear configurations—one with the points $w_{i,j}$ and $w'_{i,j}$ pointing right (as illustrated), and one with them

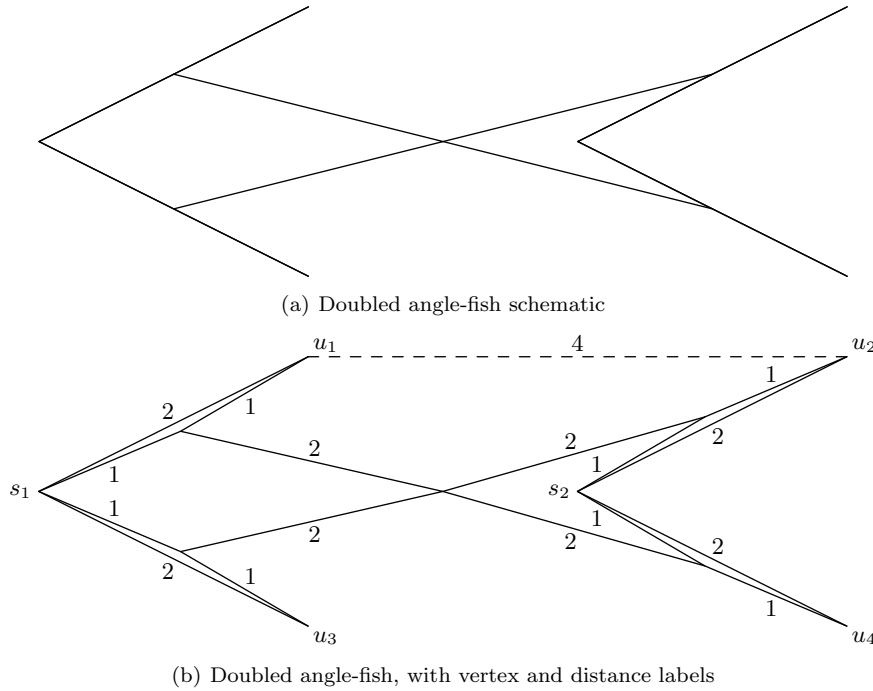


Fig. 6. Figure (a) shows a schematic for the doubled angle-fish plane graph. Figure (b) shows the same doubled angle-fish graph with important vertices named and edges and distances labeled with their lengths. To make the edges of the graph visually distinct, the drawing of the doubled angle-fish in Figure (b) does not respect edge lengths. The dashed line in Figure (b) indicates the distance from u_1 to u_2 and is not an edge in the graph.

pointing left. In truth values, these configurations will correspond to k copies of variable x_i (specifically, the k copies of x_i in graph G') all having value true (right, as illustrated) or false (left).

Lemma 3. *The k -instance variable gadget for variable x_i , defined as illustrated in Figure 5(a), has exactly two linear states shown, namely, the flat configuration indicated in the illustration and its mirror image. In particular, edges $(v_{i,0}, w_{i,1}), (w_{i,1}, v_{i,1}), (v_{i,1}, w_{i,2}), \dots, (w_{i,k}, v_{i,k})$ all lie above each other in that order, and similarly for $(v'_{i,0}, w'_{i,1}), (w'_{i,1}, v'_{i,1}), (v'_{i,1}, w'_{i,2}), \dots, (w'_{i,k}, v'_{i,k})$.*

Proof. We must verify that these are the only linear configurations. To this end, consider any linear configuration of this variable gadget. Note that the doubled angle-fish based at $w_{i,1}$ and $w'_{i,1}$ indeed has vertices $v_{i,0}$ and $v'_{i,0}$ constrained at 4 units apart, so this subgraph must be configured in the left-pointing or right-pointing state. We may assume it points right by symmetry. Note also that $w_{i,1}$ and $w'_{i,1}$ are 4 units apart, so the doubled angle-fish based at $v_{i,1}$ and $v'_{i,1}$ must also be left- or right-facing. By its overlap with the first angle-fish, we may conclude that

this second angle-fish in fact points left. Continuing inductively down the variable clause, we find that the given configuration is identical to the one illustrated in Figure 5(a). \square

The variable gadgets for variables x_1, \dots, x_n are joined together to form the *variable column* as illustrated in Figure 5(b). By construction, this graph has exactly 2^n linear configurations—one corresponding to each truth assignment to variables x_1, \dots, x_n .

5.3. Clause Gadget

We now describe the clause gadget for clause c . We describe the case in which $c \in C_+$ contains three literals; the cases of one or two literals are analogous, and the C_- clauses are symmetric. Clause c is represented using four new vertices t, p, b_1, b_2 and six new edges as depicted in Figure 5(c). The clause gadget for c is connected to the three variable entries corresponding to c 's neighbors in G' , thereby preventing the clause gadgets from intersecting each other. The *probe* edge (t, p) is a length-5 edge inside the clause gadget that permits a linear state if and only if c is satisfied. The *blockers*—vertices b_1, b_2 and their incident length-3 edges—prevent the probe from accessing variable entries for variables not in c . By construction, the clause gadget has the following property:

Lemma 4. *For each clause $c \in C_+ \cup C_-$, the variable column with the clause gadget for c has a linear state if and only if c is satisfied.*

Proof. If a clause is satisfied, then the clause gadget spans a distance of five and the probe gadget can be placed in the satisfied variable as in Figure 5(c). The diagram demonstrates a linear state.

To prove necessity, consider a linear state of the clause gadget c ; by symmetry, we may assume $c \in C_+$. The structure of the variable column ensures that all of the positive spine points must be collocated at a point. Consider the magnified view of the linear configuration around this point. From the ordering of the spine points and the combinatorial embedding, the counterclockwise ordering of the edges incident to t, b_1 , and b_2 must be consistent with the top-to-bottom ordering shown in Figure 5(c). The points b_1, b_2 , and t must lie three units away from the spine, and because the distance from the spine to each flex point is two, b_1 and b_2 must lie at the same location as t . The probe has length five and is on the internal face of the clause gadget, so t must point away from the variable column, and at least one of the three variable entries must be set to true. Hence the clause gadget and variable column have a linear configuration only if the clause is satisfied. \square

5.4. Proofs of Hardness

Theorem 3. *Determining whether a plane graph has a linear state is strongly NP-complete.*

Proof. It is possible to verify in polynomial time whether a graph G is in a linear state by checking whether the embedding is planar, linear, and consistent with G , so the problem is in NP.

Let G be an instance of the planar monotone 3-SAT problem, and let H be the graph generated by this reduction. Given a satisfying assignment of G , we position the flex points according to the assignment, and by Lemma 4 each clause has a linear state, yielding a global linear state for H . Every linear state of H corresponds to an assignment to the variables, and by Lemma 4 the assignment must satisfy all clauses. Hence, the reduction is correct. Because the reduction takes polynomial time, the problem is strongly NP-hard. \square

As mentioned in Section 1.2, the problem of linearly folding graphs corresponds to a generalization of the single-vertex origami problem dealing with a *single-vertex complex* instead of simply a cone of paper. The primary difference is that a plane graph comes with a chosen outer face whereas the local structure of a single-vertex complex more naturally lies on the sphere and does not have a chosen outer face. It is thus natural to study the question of linearly folding a given metric *sphere graph*, i.e., a metric plane graph with the additional freedom to pick any face as the outer face. A small modification of the above reduction shows that this problem is also strongly NP-hard.

Theorem 4. *Determining whether a metric sphere graph has a linear state is strongly NP-complete.*

Proof. For an instance of the planar monotone 3-SAT problem, let H be the graph generated by the reduction used for Theorem 3. Let ρ be the perimeter of the outside face of H , and attach a leaf edge e to vertex $v_{1,0}$ (at the top of the variable column) of length $\rho + 1$ in the outside face to form plane graph H' . The output of the reduction is graph H' interpreted as a sphere graph.

If sphere graph H' has a linear embedding, then the chosen outer face must agree with that of H , because otherwise edge e does not fit inside its face. Conversely, if H has a linear configuration then edge e may be inserted into to this configuration adjacent to edge $(v_{1,0}, v'_{1,0})$. This completes the proof. \square

Corollary 1. *Given a plane graph or a metric sphere graph with edge lengths in $\{1, 2\}$, determining whether the graph has a linear state is strongly NP-complete.*

Proof. We may simulate an edge of integer length $r \geq 3$ by joining $r - 1$ flat $1, 1, 2$ triangles; this uses a total of $2r - 1$ edges. The hardness results follow by applying this transformation to each edge of length greater than 2 in the graph produced by the reductions of Theorems 3 and 4, respectively. \square

6. Open Problems

Our study of instantaneous flat foldings of single-vertex complexes leaves open natural directions of exploration, which can be translated into open problems about instantaneous flat foldings of plane graphs.

First, what if we require all creases to be folded, i.e., the flat folding to not have any 180° angles between incident edges? For a cycle, this condition enables testing flat foldability in linear time by Kawasaki's Theorem,¹⁰ whereas the problem is weakly NP-complete in the setting considered here in which each crease may or may not be folded. For equilateral graphs, bipartiteness is again necessary and sufficient, because any linear folding of an equilateral graph may be accordion-folded to fold every crease. For general metric graphs, does the problem remain strongly NP-complete, or does the extra information enable efficient algorithms?

Second, what if we further require a specified mountain-valley assignment, i.e., every two consecutive edges around a vertex must form a specified angle of 0° or 360° ? This problem is even more constrained than the previous problem, and even the equilateral case is open. A natural analogy is the NP-hardness of instantaneous flat folding of general crease patterns,⁵ which holds with or without a prescribed mountain-valley assignment.

Finally, when we can find a flat folded state such as the equilateral case, we can ask additional questions about finding the “best” folded state. Two natural measures, previously studied in the context of a path, are minimizing the maximum *crease width* (the number of layers stuck between two consecutive edges around a vertex) and minimizing the total crease width.^{16,17}

Acknowledgments

We thank Ilya Baran for early discussions about instantaneous graph folding, in particular conjecturing Theorem 2. We also thank Muriel Dulieu for helpful discussions on this topic. This research was continued during the open-problem sessions organized around MIT class 6.849: Geometric Folding Algorithms in Fall 2010. We thank the other participants of these sessions—Scott Kominers, Jason Ku, Thomas Morgan, Jie Qi, Tomohiro Tachi, and Andrew Winslow—for providing a conducive research environment.

References

1. G. Aloupis, E. D. Demaine, V. Dujmovi, J. Erickson, S. Langerman, H. Meijer, M. Overmars, M. Soss, I. Streinu and G. T. Toussaint, Flat-state connectivity of linkages under dihedral motions, in *In Proc. 13th Annu. Internat. Sympos. Alg. Comput* (Springer, 2002) pp. 369–380.
2. G. Aloupis, E. D. Demaine, H. Meijer, J. O'Rourke, I. Streinu and G. Toussaint, On flat-state connectivity of chains with fixed acute angles, in *In Proceedings of the 14th Canadian Conference on Computational Geometry* (2002) pp. 27–30.
3. H. Alt, C. Knauer, G. Rote and S. Whitesides, On the complexity of the linkage recon-

18 *Abel, Demaine, Demaine, Eisenstat, Lynch, Schardl, Shapiro-Elowitz*

- figuration problem, in *Towards a Theory of Geometric Graphs*, ed. J. Pach (American Mathematical Society, 2004), pp. 1–14.
4. B. Ballinger, D. Charlton, E. D. Demaine, M. L. Demaine, J. Iacono, C.-H. Liu and S.-H. Poon, Minimal locked trees, in *Proceedings of the 11th Algorithms and Data Structures Symposium* (Banff, Canada, 2009) pp. 61–73.
 5. M. Bern and B. Hayes, The complexity of flat origami, in *Proceedings of the 7th Annual ACM-SIAM Symposium on Discrete Algorithms* (Atlanta, 1996) pp. 175–183.
 6. T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O’Rourke, S. Robbins, I. Streinu, G. Toussaint and S. Whitesides, A note on reconfiguring tree linkages: Trees can lock, *Discrete Applied Mathematics* **117** (2002) 293–297.
 7. R. Connelly, E. D. Demaine and G. Rote, Infinitesimally locked self-touching linkages with applications to locked trees, in *Physical Knots: Knotting, Linking, and Folding of Geometric Objects in R^3* , eds. J. Calvo, K. Millett and E. Rawdon (American Mathematical Society, 2002) pp. 287–311.
 8. M. de Berg and A. Khosravi, Optimal binary space partitions in the plane, in *Proceedings of the 16th Annual International Conference on Computing and Combinatorics* (2010) pp. 216–225.
 9. E. D. Demaine, S. L. Devadoss, J. S. B. Mitchell and J. O’Rourke, Continuous foldability of polygonal paper, in *Proceedings of the 16th Canadian Conference on Computational Geometry* (Montréal, Canada, 2004) pp. 64–67.
 10. E. D. Demaine and J. O’Rourke, *Geometric Folding Algorithms: Linkages, Origami, Polyhedra* (Cambridge University Press, 2007), chap. 12.
 11. J. Justin, Aspects mathématiques du pliage de papier (Mathematical aspects of paper folding), in *Proceedings of the 1st International Meeting of Origami Science and Technology*, ed. H. Huzita (Ferrara, Italy, 1989) pp. 263–277.
 12. R. M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, eds. R. E. Miller and J. W. Thatcher (Plenum Press, New York, NY, 1972) pp. 85–103.
 13. T. Kawasaki, On the relation between mountain-creases and valley-creases of a flat origami, in *Proceedings of the 1st International Meeting of Origami Science and Technology*, ed. H. Huzita (Ferrara, Italy, 1989) pp. 229–237.
 14. S. Poon, On straightening low-diameter unit trees, in *Graph Drawing* (Springer, 2006) pp. 519–521.
 15. I. Streinu and W. Whiteley, Single-vertex origami and spherical expansive motions, in *Revised Selected Papers from the Japan Conference on Discrete and Computational Geometry*, eds. J. Akiyama, M. Kano and X. Tan (Tokyo, Japan, 2004) pp. 161–173.
 16. R. Uehara, Stamp foldings with a given mountain-valley assignment, in *Origami⁵: Proceedings of the 5th International Meeting of Origami Science, Mathematics, and Education* (A K Peters/CRC Press, 2011), pp. 585–597.
 17. T. Umehato, T. Saitoh, R. Uehara and H. Ito, Complexity of the stamp folding problem, in *Proceedings of the 5th international conference on Combinatorial optimization and applications* (Zhangjiajie, China, 2011) pp. 311–321.